Open Source Used In Cisco DNA Center Platform 1.3.1.0

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please contact us at external-opensource-requests@cisco.com.

In your requests please include the following reference number 78EE117C99-202623217

Contents

1.1 ajv 5.5.2

1.1.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-2017 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2 ajv-keywords 3.1.0

1.2.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Evgeny Poberezkin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3 akkahttp 10.0.9

1.3.1 Available under license :

```
/*
* Copyright (C) 2018 Lightbend Inc. <https://www.lightbend.com>
*/

package akka

import sbt._, Keys._
import de.heikoseeberger.sbtheader.{ CommentCreator, HeaderPlugin }

object CopyrightHeader extends AutoPlugin {
 import HeaderPlugin.autoImport._
 import ValidatePullRequest.{ additionalTasks, ValidatePR }
```

```scala
override def requires = HeaderPlugin
override def trigger = allRequirements

override def projectSettings = Def.settings(
  Seq(Compile, Test).flatMap { config =>
    inConfig(config)(
      Seq(
        headerLicense := Some(HeaderLicense.Custom(headerFor(CurrentYear))),
        headerMappings := headerMappings.value ++ Map(
          HeaderFileType.scala      -> cStyleComment,
          HeaderFileType.java       -> cStyleComment,
          HeaderFileType("template") -> cStyleComment
        )
      )
    )
  },
  additionalTasks in ValidatePR += headerCheck in Compile,
  additionalTasks in ValidatePR += headerCheck in Test
)

val CurrentYear = java.time.Year.now.getValue.toString
val CopyrightPattern = "Copyright \\([Cc]\\) (\\d{4}(-\\d{4})?) (Lightbend|Typesafe) Inc. <.*>".r
val CopyrightHeaderPattern = s"(?s).*${CopyrightPattern}.*".r

def headerFor(year: String): String =
  s"Copyright (C) $year Lightbend Inc. <https://www.lightbend.com>"

val cStyleComment = HeaderCommentStyle.cStyleBlockComment.copy(commentCreator = new
CommentCreator() {
  import HeaderCommentStyle.cStyleBlockComment.commentCreator

  def updateLightbendHeader(header: String): String = header match {
    case CopyrightHeaderPattern(years, null, _) =>
      if (years != CurrentYear)
        CopyrightPattern.replaceFirstIn(header, headerFor(years + "-" + CurrentYear))
      else
        CopyrightPattern.replaceFirstIn(header, headerFor(years))
    case CopyrightHeaderPattern(years, endYears, _) =>
      CopyrightPattern.replaceFirstIn(header, headerFor(years.replace(endYears, "-" + CurrentYear)))
    case _ =>
      header
  }

  override def apply(text: String, existingText: Option[String]): String = {
    existingText
      .map(updateLightbendHeader)
      .getOrElse(commentCreator(text, existingText))
```

```
        .trim
    }
  })
}
```

This software is licensed under the Apache 2 license, quoted below.

Copyright 2009-2018 Lightbend Inc. [https://www.lightbend.com]

Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License. You may obtain a copy of
the License at

  [http://www.apache.org/licenses/LICENSE-2.0]

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations under
the License.

1.4 akkahttpcore 10.0.9

1.5 akkahttpjackson 10.0.9

1.5.1 Available under license :

Copyright 2015 Heiko Seeberger

                          Apache License
                    Version 2.0, January 2004
                   http://www.apache.org/licenses/

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

    1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the

```

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.6 akkahttptestkit 10.0.9

# 1.7 akkaslf4j 2.5.6

# 1.8 akkastream 2.5.6

# 1.9 api-spec-converter 2.6.0

## 1.9.1 Available under license :

core-js@2.5.1

MIT

Copyright (c) 2014-2017 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

webpack@3.6.0

MIT

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

zone.js@0.8.18
MIT
The MIT License

Copyright (c) 2016 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

@angular/core@4.4.5
MIT
MIT

@angular/router@4.4.5
MIT
MIT

@angular/http@4.4.5
MIT
MIT

@angular/forms@4.4.5
MIT

MIT

@angular/platform-browser@4.4.5
MIT
MIT

@angular/common@4.4.5
MIT
MIT
babel-polyfill@6.26.0
MIT
MIT

core-js@2.5.1
MIT
Copyright (c) 2014-2017 Denis Pushkarev

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

webpack@3.6.0
MIT
Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

zone.js@0.8.18
MIT
The MIT License

Copyright (c) 2016 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

@angular/core@4.4.5
MIT
MIT

@angular/http@4.4.5
MIT
MIT

file-saver@1.3.3
MIT
The MIT License

Copyright  2016 [Eli Grey][1].

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

 [1]: http://eligrey.com

@angular/forms@4.4.5
MIT
MIT

@angular/platform-browser@4.4.5
MIT
MIT

@angular/common@4.4.5
MIT
MIT
(MIT License)

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.10 axios 0.16.2

## 1.10.1 Available under license :

Copyright (c) 2014-present Matt Zabriskie

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

# 1.11 babel-cli 6.8.0

# 1.12 babel-cli 6.26.0

# 1.13 babel-core 6.8.0

# 1.14 babel-core 6.26.0

# 1.15 babel-eslint 8.2.2

## 1.15.1 Available under license :

Copyright (c) 2014-2016 Sebastian McKenzie <sebmck@gmail.com>

MIT License

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.16 babel-jest 21.2.0

## 1.16.1 Available under license :

MIT License

For Jest software

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.17 babel-jest 21.2.0

# 1.18 babel-plugin-transform-async-to-generator 6.24.1

# 1.19 babel-plugin-transform-class-properties 6.24.1

# 1.20 babel-plugin-transform-regenerator 6.26.0

# 1.21 babel-preset-es2015 6.24.1

## 1.21.1 Available under license :

Apache License

Version 2.0, January 2011

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.22 babel-preset-es2015 6.6.0

## 1.22.1 Available under license :

Apache License
Version 2.0, January 2011
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!)  The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.23 babel-preset-es2016 6.24.1

# 1.24 babel-preset-react 6.24.1

# 1.25 babel-preset-stage-2 6.24.1

# 1.26 babel-preset-stage-3 6.24.1

# 1.27 body-parser 1.18.1

## 1.27.1 Available under license :

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.28 bootstrap 3.3.7

## 1.28.1 Available under license :

---
layout: docs
title: License FAQs
description: Commonly asked questions about Bootstrap's open source license.
group: about
---

Bootstrap is released under the MIT license and is copyright {{ site.time | date: "%Y" }} Twitter. Boiled down to smaller chunks, it can be described with the following conditions.

#### It requires you to:

* Keep the license and copyright notice included in Bootstrap's CSS and JavaScript files when you use them in your works

#### It permits you to:

- Freely download and use Bootstrap, in whole or in part, for personal, private, company internal, or commercial purposes
- Use Bootstrap in packages or distributions that you create
- Modify the source code
- Grant a sublicense to modify and distribute Bootstrap to third parties not included in the license

#### It forbids you to:

- Hold the authors and license owners liable for damages as Bootstrap is provided without warranty
- Hold the creators or copyright holders of Bootstrap liable
- Redistribute any piece of Bootstrap without proper attribution
- Use any marks owned by Twitter in any way that might state or imply that Twitter endorses your distribution

- Use any marks owned by Twitter in any way that might state or imply that you created the Twitter software in question

#### It does not require you to:

- Include the source of Bootstrap itself, or of any modifications you may have made to it, in any redistribution you may assemble that includes it
- Submit changes that you make to Bootstrap back to the Bootstrap project (though such feedback is encouraged)

The full Bootstrap license is located [in the project repository]({{ site.repo }}/blob/v{{ site.current_version }}/LICENSE) for more information.
The MIT License (MIT)

Copyright (c) 2011-2018 Twitter, Inc.
Copyright (c) 2011-2018 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

# 1.29 chai 4.1.2

## 1.29.1 Available under license :

MIT License

Copyright (c) 2017 Chai.js Assertion Library

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

# 1.30 chai-http 3.0.0

# 1.31 classnames 2.2.5

## 1.31.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

# 1.32 codemirror 5.37.0

## 1.32.1 Available under license :

MIT License

Copyright (C) 2017 by Marijn Haverbeke <marijnh@gmail.com> and others

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

# 1.33 commons-compress 1.16.1

## 1.33.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!)  The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Apache Commons Compress
Copyright 2002-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (https://www.apache.org/).

The files in the package org.apache.commons.compress.archivers.sevenz
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)

# 1.34 commons-text 1.2

## 1.34.1 Available under license :

```
/*
* Licensed to the Apache Software Foundation (ASF) under one or more
* contributor license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright ownership.
* The ASF licenses this file to You under the Apache License, Version 2.0
* (the "License"); you may not use this file except in compliance with
* the License. You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

Apache Commons Text

Copyright 2014-2018 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!)  The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.35 commonsio 1.3.2

## 1.35.1 Available under license :

Apache Commons IO
Copyright 2002-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions

to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices

stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!)  The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.36 commonslang3 3.6

## 1.36.1 Available under license :

Apache Commons Lang
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!)  The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

# 1.37 cookie-parser 1.4.3

## 1.37.1 Available under license :

(The MIT License)

Copyright (c) 2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.38 css-loader 0.28.4

## 1.38.1 Available under license :

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# 1.39 csv-parse 2.0.0

## 1.39.1 Available under license :

Software License Agreement (BSD License)
==========================================

Copyright (c) 2011, SARL Adaltas.

All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of SARL Adaltas nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of the SARL Adaltas.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# 1.40 d3 3.5.17

## 1.40.1 Available under license :

Copyright 2010-2017 Mike Bostock
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
 list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
 this list of conditions and the following disclaimer in the documentation
 and/or other materials provided with the distribution.

* Neither the name of the author nor the names of contributors may be used to
 endorse or promote products derived from this software without specific prior
 written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

# 1.41 eclipse-github 2.1.15

## 1.41.1 Available under license :

```
/****************************************************************************
 * Copyright (c) 2011 GitHub Inc.
 * All rights reserved. This program and the accompanying materials
 * are made available under the terms of the Eclipse Public License 2.0
 * which accompanies this distribution, and is available at
 * https://www.eclipse.org/legal/epl-2.0/
 *
 * SPDX-License-Identifier: EPL-2.0
 *
 * Contributors:
 *   Kevin Sawicki (GitHub Inc.) - initial API and implementation
 ****************************************************************************/
package org.eclipse.egit.github.core;

import java.io.Serializable;

/**
 * Repository contributor model class
 */
public class Contributor implements Serializable {

 /** serialVersionUID */
 private static final long serialVersionUID = -8434028880839230626L;

 /**
  * Anonymous contributor type value
  */
 public static final String TYPE_ANONYMOUS = "Anonymous"; //$NON-NLS-1$

 private int contributions;

 private int id;

 private String avatarUrl;

 private String login;
```

```java
    private String name;

    private String type;

    private String url;

    /**
     * @return contributions
     */
    public int getContributions() {
     return contributions;
    }

    /**
     * @param contributions
     * @return this contributor
     */
    public Contributor setContributions(int contributions) {
     this.contributions = contributions;
     return this;
    }

    /**
     * @return id
     */
    public int getId() {
     return id;
    }

    /**
     * @param id
     * @return this contributor
     */
    public Contributor setId(int id) {
     this.id = id;
     return this;
    }

    /**
     * @return avatarUrl
     */
    public String getAvatarUrl() {
     return avatarUrl;
    }

    /**
     * @param avatarUrl
     * @return this contributor
```

```java
 */
public Contributor setAvatarUrl(String avatarUrl) {
 this.avatarUrl = avatarUrl;
 return this;
}

/**
 * @return login
 */
public String getLogin() {
 return login;
}

/**
 * @param login
 * @return this contributor
 */
public Contributor setLogin(String login) {
 this.login = login;
 return this;
}

/**
 * @return name
 */
public String getName() {
 return name;
}

/**
 * @param name
 * @return this contributor
 */
public Contributor setName(String name) {
 this.name = name;
 return this;
}

/**
 * @return type
 */
public String getType() {
 return type;
}

/**
 * @param type
 * @return this contributor
```

```
        */
       public Contributor setType(String type) {
        this.type = type;
        return this;
       }

       /**
        * @return url
        */
       public String getUrl() {
        return url;
       }

       /**
        * @param url
        * @return this contributor
        */
       public Contributor setUrl(String url) {
        this.url = url;
        return this;
       }
      }
```

1.42 elasticsearch 6.2.0

1.42.1 Available under license :

```
      /*
      The MIT License

      Copyright (c) 2004-2015 Paul R. Holser, Jr.

      Permission is hereby granted, free of charge, to any person obtaining
      a copy of this software and associated documentation files (the
      "Software"), to deal in the Software without restriction, including
      without limitation the rights to use, copy, modify, merge, publish,
      distribute, sublicense, and/or sell copies of the Software, and to
      permit persons to whom the Software is furnished to do so, subject to
      the following conditions:

      The above copyright notice and this permission notice shall be
      included in all copies or substantial portions of the Software.

      THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
      EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
      MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
      NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
      LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
```

OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
*/
Joni is released under the MIT License.
JCodings is released under the MIT License.
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

Licensing

Jackson core and extension components may licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.
The art of simplicity is a puzzle of complexity.

Overview
[YAML](http://yaml.org) is a data serialization format designed for human readability and interaction with scripting
languages.

SnakeYAML is a YAML processor for the Java Virtual Machine.

SnakeYAML features

* a **complete** [YAML 1.1 processor](http://yaml.org/spec/1.1/current.html). In particular, SnakeYAML can parse all examples from the specification.
* Unicode support including UTF-8/UTF-16 input/output.
* high-level API for serializing and deserializing native Java objects.
* support for all types from the [YAML types repository](http://yaml.org/type/index.html).
* relatively sensible error messages.

Info
* [Changes](https://bitbucket.org/asomov/snakeyaml/wiki/Changes)
* [Documentation](https://bitbucket.org/asomov/snakeyaml/wiki/Documentation)

Contribute
* Mercurial DVCS is used to dance with the [source code](https://bitbucket.org/asomov/snakeyaml/src).
* If you find a bug in SnakeYAML, please [file a bug report](https://bitbucket.org/asomov/snakeyaml/issues?status=new&status=open).
* You may discuss SnakeYAML at [the mailing list](http://groups.google.com/group/snakeyaml-core).
This copy of Jackson JSON processor streaming parser/generator is licensed under the Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

 END OF TERMS AND CONDITIONS
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
/*
* Licensed to Elasticsearch under one or more contributor
* license agreements. See the NOTICE file distributed with
* this work for additional information regarding copyright
* ownership. Elasticsearch licenses this file to you under
* the Apache License, Version 2.0 (the "License"); you may
* not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
* software distributed under the License is distributed on an
* "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
* KIND, either express or implied. See the License for the
* specific language governing permissions and limitations
* under the License.
*/
ELASTIC LICENSE AGREEMENT

PLEASE READ CAREFULLY THIS ELASTIC LICENSE AGREEMENT (THIS "AGREEMENT"), WHICH
CONSTITUTES A LEGALLY BINDING AGREEMENT AND GOVERNS ALL OF YOUR USE OF ALL OF
THE ELASTIC SOFTWARE WITH WHICH THIS AGREEMENT IS INCLUDED ("ELASTIC SOFTWARE")
THAT IS PROVIDED IN OBJECT CODE FORMAT, AND, IN ACCORDANCE WITH SECTION 2 BELOW,
CERTAIN OF THE ELASTIC SOFTWARE THAT IS PROVIDED IN SOURCE CODE FORMAT. BY
INSTALLING OR USING ANY OF THE ELASTIC SOFTWARE GOVERNED BY THIS AGREEMENT, YOU
ARE ASSENTING TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE
WITH SUCH TERMS AND CONDITIONS, YOU MAY NOT INSTALL OR USE THE ELASTIC SOFTWARE
GOVERNED BY THIS AGREEMENT. IF YOU ARE INSTALLING OR USING THE SOFTWARE ON
BEHALF OF A LEGAL ENTITY, YOU REPRESENT AND WARRANT THAT YOU HAVE THE ACTUAL

AUTHORITY TO AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT ON BEHALF OF SUCH ENTITY.

Posted Date: April 20, 2018

This Agreement is entered into by and between Elasticsearch BV ("Elastic") and You, or the legal entity on behalf of whom You are acting (as applicable, "You").

1. OBJECT CODE END USER LICENSES, RESTRICTIONS AND THIRD PARTY OPEN SOURCE SOFTWARE

1.1 Object Code End User License. Subject to the terms and conditions of Section 1.2 of this Agreement, Elastic hereby grants to You, AT NO CHARGE and for so long as you are not in breach of any provision of this Agreement, a License to the Basic Features and Functions of the Elastic Software.

1.2 Reservation of Rights; Restrictions. As between Elastic and You, Elastic and its licensors own all right, title and interest in and to the Elastic Software, and except as expressly set forth in Sections 1.1, and 2.1 of this Agreement, no other license to the Elastic Software is granted to You under this Agreement, by implication, estoppel or otherwise. You agree not to: (i) reverse engineer or decompile, decrypt, disassemble or otherwise reduce any Elastic Software provided to You in Object Code, or any portion thereof, to Source Code, except and only to the extent any such restriction is prohibited by applicable law, (ii) except as expressly permitted in this Agreement, prepare derivative works from, modify, copy or use the Elastic Software Object Code or the Commercial Software Source Code in any manner; (iii) except as expressly permitted in Section 1.1 above, transfer, sell, rent, lease, distribute, sublicense, loan or otherwise transfer, Elastic Software Object Code, in whole or in part, to any third party; (iv) use Elastic Software Object Code for providing time-sharing services, any software-as-a-service, service bureau services or as part of an application services provider or other service offering (collectively, "SaaS Offering") where obtaining access to the Elastic Software or the features and functions of the Elastic Software is a primary reason or substantial motivation for users of the SaaS Offering to access and/or use the SaaS Offering ("Prohibited SaaS Offering"); (v) circumvent the limitations on use of Elastic Software provided to You in Object Code format that are imposed or preserved by any License Key, or (vi) alter or remove any Marks and Notices in the Elastic Software. If You have any question as to whether a specific SaaS Offering constitutes a Prohibited SaaS Offering, or are interested in obtaining Elastic's permission to engage in commercial or non-commercial distribution of the Elastic Software, please contact elastic_license@elastic.co.

1.3 Third Party Open Source Software. The Commercial Software may contain or be provided with third party open source libraries, components, utilities and other open source software (collectively, "Open Source Software"), which Open

Source Software may have applicable license terms as identified on a website designated by Elastic. Notwithstanding anything to the contrary herein, use of the Open Source Software shall be subject to the license terms and conditions applicable to such Open Source Software, to the extent required by the applicable licensor (which terms shall not restrict the license rights granted to You hereunder, but may contain additional rights). To the extent any condition of this Agreement conflicts with any license to the Open Source Software, the Open Source Software license will govern with respect to such Open Source Software only. Elastic may also separately provide you with certain open source software that is licensed by Elastic. Your use of such Elastic open source software will not be governed by this Agreement, but by the applicable open source license terms.

2. COMMERCIAL SOFTWARE SOURCE CODE

2.1 Limited License. Subject to the terms and conditions of Section 2.2 of this Agreement, Elastic hereby grants to You, AT NO CHARGE and for so long as you are not in breach of any provision of this Agreement, a limited, non-exclusive, non-transferable, fully paid up royalty free right and license to the Commercial Software in Source Code format, without the right to grant or authorize sublicenses, to prepare Derivative Works of the Commercial Software, provided You (i) do not hack the licensing mechanism, or otherwise circumvent the intended limitations on the use of Elastic Software to enable features other than Basic Features and Functions or those features You are entitled to as part of a Subscription, and (ii) use the resulting object code only for reasonable testing purposes.

2.2 Restrictions. Nothing in Section 2.1 grants You the right to (i) use the Commercial Software Source Code other than in accordance with Section 2.1 above, (ii) use a Derivative Work of the Commercial Software outside of a Non-production Environment, in any production capacity, on a temporary or permanent basis, or (iii) transfer, sell, rent, lease, distribute, sublicense, loan or otherwise make available the Commercial Software Source Code, in whole or in part, to any third party. Notwithstanding the foregoing, You may maintain a copy of the repository in which the Source Code of the Commercial Software resides and that copy may be publicly accessible, provided that you include this Agreement with Your copy of the repository.

3. TERMINATION

3.1 Termination. This Agreement will automatically terminate, whether or not You receive notice of such Termination from Elastic, if You breach any of its provisions.

3.2 Post Termination. Upon any termination of this Agreement, for any reason, You shall promptly cease the use of the Elastic Software in Object Code format and cease use of the Commercial Software in Source Code format. For the avoidance of doubt, termination of this Agreement will not affect Your right

to use Elastic Software, in either Object Code or Source Code formats, made available under the Apache License Version 2.0.

3.3 Survival. Sections 1.2, 2.2. 3.3, 4 and 5 shall survive any termination or expiration of this Agreement.

4. DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY

4.1 Disclaimer of Warranties. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, THE ELASTIC SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, AND ELASTIC AND ITS LICENSORS MAKE NO WARRANTIES WHETHER EXPRESSED, IMPLIED OR STATUTORY REGARDING OR RELATING TO THE ELASTIC SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, ELASTIC AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE ELASTIC SOFTWARE, AND WITH RESPECT TO THE USE OF THE FOREGOING. FURTHER, ELASTIC DOES NOT WARRANT RESULTS OF USE OR THAT THE ELASTIC SOFTWARE WILL BE ERROR FREE OR THAT THE USE OF THE ELASTIC SOFTWARE WILL BE UNINTERRUPTED.

4.2 Limitation of Liability. IN NO EVENT SHALL ELASTIC OR ITS LICENSORS BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY DIRECT OR INDIRECT DAMAGES, INCLUDING, WITHOUT LIMITATION, FOR ANY LOSS OF PROFITS, LOSS OF USE, BUSINESS INTERRUPTION, LOSS OF DATA, COST OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONNECTION WITH OR ARISING OUT OF THE USE OR INABILITY TO USE THE ELASTIC SOFTWARE, OR THE PERFORMANCE OF OR FAILURE TO PERFORM THIS AGREEMENT, WHETHER ALLEGED AS A BREACH OF CONTRACT OR TORTIOUS CONDUCT, INCLUDING NEGLIGENCE, EVEN IF ELASTIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

5. MISCELLANEOUS

This Agreement completely and exclusively states the entire agreement of the parties regarding the subject matter herein, and it supersedes, and its terms govern, all prior proposals, agreements, or other communications between the parties, oral or written, regarding such subject matter. This Agreement may be modified by Elastic from time to time, and any such modifications will be effective upon the "Posted Date" set forth at the top of the modified Agreement. If any provision hereof is held unenforceable, this Agreement will continue without said provision and be interpreted to reflect the original intent of the parties. This Agreement and any non-contractual obligation arising out of or in connection with it, is governed exclusively by Dutch law. This Agreement shall not be governed by the 1980 UN Convention on Contracts for the International Sale of Goods. All disputes arising out of or in connection with this Agreement, including its existence and validity, shall be resolved by the courts with jurisdiction in Amsterdam, The Netherlands, except where mandatory law provides for the courts at another location in The Netherlands to have jurisdiction. The parties hereby irrevocably waive any and all claims and defenses either might otherwise have in any such action or

proceeding in any of such courts based upon any alleged lack of personal jurisdiction, improper venue, forum non conveniens or any similar claim or defense. A breach or threatened breach, by You of Section 2 may cause irreparable harm for which damages at law may not provide adequate relief, and therefore Elastic shall be entitled to seek injunctive relief without being required to post a bond. You may not assign this Agreement (including by operation of law in connection with a merger or acquisition), in whole or in part to any third party without the prior written consent of Elastic, which may be withheld or granted by Elastic in its sole and absolute discretion. Any assignment in violation of the preceding sentence is void. Notices to Elastic may also be sent to legal@elastic.co.

6. DEFINITIONS

The following terms have the meanings ascribed:

6.1 "Affiliate" means, with respect to a party, any entity that controls, is controlled by, or which is under common control with, such party, where "control" means ownership of at least fifty percent (50%) of the outstanding voting shares of the entity, or the contractual right to establish policy for, and manage the operations of, the entity.

6.2 "Basic Features and Functions" means those features and functions of the Elastic Software that are eligible for use under a Basic license, as set forth at https://www.elastic.co/subscriptions, as may be modified by Elastic from time to time.

6.3 "Commercial Software" means the Elastic Software Source Code in any file containing a header stating the contents are subject to the Elastic License or which is contained in the repository folder labeled "x-pack", unless a LICENSE file present in the directory subtree declares a different license.

6.4 "Derivative Work of the Commercial Software" means, for purposes of this Agreement, any modification(s) or enhancement(s) to the Commercial Software, which represent, as a whole, an original work of authorship.

6.5 "License" means a limited, non-exclusive, non-transferable, fully paid up, royalty free, right and license, without the right to grant or authorize sublicenses, solely for Your internal business operations to (i) install and use the applicable Features and Functions of the Elastic Software in Object Code, and (ii) permit Contractors and Your Affiliates to use the Elastic software as set forth in (i) above, provided that such use by Contractors must be solely for Your benefit and/or the benefit of Your Affiliates, and You shall be responsible for all acts and omissions of such Contractors and Affiliates in connection with their use of the Elastic software that are contrary to the terms and conditions of this Agreement.

6.6 "License Key" means a sequence of bytes, including but not limited to a

JSON blob, that is used to enable certain features and functions of the
Elastic Software.

6.7 "Marks and Notices" means all Elastic trademarks, trade names, logos and
notices present on the Documentation as originally provided by Elastic.

6.8 "Non-production Environment" means an environment for development, testing
or quality assurance, where software is not used for production purposes.

6.9 "Object Code" means any form resulting from mechanical transformation or
translation of Source Code form, including but not limited to compiled object
code, generated documentation, and conversions to other media types.

6.10 "Source Code" means the preferred form of computer software for making
modifications, including but not limited to software source code,
documentation source, and configuration files.

6.11 "Subscription" means the right to receive Support Services and a License
to the Commercial Software.

<div align="center">

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

</div>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following

   boilerplate notice, with the fields enclosed by brackets "[]"

   replaced with your own identifying information. (Don't include

   the brackets!) The text should be enclosed in the appropriate

   comment syntax for the file format. We also recommend that a

   file or class name and description of purpose be included on the

   same "printed page" as the copyright notice for easier

   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Elasticsearch

Copyright 2009-2018 Elasticsearch

This product includes software developed by The Apache Software

Foundation (http://www.apache.org/).

Apache HttpComponents Core

Copyright 2005-2016 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

Apache HttpCore NIO

Copyright 2005-2016 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

Apache Commons Logging

Copyright 2003-2013 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache HttpComponents AsyncClient
Copyright 2010-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the

editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the

Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

This project contains annotations derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
Apache Commons Codec
Copyright 2002-2014 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

===

This project includes Public Suffix List copied from
<https://publicsuffix.org/list/effective_tld_names.dat>
licensed under the terms of the Mozilla Public License, v. 2.0

Full license text: <http://mozilla.org/MPL/2.0/>

Mozilla Public License Version 2.0
==================================

1. Definitions

1.1. "Contributor"
   means each individual or legal entity that creates, contributes to
   the creation of, or owns Covered Software.

1.2. "Contributor Version"
   means the combination of the Contributions of others (if any) used
   by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
   means Covered Software of a particular Contributor.

1.4. "Covered Software"
   means Source Code Form to which the initial Contributor has attached
   the notice in Exhibit A, the Executable Form of such Source Code
   Form, and Modifications of such Source Code Form, in each case
   including portions thereof.

1.5. "Incompatible With Secondary Licenses"
   means

   (a) that the initial Contributor has attached the notice described
       in Exhibit B to the Covered Software; or

   (b) that the Covered Software was made available under the terms of
       version 1.1 or earlier of the License, but not also under the
       terms of a Secondary License.

1.6. "Executable Form"
   means any form of the work other than Source Code Form.

1.7. "Larger Work"
   means a work that combines Covered Software with other material, in
   a separate file or files, that is not Covered Software.

1.8. "License"
   means this document.

1.9. "Licensable"
  means having the right to grant, to the maximum extent possible,
  whether at the time of the initial grant or subsequently, any and
  all of the rights conveyed by this License.

1.10. "Modifications"
  means any of the following:

  (a) any file in Source Code Form that results from an addition to,
      deletion from, or modification of the contents of Covered
      Software; or

  (b) any new file in Source Code Form that contains any Covered
      Software.

1.11. "Patent Claims" of a Contributor
  means any patent claim(s), including without limitation, method,
  process, and apparatus claims, in any patent Licensable by such
  Contributor that would be infringed, but for the grant of the
  License, by the making, using, selling, offering for sale, having
  made, import, or transfer of either its Contributions or its
  Contributor Version.

1.12. "Secondary License"
  means either the GNU General Public License, Version 2.0, the GNU
  Lesser General Public License, Version 2.1, the GNU Affero General
  Public License, Version 3.0, or any later versions of those
  licenses.

1.13. "Source Code Form"
  means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
  means an individual or a legal entity exercising rights under this
  License. For legal entities, "You" includes any entity that
  controls, is controlled by, or is under common control with You. For
  purposes of this definition, "control" means (a) the power, direct
  or indirect, to cause the direction or management of such entity,
  whether by contract or otherwise, or (b) ownership of more than
  fifty percent (50%) of the outstanding shares or beneficial
  ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its
Contributions are its original creation(s) or it has sufficient rights
to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under
applicable copyright doctrines of fair use, fair dealing, or other
equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted
in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any
Modifications that You create or to which You contribute, must be under
the terms of this License. You must inform recipients that the Source
Code Form of the Covered Software is governed by the terms of this
License, and how they can obtain a copy of this License. You may not
attempt to alter or restrict the recipients' rights in the Source Code
Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code
    Form, as described in Section 3.1, and You must inform recipients of
    the Executable Form how they can obtain a copy of such Source Code
    Form by reasonable means in a timely manner, at a charge no more
    than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this
    License, or sublicense it under different terms, provided that the
    license for the Executable Form does not attempt to limit or alter
    the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
************************************************************************
*                                                       *
*  6. Disclaimer of Warranty                            *
*  -------------------------                            *
*                                                       *
*  Covered Software is provided under this License on an "as is"    *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the     *
*  Covered Software is free of defects, merchantable, fit for a       *
*  particular purpose or non-infringing. The entire risk as to the    *
*  quality and performance of the Covered Software is with You.       *
*  Should any Covered Software prove defective in any respect, You    *
*  (not any Contributor) assume the cost of any necessary servicing,  *
*  repair, or correction. This disclaimer of warranty constitutes an  *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.        *
*                                                       *
************************************************************************


************************************************************************
*                                                       *
*  7. Limitation of Liability                           *
```

```
*  ------------------------                      *
*                                                *
*  Under no circumstances and under no legal theory, whether tort    *
*  (including negligence), contract, or otherwise, shall any         *
*  Contributor, or anyone who distributes Covered Software as        *
*  permitted above, be liable to You for any direct, indirect,       *
*  special, incidental, or consequential damages of any character    *
*  including, without limitation, damages for lost profits, loss of  *
*  goodwill, work stoppage, computer failure or malfunction, or any  *
*  and all other commercial damages or losses, even if such party    *
*  shall have been informed of the possibility of such damages. This *
*  limitation of liability shall not apply to liability for death or *
*  personal injury resulting from such party's negligence to the     *
*  extent applicable law prohibits such limitation. Some             *
*  jurisdictions do not allow the exclusion or limitation of         *
*  incidental or consequential damages, so this exclusion and        *
*  limitation may not apply to You.                                  *
*                                                *
**************************************************************************
```

8. Litigation

Any litigation relating to this License may be brought only in the
courts of a jurisdiction where the defendant maintains its principal
place of business and such litigation shall be governed by laws of that
jurisdiction, without reference to its conflict-of-law provisions.
Nothing in this Section shall prevent a party's ability to bring
cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. Any law or regulation which provides
that the language of a contract shall be construed against the drafter
shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section
10.3, no one other than the license steward has the right to modify or
publish new versions of this License. Each version will be given a

distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version
of the License under which You originally received the Covered Software,
or under the terms of any subsequent version published by the license
steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a
modified version of this License if you rename the license and remove
any references to the name of the license steward (except to note that
such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary
Licenses

If You choose to distribute Source Code Form that is Incompatible With
Secondary Licenses under the terms of this version of the License, the
notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice

 This Source Code Form is subject to the terms of the Mozilla Public
 License, v. 2.0. If a copy of the MPL was not distributed with this
 file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular
file, then You may include the notice in a location (such as a LICENSE
file in a relevant directory) where a recipient would be likely to look
for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice

 This Source Code Form is "Incompatible With Secondary Licenses", as
 defined by the Mozilla Public License, v. 2.0.
Apache HttpComponents Client
Copyright 1999-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
   distribution, then any Derivative Works that You distribute must
   include a readable copy of the attribution notices contained
   within such NOTICE file, excluding those notices that do not
   pertain to any part of the Derivative Works, in at least one
   of the following places: within a NOTICE text file distributed
   as part of the Derivative Works; within the Source form or
   documentation, if provided along with the Derivative Works; or,
   within a display generated by the Derivative Works, if and
   wherever such third-party notices normally appear. The contents
   of the NOTICE file are for informational purposes only and
   do not modify the License. You may add Your own attribution
   notices within Derivative Works that You distribute, alongside
   or as an addendum to the NOTICE text from the Work, provided
   that such additional attribution notices cannot be construed
   as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
Copyright (c) 2000-2015 The Legion of the Bouncy Castle Inc. (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software
and associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial
portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS

BE

LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
CONTRACT, TORT OR

OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER

DEALINGS IN THE SOFTWARE.

Format: https://www.debian.org/doc/packaging-manuals/copyright-format/1.0/

Copyright: Elasticsearch B.V. <info@elastic.co>

License: ${license.name}

${license.text}

== License

Copyright 2013-2018 Elasticsearch

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

== Copyright and License

This software is Copyright (c) 2013-2018 by Elasticsearch BV.

This is free software, licensed under The Apache License Version 2.0.

[role="xpack"]

[[license-settings]]

=== {xpack} License Settings

++++

<titleabbrev>License Settings</titleabbrev>

++++

You can configure this licensing setting in the `elasticsearch.yml` file.
For more information, see
{xpack-ref}/license-management.html[{xpack} License Management].

`xpack.license.self_generated.type`::
Set to `basic` (default) to enable basic {xpack} features. +
+
--
If set to `trial`, the self-generated license gives access only to all the features
of a x-pack for 30 days. You can later downgrade the cluster to a basic license if
needed.

--
[role="xpack"]
[testenv="basic"]
[[delete-license]]
=== Delete License API

This API enables you to delete licensing information.

[float]
==== Request

`DELETE /_xpack/license`

[float]
==== Description

When your license expires, {xpack} operates in a degraded mode. For more
information, see {xpack-ref}/license-expiration.html[License Expiration].

[float]
==== Authorization

You must have `manage` cluster privileges to use this API.
For more information, see
{xpack-ref}/security-privileges.html[Security Privileges].

[float]
==== Examples

The following example queries the info API:

[source,js]

DELETE _xpack/license

// CONSOLE
// TEST[skip:license testing issues]

When the license is successfully deleted, the API returns the following response:
[source,js]

{
 "acknowledged": true
}

// NOTCONSOLE
[role="xpack"]
[testenv="basic"]

[[get-license]]
=== Get License API

This API enables you to retrieve licensing information.

[float]
==== Request

`GET /_xpack/license`

[float]
==== Description

This API returns information about the type of license, when it was issued, and when it expires, for example.

For more information about the different types of licenses, see https://www.elastic.co/subscriptions.

[float]
==== Query Parameters

`local`::
 (boolean) Specifies whether to retrieve local information. The default value is `false`, which means the information is retrieved from the master node.

[float]
==== Authorization

You must have `monitor` cluster privileges to use this API.
For more information, see
{xpack-ref}/security-privileges.html[Security Privileges].

[float]
==== Examples

The following example provides information about a basic license:

[source,js]

GET _xpack/license

// CONSOLE

[source,js]

```
-------------------------------------------------
{
 "license" : {
   "status" : "active",
   "uid" : "cbff45e7-c553-41f7-ae4f-9205eabd80xx",
   "type" : "basic",
   "issue_date" : "2018-02-22T23:12:05.550Z",
   "issue_date_in_millis" : 1519341125550,
   "max_nodes" : 1000,
   "issued_to" : "test",
   "issuer" : "elasticsearch",
   "start_date_in_millis" : -1
 }
}
-------------------------------------------------
// TESTRESPONSE[s/"cbff45e7-c553-41f7-ae4f-9205eabd80xx"/$body.license.uid/]
// TESTRESPONSE[s/"basic"/$body.license.type/]
// TESTRESPONSE[s/"2018-02-22T23:12:05.550Z"/$body.license.issue_date/]
// TESTRESPONSE[s/1519341125550/$body.license.issue_date_in_millis/]
// TESTRESPONSE[s/1000/$body.license.max_nodes/]
// TESTRESPONSE[s/"test"/$body.license.issued_to/]
// TESTRESPONSE[s/"elasticsearch"/$body.license.issuer/]
[role="xpack"]
[testenv="basic"]
[[update-license]]
=== Update License API
```

This API enables you to update your license.

```
[float]
==== Request
```

`PUT _xpack/license`

```
[float]
==== Description
```

You can update your license at runtime without shutting down your nodes.
License updates take effect immediately. If the license you are installing does
not support all of the features that were available with your previous license,
however, you are notified in the response. You must then re-submit the API
request with the `acknowledge` parameter set to `true`.

For more information about the different types of licenses, see
https://www.elastic.co/subscriptions.

```
[float]
==== Query Parameters
```

`acknowledge`::
 (boolean) Specifies whether you acknowledge the license changes. The default
 value is `false`.

[float]
==== Request Body

`licenses`::
 (array) A sequence of one or more JSON documents containing the license
 information.

[float]
==== Authorization

If {security} is enabled, you need `manage` cluster privileges to install the
license.

If {security} is enabled and you are installing a gold or platinum license, you
must enable TLS on the transport networking layer before you install the license.
See <<configuring-tls>>.

[float]
==== Examples

The following example updates to a basic license:

[source,js]
```
--------------------------------------------------------
POST _xpack/license
{
 "licenses": [
  {
    "uid":"893361dc-9749-4997-93cb-802e3d7fa4xx",
    "type":"basic",
    "issue_date_in_millis":1411948800000,
    "expiry_date_in_millis":1914278399999,
    "max_nodes":1,
    "issued_to":"issuedTo",
    "issuer":"issuer",
    "signature":"xx"
  }
  ]
}
--------------------------------------------------------
// CONSOLE
// TEST[skip:license testing issues]
```

NOTE: These values are invalid; you must substitute the appropriate content
from your license file.

You can alternatively use a `curl` command, for example:

[source,js]
[source,shell]
```
-----------------------------------------------------------
curl -XPUT -u <user> 'http://<host>:<port>/_xpack/license' -H "Content-Type: application/json" -d @license.json
-----------------------------------------------------------
```
// NOTCONSOLE

On Windows machine, use the following command:

[source,shell]
```
-----------------------------------------------------------
gc .\license.json | Invoke-WebRequest -uri http://<host>:<port>/_xpack/license -Credential elastic -Method Put -
ContentType "application/json"
-----------------------------------------------------------
```

In these examples,

* `<user>` is a user ID with the appropriate authority.
* `<host>` is the hostname of the {es} node (`localhost` if executing
 locally)
* `<port>` is the http port (defaults to `9200`)
* `license.json` is the license JSON file

NOTE: If your {es} node has SSL enabled on the HTTP interface, you must
 start your URL with `https://`

If you previously had a license with more features than the basic license, you
receive the following response:

[source,js]
```
-----------------------------------------------------------
 {
 "acknowledged": false,
 "license_status": "valid",
 "acknowledge": {
   "message": """This license update requires acknowledgement. To acknowledge the license, please read the
following messages and update the license again, this time with the "acknowledge=true" parameter:""",
   "watcher": [
     "Watcher will be disabled"
   ],
   "logstash": [
     "Logstash will no longer poll for centrally-managed pipelines"
```

```
    ],
    "security": [
      "The following X-Pack security functionality will be disabled: ..." ]
    }
}
--------------------------------------------------------
// NOTCONSOLE

To complete the update, you must re-submit the API request and set the
`acknowledge` parameter to `true`. For example:

[source,js]
--------------------------------------------------------
POST _xpack/license?acknowledge=true
{
 "licenses": [
   {
     "uid":"893361dc-9749-4997-93cb-802e3d7fa4xx",
     "type":"basic",
     "issue_date_in_millis":1411948800000,
     "expiry_date_in_millis":1914278399999,
     "max_nodes":1,
     "issued_to":"issuedTo",
     "issuer":"issuer",
     "signature":"xx"
   }
   ]
}
--------------------------------------------------------
// CONSOLE
// TEST[skip:license testing issues]

Alternatively:

[source,sh]
--------------------------------------------------------
curl -XPUT -u elastic 'http://<host>:<port>/_xpack/license?acknowledge=true' -H "Content-Type: application/json"
-d @license.json
--------------------------------------------------------
// NOTCONSOLE
```

For more information about the features that are disabled when you downgrade
your license, see {xpack-ref}/license-expiration.html[License Expiration].
This package is free to use under the Elastic license. It contains open source
and free commercial features and access to paid commercial features.
{stack-ov}/license-management.html[Start a 30-day trial] to try out all of the
paid commercial features. See the
https://www.elastic.co/subscriptions[Subscriptions] page for information about

Elastic license levels.

The Netty Project
==================

Please visit the Netty web site for more information:

 * http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

 * LICENSE:
   * license/LICENSE.jsr166y.txt (Public Domain)
 * HOMEPAGE:
   * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
   * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

 * LICENSE:
   * license/LICENSE.base64.txt (Public Domain)
 * HOMEPAGE:
   * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified version of 'JZlib', a re-implementation of
zlib in pure Java, which can be obtained at:

* LICENSE:
  * license/LICENSE.jzlib.txt (BSD Style License)
 * HOMEPAGE:
  * http://www.jcraft.com/jzlib/

This product contains a modified version of 'Webbit', a Java event based
WebSocket and HTTP server:

 * LICENSE:
  * license/LICENSE.webbit.txt (BSD License)
 * HOMEPAGE:
  * https://github.com/joewalnes/webbit

This product optionally depends on 'Protocol Buffers', Google's data
interchange format, which can be obtained at:

 * LICENSE:
  * license/LICENSE.protobuf.txt (New BSD License)
 * HOMEPAGE:
  * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate
a temporary self-signed X.509 certificate when the JVM does not provide the
equivalent functionality. It can be obtained at:

 * LICENSE:
  * license/LICENSE.bouncycastle.txt (MIT License)
 * HOMEPAGE:
  * http://www.bouncycastle.org/

This product optionally depends on 'SLF4J', a simple logging facade for Java,
which can be obtained at:

 * LICENSE:
  * license/LICENSE.slf4j.txt (MIT License)
 * HOMEPAGE:
  * http://www.slf4j.org/

This product optionally depends on 'Apache Commons Logging', a logging
framework, which can be obtained at:

 * LICENSE:
  * license/LICENSE.commons-logging.txt (Apache License 2.0)
 * HOMEPAGE:
  * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework,
which can be obtained at:

* LICENSE:
  * license/LICENSE.log4j.txt (Apache License 2.0)
 * HOMEPAGE:
  * http://logging.apache.org/log4j/

This product optionally depends on 'JBoss Logging', a logging framework,
which can be obtained at:

 * LICENSE:
  * license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
 * HOMEPAGE:
  * http://anonsvn.jboss.org/repos/common/common-logging-spi/

This product optionally depends on 'Apache Felix', an open source OSGi
framework implementation, which can be obtained at:

 * LICENSE:
  * license/LICENSE.felix.txt (Apache License 2.0)
 * HOMEPAGE:
  * http://felix.apache.org/
Apache Lucene
Copyright 2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and

http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
 http://snowball.tartarus.org/

The KStem stemmer in
 analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java

analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

 mecab-ipadic-2.7.0-20070801

which can be obtained from

 http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

==

mecab-ipadic-2.7.0-20070801 Notice

==

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the

development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.
Copyright (c) 2012 France Tlcom
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

---

the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

/*
* Copyright 2001-2004 Unicode, Inc.
*
* Disclaimer

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/. Here is the copyright from those sources:

```
*    derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/
```

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

```
# Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
# Permission is hereby granted, free of charge, to any person
# obtaining a copy of this software and associated documentation
# files (the "Software"), to deal in the Software without
# restriction, including without limitation the rights to use,
# copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the
# Software is furnished to do so, subject to the following
# conditions:
#
# The above copyright notice and this permission notice shall be
# included in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
# OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
# WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
# OTHER DEALINGS IN THE SOFTWARE.
```

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

```
/*
* Copyright (C) 1999-2010, International Business Machines
```

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license applies to the KStemmer:

The following license applies to the Morfologik project:

are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

  * Neither the name of Morfologik nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from
Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and
is licenced on the terms of (inter alia) LGPL and Creative Commons
ShareAlike. The part-of-speech tags were added in Morfologik project and
are not found in the data from sjp.pl. The tagset is similar to IPI PAN
tagset.

The following license applies to the Morfeusz project,
used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
      Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are

met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the
   distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2010 RightTime, Inc.

 Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License.

                 Apache License
             Version 2.0, January 2004
           http://www.apache.org/licenses/

  TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
    and distribution as defined by Sections 1 through 9 of this document.

    "Licensor" shall mean the copyright owner or entity authorized by
    the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2010-2013, Carrot Search s.c., Boznicza 11/56, Poznan, Poland

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Eclipse Foundation Software User Agreement

April 9, 2014

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION
AND/OR OTHER MATERIALS FOR OPEN SOURCE
PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND
CONDITIONS OF THIS AGREEMENT AND/OR
THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR
REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE
THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND
CONDITIONS OF ANY APPLICABLE LICENSE
AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE
TERMS AND CONDITIONS OF THIS AGREEMENT
AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES
INDICATED OR REFERENCED BELOW, THEN YOU MAY
NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and
conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is
also available at http://www.eclipse.org/legal/epl-v10.html. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the
Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made available as
downloadable archives ("Downloads").

*   Content may be structured and packaged into modules to facilitate delivering, extending, and upgrading the Content.
    Typical modules may include plug-ins ("Plug-ins"), plug-in fragments ("Fragments"), and features ("Features").
*   Each Plug-in or Fragment may be packaged as a sub-directory or JAR (Java ARchive) in a directory named "plugins".
*   A Feature is a bundle of one or more Plug-ins and/or Fragments and associated material. Each Feature may be packaged
    as a sub-directory in a directory named "features". Within a Feature, files named "feature.xml" may contain a list
    of the names and version numbers of the Plug-ins and/or Fragments associated with that Feature.
*   Features may also include other Features ("Included Features"). Within a Feature, files named "feature.xml" may
    contain a list of the names and version numbers of Included Features.

The terms and conditions governing Plug-ins and Fragments should be contained in files named "about.html"
("Abouts").
The terms and conditions governing Features and Included Features should be contained in files named "license.html"
("Feature Licenses"). Abouts and Feature Licenses may be located in any directory of a Download or Module including, but
not limited to the following locations:

* The top-level (root) directory
* Plug-in and Fragment directories
* Inside Plug-ins and Fragments packaged as JARs
* Sub-directories of the directory named "src" of certain Plug-ins
* Feature directories

Note: if a Feature made available by the Eclipse Foundation is installed using the Provisioning Technology (as defined
below), you must agree to a license ("Feature Update License") during the installation process. If the Feature contains
Included Features, the Feature Update License should either provide you with the terms and conditions governing the
Included Features or inform you where you can locate them. Feature Update Licenses may be found in the "license"
property of files named "feature.properties" found within a Feature. Such Abouts, Feature Licenses, and Feature Update
Licenses contain the terms and conditions (or references to such terms and conditions) that govern your use of the
associated Content in that directory.

THE ABOUTS, FEATURE LICENSES, AND FEATURE UPDATE LICENSES MAY REFER TO THE EPL OR OTHER LICENSE AGREEMENTS, NOTICES OR
TERMS AND CONDITIONS. SOME OF THESE OTHER LICENSE AGREEMENTS MAY INCLUDE (BUT ARE NOT LIMITED TO):

* Eclipse Distribution License Version 1.0 (available at http://www.eclipse.org/licenses/edl-v10.html)
* Common Public License Version 1.0 (available at http://www.eclipse.org/legal/cpl-v10.html)
* Apache Software License 1.1 (available at http://www.apache.org/licenses/LICENSE)
* Apache Software License 2.0 (available at http://www.apache.org/licenses/LICENSE-2.0)
* Mozilla Public License Version 1.1 (available at http://www.mozilla.org/MPL/MPL-1.1.html)

IT IS YOUR OBLIGATION TO READ AND ACCEPT ALL SUCH TERMS AND CONDITIONS PRIOR TO USE OF THE CONTENT. If no About, Feature
License, or Feature Update License is provided, please contact the Eclipse Foundation to determine what terms and
conditions govern that particular Content.

Use of Provisioning Technology

The Eclipse Foundation makes available provisioning software, examples of which include, but are not limited to, p2 and
the Eclipse Update Manager ("Provisioning Technology") for the purpose of allowing users to install software,
documentation, information and/or other materials (collectively "Installable Software"). This capability is provided
with the intent of allowing such users to install, extend and update Eclipse-based products. Information about packaging
Installable Software is available at http://eclipse.org/equinox/p2/repository_packaging.html ("Specification").

You may use Provisioning Technology to allow other parties to install Installable Software. You shall be responsible for
enabling the applicable license agreements relating to the Installable Software to be presented to, and accepted by,

the
users of the Provisioning Technology in accordance with the Specification. By using Provisioning Technology in such a
manner and making it available in accordance with the Specification, you further acknowledge your agreement to, and the
acquisition of all necessary rights to permit the following:

1. A series of actions may occur ("Provisioning Process") in which a user may execute the Provisioning Technology on a
   machine ("Target Machine") with the intent of installing, extending or updating the functionality of an
   Eclipse-based product.
2. During the Provisioning Process, the Provisioning Technology may cause third party Installable Software or a portion
   thereof to be accessed and copied to the Target Machine.
3. Pursuant to the Specification, you will provide to the user the terms and conditions that govern the use of the
   Installable Software ("Installable Software Agreement") and such Installable Software Agreement shall be accessed
   from the Target Machine in accordance with the Specification. Such Installable Software Agreement must inform the
   user of the terms and conditions that govern the Installable Software and must solicit acceptance by the end user in
   the manner prescribed in such Installable Software Agreement. Upon such indication of agreement by the user, the
   provisioning Technology will complete installation of the Installable Software.

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import,
possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software,
please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of
encryption software, to see if this is permitted.

Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
The code in this repository code was Written by Gil Tene, Michael Barker,
and Matt Warren, and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

For users of this code who wish to consume it under the "BSD" license
rather than under the public domain or CC0 contribution text mentioned
above, the code found under this directory is *also* provided under the
following license (commonly referred to as the BSD 2-Clause License). This
license does not detract from the above stated release of the code into
the public domain, and simply represents an additional license granted by
the Author.

** Beginning of "BSD 2-Clause License" text. **

Copyright (c) 2012, 2013, 2014 Gil Tene
Copyright (c) 2014 Michael Barker
Copyright (c) 2014 Matt Warren
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

                          Apache License
                    Version 2.0, January 2004
                  http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Apache log4j

Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache License
                    Version 2.0, January 2004
                    http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

  "License" shall mean the terms and conditions for use, reproduction,
  and distribution as defined by Sections 1 through 9 of this document.

  "Licensor" shall mean the copyright owner or entity authorized by
  the copyright owner that is granting the License.

  "Legal Entity" shall mean the union of the acting entity and all
  other entities that control, are controlled by, or are under common
  control with that entity. For the purposes of this definition,
  "control" means (i) the power, direct or indirect, to cause the
  direction or management of such entity, whether by contract or
  otherwise, or (ii) ownership of fifty percent (50%) or more of the
  outstanding shares, or (iii) beneficial ownership of such entity.

  "You" (or "Your") shall mean an individual or Legal Entity
  exercising permissions granted by this License.

  "Source" form shall mean the preferred form for making modifications,
  including but not limited to software source code, documentation
  source, and configuration files.

  "Object" form shall mean any form resulting from mechanical
  transformation or translation of a Source form, including but
  not limited to compiled object code, generated documentation,
  and conversions to other media types.

  "Work" shall mean the work of authorship, whether in Source or
  Object form, made available under the License, as indicated by a
  copyright notice that is included in or attached to the work
  (an example is provided in the Appendix below).

  "Derivative Works" shall mean any work, whether in Source or Object
  form, that is based on (or derived from) the Work and for which the
  editorial revisions, annotations, elaborations, or other modifications
  represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
    distribution, then any Derivative Works that You distribute must
    include a readable copy of the attribution notices contained
    within such NOTICE file, excluding those notices that do not
    pertain to any part of the Derivative Works, in at least one
    of the following places: within a NOTICE text file distributed
    as part of the Derivative Works; within the Source form or
    documentation, if provided along with the Derivative Works; or,
    within a display generated by the Derivative Works, if and
    wherever such third-party notices normally appear. The contents
    of the NOTICE file are for informational purposes only and
    do not modify the License. You may add Your own attribution
    notices within Derivative Works that You distribute, alongside
    or as an addendum to the NOTICE text from the Work, provided
    that such additional attribution notices cannot be construed
    as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

About This Content

May 22, 2015

License

The Eclipse Foundation makes available all content in this plug-in ("Content"). Unless otherwise indicated below, the
Content is provided to you under the terms and conditions of the Apache License, Version 2.0. A copy of the Apache
License, Version 2.0 is available at http://www.apache.org/licenses/LICENSE-2.0.txt

If you did not receive this Content directly from the Eclipse Foundation, the Content is being redistributed by another
party ("Redistributor") and different terms and conditions may apply to your use of any object code in the Content.
Check the Redistributors license that was provided with the Content. If no such license exists, contact the
Redistributor. Unless otherwise indicated below, the terms and conditions of the Apache License, Version 2.0 still apply
to any source code in the Content and such source code may be obtained at
http://www.eclipse.org](http://www.eclipse.org.
The code for the t-digest was originally authored by Ted Dunning

A number of small but very helpful changes have been contributed by Adrien Grand (https://github.com/jpountz)
===
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
===
This product includes software developed by
Joda.org (http://www.joda.org/).
// --
// Transitive dependencies of this project determined from the
// maven pom organized by organization.
// --

Apache Extras for Apache log4j.

From: 'an unknown organization'
 - geronimo-jms_1.1_spec org.apache.geronimo.specs:geronimo-jms_1.1_spec:jar:1.0

From: 'Apache Software Foundation' (http://www.apache.org)
 - Apache Log4j (http://logging.apache.org/log4j/1.2/) log4j:log4j:bundle:1.2.17
   License: The Apache Software License, Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0.txt)
ACKNOWLEDGEMENT

===============

HPPC borrowed code, ideas or both from:

* Apache Lucene, http://lucene.apache.org/
  (Apache license)
* Fastutil, http://fastutil.di.unimi.it/
  (Apache license)
* Koloboke, https://github.com/OpenHFT/Koloboke
  (Apache license)
Eclipse Distribution License - v 1.0

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors.

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

 Redistributions of source code must retain the above copyright notice, this
 list of conditions and the following disclaimer.

 Redistributions in binary form must reproduce the above copyright notice,
 this list of conditions and the following disclaimer in the documentation
 and/or other materials provided with the distribution.

 Neither the name of the Eclipse Foundation, Inc. nor the names of its
 contributors may be used to endorse or promote products derived from this
 software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

```
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/

/**
* Licensing for xpack.
*
* A {@link org.elasticsearch.license.License} is a signed set of json properties that determine what features
* are available in a running cluster. Licenses are registered through a
* {@link org.elasticsearch.license.PutLicenseRequest}. This action is handled by the master node, which places
* the signed license into the cluster state. Each node listens for cluster state updates via the
* {@link org.elasticsearch.license.LicenseService}, and updates its local copy of the license when it detects
* changes in the cluster state.
*
* The logic for which features are available given the current license is handled by
* {@link org.elasticsearch.license.XPackLicenseState}, which is updated by the
* {@link org.elasticsearch.license.LicenseService} when the license changes.
*/
package org.elasticsearch.license;

/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.MasterNodeReadOperationRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

public class GetLicenseRequestBuilder extends MasterNodeReadOperationRequestBuilder<GetLicenseRequest,
GetLicenseResponse,
    GetLicenseRequestBuilder> {

  public GetLicenseRequestBuilder(ElasticsearchClient client) {
    this(client, GetLicenseAction.INSTANCE);
  }

  /**
   * Creates new get licenses request builder
   *
   * @param client elasticsearch client
   */
  public GetLicenseRequestBuilder(ElasticsearchClient client, GetLicenseAction action) {
    super(client, action, new GetLicenseRequest());
```

```
  }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.AcknowledgedRequest;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;


public class DeleteLicenseRequest extends AcknowledgedRequest<DeleteLicenseRequest> {

  public DeleteLicenseRequest() {
  }

  @Override
  public ActionRequestValidationException validate() {
    return null;
  }

  @Override
  public void readFrom(StreamInput in) throws IOException {
    super.readFrom(in);
  }

  @Override
  public void writeTo(StreamOutput out) throws IOException {
    super.writeTo(out);
  }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.Action;

public class GetBasicStatusAction extends Action<GetBasicStatusResponse> {
```

```java
    public static final GetBasicStatusAction INSTANCE = new GetBasicStatusAction();
    public static final String NAME = "cluster:admin/xpack/license/basic_status";

    private GetBasicStatusAction() {
        super(NAME);
    }

    @Override
    public GetBasicStatusResponse newResponse() {
        return new GetBasicStatusResponse();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedResponse;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentBuilder;

import java.io.IOException;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

public class PutLicenseResponse extends AcknowledgedResponse {

    private LicensesStatus status;
    private Map<String, String[]> acknowledgeMessages;
    private String acknowledgeHeader;

    PutLicenseResponse() {
    }

    public PutLicenseResponse(boolean acknowledged, LicensesStatus status) {
        this(acknowledged, status, null, Collections.<String, String[]>emptyMap());
    }

    public PutLicenseResponse(boolean acknowledged, LicensesStatus status, String acknowledgeHeader,
                Map<String, String[]> acknowledgeMessages) {
        super(acknowledged);
        this.status = status;
        this.acknowledgeHeader = acknowledgeHeader;
```

```java
        this.acknowledgeMessages = acknowledgeMessages;
    }

    public LicensesStatus status() {
        return status;
    }

    public Map<String, String[]> acknowledgeMessages() {
        return acknowledgeMessages;
    }

    public String acknowledgeHeader() {
        return acknowledgeHeader;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        status = LicensesStatus.fromId(in.readVInt());
        acknowledgeHeader = in.readOptionalString();
        int size = in.readVInt();
        Map<String, String[]> acknowledgeMessages = new HashMap<>(size);
        for (int i = 0; i < size; i++) {
            String feature = in.readString();
            int nMessages = in.readVInt();
            String[] messages = new String[nMessages];
            for (int j = 0; j < nMessages; j++) {
                messages[j] = in.readString();
            }
            acknowledgeMessages.put(feature, messages);
        }
        this.acknowledgeMessages = acknowledgeMessages;
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        super.writeTo(out);
        out.writeVInt(status.id());
        out.writeOptionalString(acknowledgeHeader);
        out.writeVInt(acknowledgeMessages.size());
        for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
            out.writeString(entry.getKey());
            out.writeVInt(entry.getValue().length);
            for (String message : entry.getValue()) {
                out.writeString(message);
            }
        }
    }
```

```java
    @Override
    protected void addCustomFields(XContentBuilder builder, Params params) throws IOException {
        switch (status) {
            case VALID:
                builder.field("license_status", "valid");
                break;
            case INVALID:
                builder.field("license_status", "invalid");
                break;
            case EXPIRED:
                builder.field("license_status", "expired");
                break;
            default:
                throw new IllegalArgumentException("unknown status [" + status + "] found");
        }
        if (!acknowledgeMessages.isEmpty()) {
            builder.startObject("acknowledge");
            builder.field("message", acknowledgeHeader);
            for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
                builder.startArray(entry.getKey());
                for (String message : entry.getValue()) {
                    builder.value(message);
                }
                builder.endArray();
            }
            builder.endObject();
        }
    }

    @Override
    public String toString() {
        return Strings.toString(this, true, true);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.action.RestToXContentListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;
```

```java
import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.POST;
import static org.elasticsearch.rest.RestRequest.Method.PUT;

public class RestPutLicenseAction extends XPackRestHandler {

    public RestPutLicenseAction(Settings settings, RestController controller) {
        super(settings);
        controller.registerHandler(POST, URI_BASE + "/license", this);
        controller.registerHandler(PUT, URI_BASE + "/license", this);
    }

    @Override
    public String getName() {
        return "xpack_put_license_action";
    }

    @Override
    public RestChannelConsumer doPrepareRequest(final RestRequest request, final XPackClient client) throws
IOException {
        if (request.hasContent() == false) {
            throw new IllegalArgumentException("The license must be provided in the request body");
        }
        PutLicenseRequest putLicenseRequest = new PutLicenseRequest();
        putLicenseRequest.license(request.content(), request.getXContentType());
        putLicenseRequest.acknowledge(request.paramAsBoolean("acknowledge", false));
        putLicenseRequest.timeout(request.paramAsTime("timeout", putLicenseRequest.timeout()));
        putLicenseRequest.masterNodeTimeout(request.paramAsTime("master_timeout",
putLicenseRequest.masterNodeTimeout()));

        if ("basic".equals(putLicenseRequest.license().type())) {
            throw new IllegalArgumentException("Installing basic licenses is no longer allowed. Use the POST " +
                    "/_xpack/license/start_basic API to install a basic license that does not expire.");
        }

        return channel -> client.es().admin().cluster().execute(PutLicenseAction.INSTANCE, putLicenseRequest,
                new RestToXContentListener<>(channel));
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;
```

```java
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;
import java.util.Map;

import static org.elasticsearch.rest.RestRequest.Method.POST;

public class RestPostStartTrialLicense extends XPackRestHandler {

    RestPostStartTrialLicense(Settings settings, RestController controller) {
        super(settings);
        controller.registerHandler(POST, URI_BASE + "/license/start_trial", this);
    }

    @Override
    protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
    {
        PostStartTrialRequest startTrialRequest = new PostStartTrialRequest();
        startTrialRequest.setType(request.param("type", "trial"));
        startTrialRequest.acknowledge(request.paramAsBoolean("acknowledge", false));
        return channel -> client.licensing().postStartTrial(startTrialRequest,
            new RestBuilderListener<PostStartTrialResponse>(channel) {
                @Override
                public RestResponse buildResponse(PostStartTrialResponse response, XContentBuilder builder) throws
Exception {
                    PostStartTrialResponse.Status status = response.getStatus();
                    builder.startObject();
                    builder.field("acknowledged", startTrialRequest.isAcknowledged());
                    if (status.isTrialStarted()) {
                        builder.field("trial_was_started", true);
                        builder.field("type", startTrialRequest.getType());
                    } else {
                        builder.field("trial_was_started", false);
                        builder.field("error_message", status.getErrorMessage());
                    }

                    Map<String, String[]> acknowledgementMessages = response.getAcknowledgementMessages();
                    if (acknowledgementMessages.isEmpty() == false) {
                        builder.startObject("acknowledge");
                        builder.field("message", response.getAcknowledgementMessage());
```

```java
                    for (Map.Entry<String, String[]> entry : acknowledgementMessages.entrySet()) {
                        builder.startArray(entry.getKey());
                        for (String message : entry.getValue()) {
                            builder.value(message);
                        }
                        builder.endArray();
                    }
                    builder.endObject();
                }
                builder.endObject();
                return new BytesRestResponse(status.getRestStatus(), builder);
            }
        });
    }

    @Override
    public String getName() {
        return "xpack_upgrade_to_trial_action";
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

class PostStartBasicRequestBuilder extends ActionRequestBuilder<PostStartBasicRequest,
PostStartBasicResponse> {

    PostStartBasicRequestBuilder(ElasticsearchClient client, PostStartBasicAction action) {
        super(client, action, new PostStartBasicRequest());
    }

    public PostStartBasicRequestBuilder setAcknowledge(boolean acknowledge) {
        request.acknowledge(acknowledge);
        return this;
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;
```

```java
import org.elasticsearch.action.support.master.AcknowledgedResponse;

public class DeleteLicenseResponse extends AcknowledgedResponse {

  DeleteLicenseResponse() {
  }

  DeleteLicenseResponse(boolean acknowledged) {
    super(acknowledged);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.RestStatus;

import java.io.IOException;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

import static org.elasticsearch.license.PostStartBasicResponse.Status.NEED_ACKNOWLEDGEMENT;

class PostStartBasicResponse extends AcknowledgedResponse {

  private Map<String, String[]> acknowledgeMessages;
  private String acknowledgeMessage;

  enum Status {
    GENERATED_BASIC(true, null, RestStatus.OK),
    ALREADY_USING_BASIC(false, "Operation failed: Current license is basic.", RestStatus.FORBIDDEN),
    NEED_ACKNOWLEDGEMENT(false, "Operation failed: Needs acknowledgement.", RestStatus.OK);

    private final boolean isBasicStarted;
    private final String errorMessage;
    private final RestStatus restStatus;

    Status(boolean isBasicStarted, String errorMessage, RestStatus restStatus) {
      this.isBasicStarted = isBasicStarted;
```

```java
            this.errorMessage = errorMessage;
            this.restStatus = restStatus;
        }

        boolean isBasicStarted() {
            return isBasicStarted;
        }

        String getErrorMessage() {
            return errorMessage;
        }

        RestStatus getRestStatus() {
            return restStatus;
        }
    }

    private Status status;

    PostStartBasicResponse() {
    }

    PostStartBasicResponse(Status status) {
        this(status, Collections.emptyMap(), null);
    }

    PostStartBasicResponse(Status status, Map<String, String[]> acknowledgeMessages, String
acknowledgeMessage) {
        super(status != NEED_ACKNOWLEDGEMENT);
        this.status = status;
        this.acknowledgeMessages = acknowledgeMessages;
        this.acknowledgeMessage = acknowledgeMessage;
    }

    public Status getStatus() {
        return status;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        status = in.readEnum(Status.class);
        acknowledgeMessage = in.readOptionalString();
        int size = in.readVInt();
        Map<String, String[]> acknowledgeMessages = new HashMap<>(size);
        for (int i = 0; i < size; i++) {
            String feature = in.readString();
            int nMessages = in.readVInt();
```

```java
        String[] messages = new String[nMessages];
        for (int j = 0; j < nMessages; j++) {
            messages[j] = in.readString();
        }
        acknowledgeMessages.put(feature, messages);
    }
    this.acknowledgeMessages = acknowledgeMessages;
}

@Override
public void writeTo(StreamOutput out) throws IOException {
    super.writeTo(out);
    out.writeEnum(status);
    out.writeOptionalString(acknowledgeMessage);
    out.writeVInt(acknowledgeMessages.size());
    for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
        out.writeString(entry.getKey());
        out.writeVInt(entry.getValue().length);
        for (String message : entry.getValue()) {
            out.writeString(message);
        }
    }
}

@Override
protected void addCustomFields(XContentBuilder builder, Params params) throws IOException {
    if (status.isBasicStarted()) {
        builder.field("basic_was_started", true);
    } else {
        builder.field("basic_was_started", false);
        builder.field("error_message", status.getErrorMessage());
    }
    if (acknowledgeMessages.isEmpty() == false) {
        builder.startObject("acknowledge");
        builder.field("message", acknowledgeMessage);
        for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
            builder.startArray(entry.getKey());
            for (String message : entry.getValue()) {
                builder.value(message);
            }
            builder.endArray();
        }
        builder.endObject();
    }
}
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
```

```
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ack.ClusterStateUpdateResponse;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportDeleteLicenseAction extends TransportMasterNodeAction<DeleteLicenseRequest,
DeleteLicenseResponse> {

    private final LicenseService licenseService;

    @Inject
    public TransportDeleteLicenseAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                          LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                          IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, DeleteLicenseAction.NAME, transportService, clusterService, threadPool, actionFilters,
            indexNameExpressionResolver, DeleteLicenseRequest::new);
        this.licenseService = licenseService;
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.MANAGEMENT;
    }

    @Override
    protected DeleteLicenseResponse newResponse() {
        return new DeleteLicenseResponse();
    }

    @Override
    protected ClusterBlockException checkBlock(DeleteLicenseRequest request, ClusterState state) {
        return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
```

```java
        }

        @Override
        protected void masterOperation(final DeleteLicenseRequest request, ClusterState state, final
ActionListener<DeleteLicenseResponse>
                listener) throws ElasticsearchException {
            licenseService.removeLicense(request, new ActionListener<ClusterStateUpdateResponse>() {
                @Override
                public void onResponse(ClusterStateUpdateResponse clusterStateUpdateResponse) {
                    listener.onResponse(new DeleteLicenseResponse(clusterStateUpdateResponse.isAcknowledged()));
                }

                @Override
                public void onFailure(Exception e) {
                    listener.onFailure(e);
                }
            });
        }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;


import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.apache.logging.log4j.util.Supplier;
import org.apache.lucene.util.BytesRef;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.watcher.FileChangesListener;
import org.elasticsearch.watcher.FileWatcher;
import org.elasticsearch.watcher.ResourceWatcherService;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.concurrent.atomic.AtomicBoolean;

/**
 * File based watcher for license {@link OperationMode}
 * Watches for changes in <code>licenseModePath</code>, use
 * {@link #getCurrentOperationMode()} to access the latest mode
 *
 * In case of failure to read a valid operation mode from <code>licenseModePath</code>,
 * the operation mode will default to PLATINUM
```

```java
*/
public final class OperationModeFileWatcher implements FileChangesListener {
  private final ResourceWatcherService resourceWatcherService;
  private final Path licenseModePath;
  private final AtomicBoolean initialized = new AtomicBoolean();
  private final OperationMode defaultOperationMode = OperationMode.PLATINUM;
  private volatile OperationMode currentOperationMode = defaultOperationMode;
  private final Logger logger;
  private final Runnable onChange;

  public OperationModeFileWatcher(ResourceWatcherService resourceWatcherService, Path licenseModePath,
                    Logger logger, Runnable onChange) {
    this.resourceWatcherService = resourceWatcherService;
    this.licenseModePath = licenseModePath;
    this.logger = logger;
    this.onChange = onChange;
  }

  public void init() {
    if (initialized.compareAndSet(false, true)) {
      final FileWatcher watcher = new FileWatcher(licenseModePath);
      watcher.addListener(this);
      try {
        resourceWatcherService.add(watcher, ResourceWatcherService.Frequency.HIGH);
        if (Files.exists(licenseModePath)) {
          onChange(licenseModePath);
        }
      } catch (IOException e) {
        logger.error("couldn't initialize watching license mode file", e);
      }
    }
  }

  /**
   * Returns the current operation mode based on license mode file.
   * Defaults to {@link OperationMode#PLATINUM}
   */
  public OperationMode getCurrentOperationMode() {
    return currentOperationMode;
  }

  @Override
  public void onFileInit(Path file) {
    onChange(file);
  }

  @Override
  public void onFileCreated(Path file) {
```

```java
        onChange(file);
    }

    @Override
    public void onFileDeleted(Path file) {
        onChange(file);
    }

    @Override
    public void onFileChanged(Path file) {
        onChange(file);
    }

    private synchronized void onChange(Path file) {
        if (file.equals(licenseModePath)) {
            OperationMode newOperationMode = defaultOperationMode;
            try {
                if (Files.exists(licenseModePath)
                        && Files.isReadable(licenseModePath)) {
                    final byte[] content;
                    try {
                        content = Files.readAllBytes(licenseModePath);
                    } catch (IOException e) {
                        logger.error(
                            (Supplier<?>) () -> new ParameterizedMessage(
                                "couldn't read operation mode from [{}]", licenseModePath.toAbsolutePath()), e);
                        return;
                    }
                    // this UTF-8 conversion is much pickier than java String
                    final String operationMode = new BytesRef(content).utf8ToString();
                    try {
                        newOperationMode = OperationMode.resolve(operationMode);
                    } catch (IllegalArgumentException e) {
                        logger.error(
                            (Supplier<?>) () -> new ParameterizedMessage(
                                "invalid operation mode in [{}]", licenseModePath.toAbsolutePath()), e);
                        return;
                    }
                }
            } finally {
                // set this after the fact to prevent that we are jumping back and forth first setting to defautl and then
reading the
                // actual op mode resetting it.
                this.currentOperationMode = newOperationMode;
            }
            onChange.run();
        }
    }
```

```
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

class GetTrialStatusResponse extends ActionResponse {

  private boolean eligibleToStartTrial;

  GetTrialStatusResponse() {
  }

  GetTrialStatusResponse(boolean eligibleToStartTrial) {
     this.eligibleToStartTrial = eligibleToStartTrial;
  }

  boolean isEligibleToStartTrial() {
     return eligibleToStartTrial;
  }

  @Override
  public void readFrom(StreamInput in) throws IOException {
     eligibleToStartTrial = in.readBoolean();
  }

  @Override
  public void writeTo(StreamOutput out) throws IOException {
     out.writeBoolean(eligibleToStartTrial);
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;


import javax.crypto.BadPaddingException;
```

```java
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.StandardCharsets;
import java.security.GeneralSecurityException;
import java.security.KeyFactory;
import java.security.MessageDigest;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.NoSuchAlgorithmException;
import java.security.InvalidKeyException;
import java.security.spec.InvalidKeySpecException;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;
import java.util.Base64;

public class CryptUtils {
    // SALT must be at least 128bits for FIPS 140-2 compliance
    private static final byte[] SALT = {
            (byte) 0x74, (byte) 0x68, (byte) 0x69, (byte) 0x73,
            (byte) 0x69, (byte) 0x73, (byte) 0x74, (byte) 0x68,
            (byte) 0x65, (byte) 0x73, (byte) 0x61, (byte) 0x6C,
            (byte) 0x74, (byte) 0x77, (byte) 0x65, (byte) 0x75
    };
    private static final String KEY_ALGORITHM = "RSA";
    private static final char[] DEFAULT_PASS_PHRASE = "elasticsearch-license".toCharArray();
    private static final String KDF_ALGORITHM = "PBKDF2WithHmacSHA512";
    private static final int KDF_ITERATION_COUNT = 10000;
    private static final String CIPHER_ALGORITHM = "AES";
    // This can be changed to 256 once Java 9 is the minimum version
    // http://www.oracle.com/technetwork/java/javase/terms/readme/jdk9-readme-3852447.html#jce
    private static final int ENCRYPTION_KEY_LENGTH = 128;
    private static final SecureRandom RANDOM = new SecureRandom();

    /**
     * Read encrypted private key file content with default pass phrase
     */
    public static PrivateKey readEncryptedPrivateKey(byte[] fileContents) {
        return readEncryptedPrivateKey(fileContents, DEFAULT_PASS_PHRASE, false);
    }

    /**
     * Returns encrypted private key file content with default pass phrase
```

```java
 */
public static byte[] writeEncryptedPrivateKey(PrivateKey privateKey) {
    return writeEncryptedPrivateKey(privateKey, DEFAULT_PASS_PHRASE);
}


/**
 * Read encrypted private key file content with provided <code>passPhrase</code>
 */
public static PrivateKey readEncryptedPrivateKey(byte[] fileContents, char[] passPhrase, boolean preV4) {
    byte[] keyBytes = preV4 ? decryptV3Format(fileContents) : decrypt(fileContents, passPhrase);
    PKCS8EncodedKeySpec privateKeySpec = new PKCS8EncodedKeySpec(keyBytes);
    try {
        return KeyFactory.getInstance(KEY_ALGORITHM).generatePrivate(privateKeySpec);
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        throw new IllegalStateException(e);
    }
}


/**
 * Read public key file content
 */
public static PublicKey readPublicKey(byte[] fileContents) {
    X509EncodedKeySpec publicKeySpec = new X509EncodedKeySpec(fileContents);
    try {
        return KeyFactory.getInstance(CryptUtils.KEY_ALGORITHM).generatePublic(publicKeySpec);
    } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
        throw new IllegalStateException(e);
    }
}


/**
 * Returns encrypted public key file content with provided <code>passPhrase</code>
 */
public static byte[] writeEncryptedPublicKey(PublicKey publicKey) {
    X509EncodedKeySpec encodedKeySpec = new X509EncodedKeySpec(publicKey.getEncoded());
    return encrypt(encodedKeySpec.getEncoded(), DEFAULT_PASS_PHRASE);
}


/**
 * Returns encrypted private key file content with provided <code>passPhrase</code>
 */
public static byte[] writeEncryptedPrivateKey(PrivateKey privateKey, char[] passPhrase) {
    PKCS8EncodedKeySpec encodedKeySpec = new PKCS8EncodedKeySpec(privateKey.getEncoded());
    return encrypt(encodedKeySpec.getEncoded(), passPhrase);
}


/**
 * Encrypts provided <code>data</code> with <code>DEFAULT_PASS_PHRASE</code>
```

```java
 */
static byte[] encrypt(byte[] data) {
  return encrypt(data, DEFAULT_PASS_PHRASE);
}

/**
 * Decrypts provided <code>encryptedData</code> with <code>DEFAULT_PASS_PHRASE</code>
 */
static byte[] decrypt(byte[] encryptedData) {
  return decrypt(encryptedData, DEFAULT_PASS_PHRASE);
}

/**
 * Encrypts provided <code>data</code> with <code>passPhrase</code>
 */
private static byte[] encrypt(byte[] data, char[] passPhrase) {
  try {
    final Cipher encryptionCipher = getEncryptionCipher(deriveSecretKey(passPhrase));
    return encryptionCipher.doFinal(data);
  } catch (IllegalBlockSizeException | BadPaddingException e) {
    throw new IllegalStateException(e);
  }
}

/**
 * Decrypts provided <code>encryptedData</code> with <code>passPhrase</code>
 */
private static byte[] decrypt(byte[] encryptedData, char[] passPhrase) {
  try {
    final Cipher cipher = getDecryptionCipher(deriveSecretKey(passPhrase));
    return cipher.doFinal(encryptedData);
  } catch (IllegalBlockSizeException | BadPaddingException e) {
    throw new IllegalStateException(e);
  }
}

static byte[] encryptV3Format(byte[] data) {
  try {
    SecretKey encryptionKey = getV3Key();
    final Cipher encryptionCipher = getEncryptionCipher(encryptionKey);
    return encryptionCipher.doFinal(pad(data, 20));
  } catch (GeneralSecurityException e) {
    throw new IllegalStateException(e);
  }
}

static byte[] decryptV3Format(byte[] data) {
  try {
```

```java
            SecretKey decryptionKey = getV3Key();
            final Cipher decryptionCipher = getDecryptionCipher(decryptionKey);
            return unPad(decryptionCipher.doFinal(data));
        } catch (GeneralSecurityException e) {
            throw new IllegalStateException(e);
        }
    }


    private static SecretKey getV3Key() throws NoSuchAlgorithmException, InvalidKeySpecException {
        final byte[] salt = {
                (byte) 0xA9, (byte) 0xA2, (byte) 0xB5, (byte) 0xDE,
                (byte) 0x2A, (byte) 0x8A, (byte) 0x9A, (byte) 0xE6
        };
        final byte[] passBytes = "elasticsearch-license".getBytes(StandardCharsets.UTF_8);
        final byte[] digest = MessageDigest.getInstance("SHA-512").digest(passBytes);
        final char[] hashedPassphrase = Base64.getEncoder().encodeToString(digest).toCharArray();
        PBEKeySpec keySpec = new PBEKeySpec(hashedPassphrase, salt, 1024, 128);
        byte[] shortKey = SecretKeyFactory.getInstance("PBEWithSHA1AndDESede").
            generateSecret(keySpec).getEncoded();
        byte[] intermediaryKey = new byte[16];
        for (int i = 0, j = 0; i < 16; i++) {
            intermediaryKey[i] = shortKey[j];
            if (++j == shortKey.length)
                j = 0;
        }
        return new SecretKeySpec(intermediaryKey, "AES");
    }


    private static SecretKey deriveSecretKey(char[] passPhrase) {
        try {
            PBEKeySpec keySpec = new PBEKeySpec(passPhrase, SALT, KDF_ITERATION_COUNT,
ENCRYPTION_KEY_LENGTH);

            SecretKey secretKey = SecretKeyFactory.getInstance(KDF_ALGORITHM).
                generateSecret(keySpec);
            return new SecretKeySpec(secretKey.getEncoded(), CIPHER_ALGORITHM);
        } catch (NoSuchAlgorithmException | InvalidKeySpecException e) {
            throw new IllegalStateException(e);
        }
    }


    private static Cipher getEncryptionCipher(SecretKey secretKey) {
        return getCipher(Cipher.ENCRYPT_MODE, secretKey);
    }


    private static Cipher getDecryptionCipher(SecretKey secretKey) {
        return getCipher(Cipher.DECRYPT_MODE, secretKey);
    }
```

```java
    private static Cipher getCipher(int mode, SecretKey secretKey) {
        try {
            Cipher cipher = Cipher.getInstance(CIPHER_ALGORITHM);
            cipher.init(mode, secretKey, RANDOM);
            return cipher;
        } catch (NoSuchAlgorithmException | InvalidKeyException | NoSuchPaddingException e) {
            throw new IllegalStateException(e);
        }
    }

    private static byte[] pad(byte[] bytes, int length) {
        if (bytes.length >= length) {
            byte[] out = new byte[bytes.length + 1];
            System.arraycopy(bytes, 0, out, 0, bytes.length);
            out[bytes.length] = (byte) 1;
            return out;
        }

        byte[] out = new byte[length + 1];

        int i = 0;
        for (; i < bytes.length; i++)
            out[i] = bytes[i];

        int padded = length - i;

        // fill the rest with random bytes
        byte[] fill = new byte[padded - 1];
        RANDOM.nextBytes(fill);
        System.arraycopy(fill, 0, out, i, padded - 1);

        out[length] = (byte) (padded + 1);

        return out;
    }

    private static byte[] unPad(byte[] bytes) {
        int padded = (int) bytes[bytes.length - 1];
        int targetLength = bytes.length - padded;

        byte[] out = new byte[targetLength];

        System.arraycopy(bytes, 0, out, 0, targetLength);

        return out;
    }
}
```

```java
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.GET;

public class RestGetTrialStatus extends XPackRestHandler {

    RestGetTrialStatus(Settings settings, RestController controller) {
        super(settings);
        controller.registerHandler(GET, URI_BASE + "/license/trial_status", this);
    }

    @Override
    protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
{
        return channel -> client.licensing().prepareGetStartTrial().execute(
            new RestBuilderListener<GetTrialStatusResponse>(channel) {
                @Override
                public RestResponse buildResponse(GetTrialStatusResponse response, XContentBuilder builder) throws
Exception {
                    builder.startObject();
                    builder.field("eligible_to_start_trial", response.isEligibleToStartTrial());
                    builder.endObject();
                    return new BytesRestResponse(RestStatus.OK, builder);
                }
            });
    }

    @Override
    public String getName() {
        return "xpack_trial_status_action";
```

```
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.Action;

public class PostStartTrialAction extends Action<PostStartTrialResponse> {

    public static final PostStartTrialAction INSTANCE = new PostStartTrialAction();
    public static final String NAME = "cluster:admin/xpack/license/start_trial";

    private PostStartTrialAction() {
        super(NAME);
    }

    @Override
    public PostStartTrialResponse newResponse() {
        return new PostStartTrialResponse();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.Version;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.rest.RestStatus;

import java.util.stream.StreamSupport;

public class LicenseUtils {

    public static final String EXPIRED_FEATURE_METADATA = "es.license.expired.feature";

    /**
     * Exception to be thrown when a feature action requires a valid license, but license
     * has expired
     *
     * <code>feature</code> accessible through {@link #EXPIRED_FEATURE_METADATA} in the
```

```java
 * exception's rest header
 */
public static ElasticsearchSecurityException newComplianceException(String feature) {
    ElasticsearchSecurityException e = new ElasticsearchSecurityException("current license is non-compliant for
[{}]",
        RestStatus.FORBIDDEN, feature);
    e.addMetadata(EXPIRED_FEATURE_METADATA, feature);
    return e;
}

/**
 * Checks if a given {@link ElasticsearchSecurityException} refers to a feature that
 * requires a valid license, but the license has expired.
 */
public static boolean isLicenseExpiredException(ElasticsearchSecurityException exception) {
    return (exception != null) && (exception.getMetadata(EXPIRED_FEATURE_METADATA) != null);
}

public static boolean licenseNeedsExtended(License license) {
    return "basic".equals(license.type()) && license.expiryDate() !=
LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
}

/**
 * Checks if the signature of a self generated license with older version needs to be
 * recreated with the new key
 */
public static boolean signatureNeedsUpdate(License license, DiscoveryNodes currentNodes) {
    assert License.VERSION_CRYPTO_ALGORITHMS == License.VERSION_CURRENT : "update this method
when adding a new version";

    return ("basic".equals(license.type()) || "trial".equals(license.type())) &&
        // only upgrade signature when all nodes are ready to deserialize the new signature
        (license.version() < License.VERSION_CRYPTO_ALGORITHMS &&
            compatibleLicenseVersion(currentNodes) == License.VERSION_CRYPTO_ALGORITHMS
        );
}

public static int compatibleLicenseVersion(DiscoveryNodes currentNodes) {
    assert License.VERSION_CRYPTO_ALGORITHMS == License.VERSION_CURRENT : "update this method
when adding a new version";

    if (StreamSupport.stream(currentNodes.spliterator(), false)
        .allMatch(node -> node.getVersion().onOrAfter(Version.V_6_4_0))) {
        // License.VERSION_CRYPTO_ALGORITHMS was introduced in 6.4.0
        return License.VERSION_CRYPTO_ALGORITHMS;
    } else {
        return License.VERSION_START_DATE;
```

```java
        }
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeReadRequest;
import org.elasticsearch.common.io.stream.StreamInput;

import java.io.IOException;

public class GetTrialStatusRequest extends MasterNodeReadRequest<GetTrialStatusRequest> {

    public GetTrialStatusRequest() {
    }

    public GetTrialStatusRequest(StreamInput in) throws IOException {
        super(in);
    }

    @Override
    public ActionRequestValidationException validate() {
        return null;
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.Action;

public class DeleteLicenseAction extends Action<DeleteLicenseResponse> {

    public static final DeleteLicenseAction INSTANCE = new DeleteLicenseAction();
    public static final String NAME = "cluster:admin/xpack/license/delete";

    private DeleteLicenseAction() {
        super(NAME);
    }
```

```java
    @Override
    public DeleteLicenseResponse newResponse() {
        return new DeleteLicenseResponse();
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportPutLicenseAction extends TransportMasterNodeAction<PutLicenseRequest,
PutLicenseResponse> {

    private final LicenseService licenseService;

    @Inject
    public TransportPutLicenseAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                        LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                        IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, PutLicenseAction.NAME, transportService, clusterService, threadPool, actionFilters,
indexNameExpressionResolver,
            PutLicenseRequest::new);
        this.licenseService = licenseService;
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.MANAGEMENT;
    }

    @Override
```

```
  protected PutLicenseResponse newResponse() {
     return new PutLicenseResponse();
  }

  @Override
  protected ClusterBlockException checkBlock(PutLicenseRequest request, ClusterState state) {
     return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
  }

  @Override
  protected void masterOperation(final PutLicenseRequest request, ClusterState state, final
ActionListener<PutLicenseResponse>
        listener) throws ElasticsearchException {
     licenseService.registerLicense(request, listener);
  }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

class GetTrialStatusRequestBuilder extends ActionRequestBuilder<GetTrialStatusRequest,
GetTrialStatusResponse> {

  GetTrialStatusRequestBuilder(ElasticsearchClient client, GetTrialStatusAction action) {
     super(client, action, new GetTrialStatusRequest());
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.Strings;
import org.elasticsearch.common.logging.LoggerMessageFormat;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.XPackSettings;
import org.elasticsearch.xpack.core.monitoring.MonitoringField;
```

```java
import java.util.Collections;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.concurrent.CopyOnWriteArrayList;
import java.util.function.BiFunction;

/**
 * A holder for the current state of the license for all xpack features.
 */
public class XPackLicenseState {

    /** Messages for each feature which are printed when the license expires. */
    static final Map<String, String[]> EXPIRATION_MESSAGES;
    static {
        Map<String, String[]> messages = new LinkedHashMap<>();
        messages.put(XPackField.SECURITY, new String[] {
            "Cluster health, cluster stats and indices stats operations are blocked",
            "All data operations (read and write) continue to work"
        });
        messages.put(XPackField.WATCHER, new String[] {
            "PUT / GET watch APIs are disabled, DELETE watch API continues to work",
            "Watches execute and write to the history",
            "The actions of the watches don't execute"
        });
        messages.put(XPackField.MONITORING, new String[] {
            "The agent will stop collecting cluster and indices metrics",
            "The agent will stop automatically cleaning indices older than [xpack.monitoring.history.duration]"
        });
        messages.put(XPackField.GRAPH, new String[] {
            "Graph explore APIs are disabled"
        });
        messages.put(XPackField.MACHINE_LEARNING, new String[] {
            "Machine learning APIs are disabled"
        });
        messages.put(XPackField.LOGSTASH, new String[] {
            "Logstash will continue to poll centrally-managed pipelines"
        });
        messages.put(XPackField.DEPRECATION, new String[] {
            "Deprecation APIs are disabled"
        });
        messages.put(XPackField.UPGRADE, new String[] {
            "Upgrade API is disabled"
        });
        messages.put(XPackField.SQL, new String[] {
            "SQL support is disabled"
```

```
        });
        messages.put(XPackField.ROLLUP, new String[] {
            "Creating and Starting rollup jobs will no longer be allowed.",
            "Stopping/Deleting existing jobs, RollupCaps API and RollupSearch continue to function."
        });
        EXPIRATION_MESSAGES = Collections.unmodifiableMap(messages);
    }

    /**
     * Messages for each feature which are printed when the license type changes.
     * The value is a function taking the old and new license type, and returns the messages for that feature.
     */
    static final Map<String, BiFunction<OperationMode, OperationMode, String[]>>
ACKNOWLEDGMENT_MESSAGES;
    static {
        Map<String, BiFunction<OperationMode, OperationMode, String[]>> messages = new LinkedHashMap<>();
        messages.put(XPackField.SECURITY, XPackLicenseState::securityAcknowledgementMessages);
        messages.put(XPackField.WATCHER, XPackLicenseState::watcherAcknowledgementMessages);
        messages.put(XPackField.MONITORING, XPackLicenseState::monitoringAcknowledgementMessages);
        messages.put(XPackField.GRAPH, XPackLicenseState::graphAcknowledgementMessages);
        messages.put(XPackField.MACHINE_LEARNING,
XPackLicenseState::machineLearningAcknowledgementMessages);
        messages.put(XPackField.LOGSTASH, XPackLicenseState::logstashAcknowledgementMessages);
        messages.put(XPackField.SQL, XPackLicenseState::sqlAcknowledgementMessages);
        ACKNOWLEDGMENT_MESSAGES = Collections.unmodifiableMap(messages);
    }

    private static String[] securityAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
        switch (newMode) {
            case BASIC:
                switch (currentMode) {
                    case TRIAL:
                    case STANDARD:
                    case GOLD:
                    case PLATINUM:
                        return new String[] {
                            "The following X-Pack security functionality will be disabled: authentication, authorization, " +
                                "ip filtering, and auditing. Please restart your node after applying the license.",
                            "Field and document level access control will be disabled.",
                            "Custom realms will be ignored."
                        };
                }
                break;
            case GOLD:
                switch (currentMode) {
                    case BASIC:
                    case STANDARD:
```

```
            // ^^ though technically it was already disabled, it's not bad to remind them
            case TRIAL:
            case PLATINUM:
                return new String[] {
                    "Field and document level access control will be disabled.",
                    "Custom realms will be ignored."
                };
        }
        break;
    case STANDARD:
        switch (currentMode) {
            case BASIC:
                // ^^ though technically it was already disabled, it's not bad to remind them
            case GOLD:
            case PLATINUM:
            case TRIAL:
                return new String[] {
                    "Authentication will be limited to the native realms.",
                    "IP filtering and auditing will be disabled.",
                    "Field and document level access control will be disabled.",
                    "Custom realms will be ignored."
                };
        }
    }
    return Strings.EMPTY_ARRAY;
}

private static String[] watcherAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
    switch (newMode) {
        case BASIC:
            switch (currentMode) {
                case TRIAL:
                case STANDARD:
                case GOLD:
                case PLATINUM:
                    return new String[] { "Watcher will be disabled" };
            }
            break;
    }
    return Strings.EMPTY_ARRAY;
}

private static String[] monitoringAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
    switch (newMode) {
        case BASIC:
            switch (currentMode) {
```

```java
                case TRIAL:
                case STANDARD:
                case GOLD:
                case PLATINUM:
                    return new String[] {
                        LoggerMessageFormat.format(
                            "Multi-cluster support is disabled for clusters with [{}] license. If you are\n" +
                                "running multiple clusters, users won't be able to access the clusters with\n" +
                                "[{}] licenses from within a single X-Pack Kibana instance. You will have to deploy a\n" +
                                "separate and dedicated X-pack Kibana instance for each [{}] cluster you wish to monitor.",
                            newMode, newMode, newMode),
                        LoggerMessageFormat.format(
                            "Automatic index cleanup is locked to {} days for clusters with [{}] license.",
                            MonitoringField.HISTORY_DURATION.getDefault(Settings.EMPTY).days(), newMode)
                    };
            }
            break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] graphAcknowledgementMessages(OperationMode currentMode, OperationMode newMode)
{
        switch (newMode) {
            case BASIC:
            case STANDARD:
            case GOLD:
                switch (currentMode) {
                    case TRIAL:
                    case PLATINUM:
                        return new String[] { "Graph will be disabled" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] machineLearningAcknowledgementMessages(OperationMode currentMode,
OperationMode newMode) {
        switch (newMode) {
            case BASIC:
            case STANDARD:
            case GOLD:
                switch (currentMode) {
                    case TRIAL:
                    case PLATINUM:
                        return new String[] { "Machine learning will be disabled" };
                }
```

```java
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] logstashAcknowledgementMessages(OperationMode currentMode, OperationMode
newMode) {
        switch (newMode) {
            case BASIC:
                switch (currentMode) {
                    case TRIAL:
                    case STANDARD:
                    case GOLD:
                    case PLATINUM:
                        return new String[] { "Logstash will no longer poll for centrally-managed pipelines" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    private static String[] sqlAcknowledgementMessages(OperationMode currentMode, OperationMode newMode) {
        switch (newMode) {
            case BASIC:
            case STANDARD:
            case GOLD:
                switch (currentMode) {
                    case TRIAL:
                    case PLATINUM:
                        return new String[] { "JDBC support will be disabled, but you can continue to use SQL CLI and
REST endpoint" };
                }
                break;
        }
        return Strings.EMPTY_ARRAY;
    }

    /** A wrapper for the license mode and state, to allow atomically swapping. */
    private static class Status {

        /** The current "mode" of the license (ie license type). */
        final OperationMode mode;

        /** True if the license is active, or false if it is expired. */
        final boolean active;

        Status(OperationMode mode, boolean active) {
            this.mode = mode;
```

```java
        this.active = active;
    }
}

private volatile Status status = new Status(OperationMode.TRIAL, true);
private final List<Runnable> listeners = new CopyOnWriteArrayList<>();
private final boolean isSecurityEnabled;
private final boolean isSecurityExplicitlyEnabled;

public XPackLicenseState(Settings settings) {
    this.isSecurityEnabled = XPackSettings.SECURITY_ENABLED.get(settings);
    // 6.0+ requires TLS for production licenses, so if TLS is enabled and security is enabled
    // we can interpret this as an explicit enabling of security if the security enabled
    // setting is not explicitly set
    this.isSecurityExplicitlyEnabled = isSecurityEnabled &&
        (settings.hasValue(XPackSettings.SECURITY_ENABLED.getKey()) ||
XPackSettings.TRANSPORT_SSL_ENABLED.get(settings));
}

/** Updates the current state of the license, which will change what features are available. */
void update(OperationMode mode, boolean active) {
    status = new Status(mode, active);
    listeners.forEach(Runnable::run);
}

/** Add a listener to be notified on license change */
public void addListener(Runnable runnable) {
    listeners.add(Objects.requireNonNull(runnable));
}

/** Remove a listener */
public void removeListener(Runnable runnable) {
    listeners.remove(runnable);
}

/** Return the current license type. */
public OperationMode getOperationMode() {
    return status.mode;
}

/** Return true if the license is currently within its time boundaries, false otherwise. */
public boolean isActive() {
    return status.active;
}

/**
 * @return true if authentication and authorization should be enabled. this does not indicate what realms are
available
```

```
 * @see #allowedRealmType() for the enabled realms
 */
public boolean isAuthAllowed() {
    OperationMode mode = status.mode;
    return mode == OperationMode.STANDARD || mode == OperationMode.GOLD || mode ==
OperationMode.PLATINUM
        || mode == OperationMode.TRIAL;
}

/**
 * @return true if IP filtering should be enabled
 */
public boolean isIpFilteringAllowed() {
    OperationMode mode = status.mode;
    return mode == OperationMode.GOLD || mode == OperationMode.PLATINUM
        || mode == OperationMode.TRIAL;
}

/**
 * @return true if auditing should be enabled
 */
public boolean isAuditingAllowed() {
    OperationMode mode = status.mode;
    return mode == OperationMode.GOLD || mode == OperationMode.PLATINUM
        || mode == OperationMode.TRIAL;
}

/**
 * Indicates whether the stats and health API calls should be allowed. If a license is expired and past the grace
 * period then we deny these calls.
 *
 * @return true if the license allows for the stats and health APIs to be used.
 */
public boolean isStatsAndHealthAllowed() {
    return status.active;
}

/**
 * Determine if Document Level Security (DLS) and Field Level Security (FLS) should be enabled.
 * <p>
 * DLS and FLS are only disabled when the mode is not:
 * <ul>
 * <li>{@link OperationMode#PLATINUM}</li>
 * <li>{@link OperationMode#TRIAL}</li>
 * </ul>
 * Note: This does not consider the <em>state</em> of the license so that Security does not suddenly leak
information!
 *
```

```java
 * @return {@code true} to enable DLS and FLS. Otherwise {@code false}.
 */
public boolean isDocumentAndFieldLevelSecurityAllowed() {
   OperationMode mode = status.mode;
   return mode == OperationMode.TRIAL || mode == OperationMode.PLATINUM;
}


/** Classes of realms that may be available based on the license type. */
public enum AllowedRealmType {
   NONE,
   NATIVE,
   DEFAULT,
   ALL
}


/**
 * @return the type of realms that are enabled based on the license {@link OperationMode}
 */
public AllowedRealmType allowedRealmType() {
   switch (status.mode) {
      case PLATINUM:
      case TRIAL:
         return AllowedRealmType.ALL;
      case GOLD:
         return AllowedRealmType.DEFAULT;
      case STANDARD:
         return AllowedRealmType.NATIVE;
      default:
         return AllowedRealmType.NONE;
   }
}


/**
 * @return whether custom role providers are allowed based on the license {@link OperationMode}
 */
public boolean isCustomRoleProvidersAllowed() {
   final Status localStatus = status;
   return (localStatus.mode == OperationMode.PLATINUM || localStatus.mode == OperationMode.TRIAL )
         && localStatus.active;
}


/**
 * Determine if Watcher is available based on the current license.
 * <p>
 * Watcher is available if the license is active (hasn't expired) and of one of the following types:
 * <ul>
 * <li>{@link OperationMode#STANDARD}</li>
 * <li>{@link OperationMode#PLATINUM}</li>
```

```
 * <li>{@link OperationMode#GOLD}</li>
 * <li>{@link OperationMode#TRIAL}</li>
 * </ul>
 *
 * @return {@code true} as long as the license is valid. Otherwise {@code false}.
 */
public boolean isWatcherAllowed() {
    // status is volatile, so a local variable is used for a consistent view
    Status localStatus = status;

    if (localStatus.active == false) {
        return false;
    }

    switch (localStatus.mode) {
        case TRIAL:
        case GOLD:
        case PLATINUM:
        case STANDARD:
            return true;
        default:
            return false;
    }
}

/**
 * Monitoring is always available as long as there is a valid license
 *
 * @return true if the license is active
 */
public boolean isMonitoringAllowed() {
    return status.active;
}

/**
 * Monitoring Cluster Alerts requires the equivalent license to use Watcher.
 *
 * @return {@link #isWatcherAllowed()}
 * @see #isWatcherAllowed()
 */
public boolean isMonitoringClusterAlertsAllowed() {
    return isWatcherAllowed();
}

/**
 * Determine if the current license allows the retention of indices to be modified.
 * <p>
 * Only users with a non-{@link OperationMode#BASIC} license can update the retention period.
```

```
 * <p>
 * Note: This does not consider the <em>state</em> of the license so that any change is remembered for when
they fix their license.
 *
 * @return {@code true} if the user is allowed to modify the retention. Otherwise {@code false}.
 */
public boolean isUpdateRetentionAllowed() {
    final OperationMode mode = status.mode;
    return mode != OperationMode.BASIC && mode != OperationMode.MISSING;
}

/**
 * Determine if Graph Exploration should be enabled.
 * <p>
 * Exploration is only disabled when the license has expired or if the mode is not:
 * <ul>
 * <li>{@link OperationMode#PLATINUM}</li>
 * <li>{@link OperationMode#TRIAL}</li>
 * </ul>
 *
 * @return {@code true} as long as the license is valid. Otherwise {@code false}.
 */
public boolean isGraphAllowed() {
    // status is volatile
    Status localStatus = status;
    OperationMode operationMode = localStatus.mode;

    boolean licensed = operationMode == OperationMode.TRIAL || operationMode ==
OperationMode.PLATINUM;

    return licensed && localStatus.active;
}

/**
 * Determine if Machine Learning should be enabled.
 * <p>
 * Machine Learning is only disabled when the license has expired or if the
 * mode is not:
 * <ul>
 * <li>{@link OperationMode#PLATINUM}</li>
 * <li>{@link OperationMode#TRIAL}</li>
 * </ul>
 *
 * @return {@code true} as long as the license is valid. Otherwise
 *      {@code false}.
 */
public boolean isMachineLearningAllowed() {
    // status is volatile
```

```java
        Status localStatus = status;
        OperationMode operationMode = localStatus.mode;

        boolean licensed = operationMode == OperationMode.TRIAL || operationMode ==
OperationMode.PLATINUM;

        return licensed && localStatus.active;
    }

    /**
     * Rollup is always available as long as there is a valid license
     *
     * @return true if the license is active
     */
    public boolean isRollupAllowed() {
        return status.active;
    }

    /**
     * Logstash is allowed as long as there is an active license of type TRIAL, STANDARD, GOLD or PLATINUM
     * @return {@code true} as long as there is a valid license
     */
    public boolean isLogstashAllowed() {
        Status localStatus = status;

        if (localStatus.active == false) {
            return false;
        }

        switch (localStatus.mode) {
            case TRIAL:
            case GOLD:
            case PLATINUM:
            case STANDARD:
                return true;
            default:
                return false;
        }
    }

    /**
     * Deprecation APIs are always allowed as long as there is an active license
     * @return {@code true} as long as there is a valid license
     */
    public boolean isDeprecationAllowed() {
        return status.active;
    }
```

```java
/**
 * Determine if Upgrade API should be enabled.
 * <p>
 *  Upgrade API is not available in for all license types except {@link OperationMode#MISSING}
 *
 * @return {@code true} as long as the license is valid. Otherwise
 *          {@code false}.
 */
public boolean isUpgradeAllowed() {
  // status is volatile
  Status localStatus = status;
  // Should work on all active licenses
  return localStatus.active;
}


/**
 * Determine if SQL support should be enabled.
 * <p>
 *  SQL is available for all license types except {@link OperationMode#MISSING}
 */
public boolean isSqlAllowed() {
  return status.active;
}


/**
 * Determine if JDBC support should be enabled.
 * <p>
 *  JDBC is available only in for {@link OperationMode#PLATINUM} and {@link OperationMode#TRIAL}
licences
 */
public boolean isJdbcAllowed() {
  // status is volatile
  Status localStatus = status;
  OperationMode operationMode = localStatus.mode;

  boolean licensed = operationMode == OperationMode.TRIAL || operationMode ==
OperationMode.PLATINUM;

  return licensed && localStatus.active;
}

public boolean isTrialLicense() {
  return status.mode == OperationMode.TRIAL;
}

public boolean isSecurityAvailable() {
  OperationMode mode = status.mode;
  return mode == OperationMode.GOLD || mode == OperationMode.PLATINUM || mode ==
```

```
OperationMode.STANDARD ||
        mode == OperationMode.TRIAL;
  }

  public boolean isSecurityEnabled() {
    final OperationMode mode = status.mode;
    return mode == OperationMode.TRIAL ? isSecurityExplicitlyEnabled : isSecurityEnabled;
  }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.elasticsearch.Version;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.time.Clock;
import java.util.Collections;
import java.util.Map;
import java.util.UUID;

public class StartTrialClusterTask extends ClusterStateUpdateTask {

  private static final String ACKNOWLEDGEMENT_HEADER = "This API initiates a free 30-day trial for all
platinum features. " +
        "By starting this trial, you agree that it is subject to the terms and conditions at" +
        " https://www.elastic.co/legal/trial_license/. To begin your free trial, call /start_trial again and specify " +
        "the \"acknowledge=true\" parameter.";

  private static final Map<String, String[]> ACK_MESSAGES = Collections.singletonMap("security",
        new String[] {"With a trial license, X-Pack security features are available, but are not enabled by default."});

  private final Logger logger;
  private final String clusterName;
  private final PostStartTrialRequest request;
  private final ActionListener<PostStartTrialResponse> listener;
  private final Clock clock;
```

```java
StartTrialClusterTask(Logger logger, String clusterName, Clock clock, PostStartTrialRequest request,
            ActionListener<PostStartTrialResponse> listener) {
    this.logger = logger;
    this.clusterName = clusterName;
    this.request = request;
    this.listener = listener;
    this.clock = clock;
}


@Override
public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
    LicensesMetaData oldLicensesMetaData = oldState.metaData().custom(LicensesMetaData.TYPE);
    logger.debug("started self generated trial license: {}", oldLicensesMetaData);

    if (request.isAcknowledged() == false) {
        listener.onResponse(new
PostStartTrialResponse(PostStartTrialResponse.Status.NEED_ACKNOWLEDGEMENT,
            ACK_MESSAGES, ACKNOWLEDGEMENT_HEADER));
    } else if (oldLicensesMetaData == null || oldLicensesMetaData.isEligibleForTrial()) {
        listener.onResponse(new PostStartTrialResponse(PostStartTrialResponse.Status.UPGRADED_TO_TRIAL));
    } else {
        listener.onResponse(new
PostStartTrialResponse(PostStartTrialResponse.Status.TRIAL_ALREADY_ACTIVATED));
    }
}

@Override
public ClusterState execute(ClusterState currentState) throws Exception {
    XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
    LicensesMetaData currentLicensesMetaData = currentState.metaData().custom(LicensesMetaData.TYPE);

    if (request.isAcknowledged() == false) {
        return currentState;
    } else if (currentLicensesMetaData == null || currentLicensesMetaData.isEligibleForTrial()) {
        long issueDate = clock.millis();
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        long expiryDate = issueDate +
LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.getMillis();

        License.Builder specBuilder = License.builder()
            .uid(UUID.randomUUID().toString())
            .issuedTo(clusterName)
            .maxNodes(LicenseService.SELF_GENERATED_LICENSE_MAX_NODES)
            .issueDate(issueDate)
            .type(request.getType())
            .expiryDate(expiryDate);
        License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder, currentState.nodes());
        LicensesMetaData newLicensesMetaData = new LicensesMetaData(selfGeneratedLicense,
```

```
Version.CURRENT);
        mdBuilder.putCustom(LicensesMetaData.TYPE, newLicensesMetaData);
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
    } else {
        return currentState;
    }
}

@Override
public void onFailure(String source, @Nullable Exception e) {
    logger.error(new ParameterizedMessage("unexpected failure during [{}]", source), e);
    listener.onFailure(e);
}
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;

public class PutLicenseAction extends Action<PutLicenseResponse> {

    public static final PutLicenseAction INSTANCE = new PutLicenseAction();
    public static final String NAME = "cluster:admin/xpack/license/put";

    private PutLicenseAction() {
        super(NAME);
    }

    @Override
    public PutLicenseResponse newResponse() {
        return new PutLicenseResponse();
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.Action;

public class GetLicenseAction extends Action<GetLicenseResponse> {
```

```java
    public static final GetLicenseAction INSTANCE = new GetLicenseAction();
    public static final String NAME = "cluster:monitor/xpack/license/get";

    private GetLicenseAction() {
        super(NAME);
    }

    @Override
    public GetLicenseResponse newResponse() {
        return new GetLicenseResponse();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.elasticsearch.Version;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.time.Clock;
import java.util.Collections;
import java.util.Map;
import java.util.UUID;
import java.util.concurrent.atomic.AtomicReference;

public class StartBasicClusterTask extends ClusterStateUpdateTask {

    private static final String ACKNOWLEDGEMENT_HEADER = "This license update requires acknowledgement. To acknowledge the license, " +
            "please read the following messages and call /start_basic again, this time with the \"acknowledge=true\" parameter:";

    private final Logger logger;
    private final String clusterName;
    private final PostStartBasicRequest request;
    private final ActionListener<PostStartBasicResponse> listener;
    private final Clock clock;
    private AtomicReference<Map<String, String[]>> ackMessages = new
```

```
            AtomicReference<>(Collections.emptyMap());

        StartBasicClusterTask(Logger logger, String clusterName, Clock clock, PostStartBasicRequest request,
                        ActionListener<PostStartBasicResponse> listener) {
            this.logger = logger;
            this.clusterName = clusterName;
            this.request = request;
            this.listener = listener;
            this.clock = clock;
        }

        @Override
        public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
            LicensesMetaData oldLicensesMetaData = oldState.metaData().custom(LicensesMetaData.TYPE);
            logger.debug("license prior to starting basic license: {}", oldLicensesMetaData);
            License oldLicense = LicensesMetaData.extractLicense(oldLicensesMetaData);
            Map<String, String[]> acknowledgeMessages = ackMessages.get();
            if (acknowledgeMessages.isEmpty() == false) {
                listener.onResponse(new
PostStartBasicResponse(PostStartBasicResponse.Status.NEED_ACKNOWLEDGEMENT, acknowledgeMessages,
                    ACKNOWLEDGEMENT_HEADER));
            } else if (oldLicense != null && oldLicense.type().equals("basic")) {
                listener.onResponse(new
PostStartBasicResponse(PostStartBasicResponse.Status.ALREADY_USING_BASIC));
            } else {
                listener.onResponse(new PostStartBasicResponse(PostStartBasicResponse.Status.GENERATED_BASIC));
            }
        }

        @Override
        public ClusterState execute(ClusterState currentState) throws Exception {
            XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
            LicensesMetaData licensesMetaData = currentState.metaData().custom(LicensesMetaData.TYPE);
            License currentLicense = LicensesMetaData.extractLicense(licensesMetaData);
            if (currentLicense == null || currentLicense.type().equals("basic") == false) {
                long issueDate = clock.millis();
                MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
                License.Builder specBuilder = License.builder()
                        .uid(UUID.randomUUID().toString())
                        .issuedTo(clusterName)
                        .maxNodes(LicenseService.SELF_GENERATED_LICENSE_MAX_NODES)
                        .issueDate(issueDate)
                        .type("basic")
                        .expiryDate(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS);
                License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder, currentState.nodes());
                if (request.isAcknowledged() == false && currentLicense != null) {
                    Map<String, String[]> ackMessages = LicenseService.getAckMessages(selfGeneratedLicense,
currentLicense);
```

```java
        if (ackMessages.isEmpty() == false) {
            this.ackMessages.set(ackMessages);
            return currentState;
        }
    }
    Version trialVersion = null;
    if (licensesMetaData != null) {
        trialVersion = licensesMetaData.getMostRecentTrialVersion();
    }
    LicensesMetaData newLicensesMetaData = new LicensesMetaData(selfGeneratedLicense, trialVersion);
    mdBuilder.putCustom(LicensesMetaData.TYPE, newLicensesMetaData);
    return ClusterState.builder(currentState).metaData(mdBuilder).build();
} else {
    return currentState;
}
}

@Override
public void onFailure(String source, @Nullable Exception e) {
    logger.error(new ParameterizedMessage("unexpected failure during [{}]", source), e);
    listener.onFailure(e);
}
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportPostStartBasicAction extends TransportMasterNodeAction<PostStartBasicRequest,
PostStartBasicResponse> {

    private final LicenseService licenseService;
```

```java
    @Inject
    public TransportPostStartBasicAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                                IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, PostStartBasicAction.NAME, transportService, clusterService, threadPool, actionFilters,
            indexNameExpressionResolver, PostStartBasicRequest::new);
        this.licenseService = licenseService;
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.SAME;
    }

    @Override
    protected PostStartBasicResponse newResponse() {
        return new PostStartBasicResponse();
    }

    @Override
    protected void masterOperation(PostStartBasicRequest request, ClusterState state,
                        ActionListener<PostStartBasicResponse> listener) throws Exception {
        licenseService.startBasicLicense(request, listener);
    }

    @Override
    protected ClusterBlockException checkBlock(PostStartBasicRequest request, ClusterState state) {
        return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

class PostStartTrialRequestBuilder extends ActionRequestBuilder<PostStartTrialRequest, PostStartTrialResponse>
{

    PostStartTrialRequestBuilder(ElasticsearchClient client, PostStartTrialAction action) {
        super(client, action, new PostStartTrialRequest());
    }
```

```java
    public PostStartTrialRequestBuilder setAcknowledge(boolean acknowledge) {
        request.acknowledge(acknowledge);
        return this;
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.action.RestToXContentListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.DELETE;

public class RestDeleteLicenseAction extends XPackRestHandler {
    public RestDeleteLicenseAction(Settings settings, RestController controller) {
        super(settings);
        controller.registerHandler(DELETE, URI_BASE + "/license", this);
    }

    @Override
    public String getName() {
        return "xpack_delete_license_action";
    }

    @Override
    public RestChannelConsumer doPrepareRequest(final RestRequest request, final XPackClient client) throws
IOException {
        DeleteLicenseRequest deleteLicenseRequest = new DeleteLicenseRequest();
        deleteLicenseRequest.timeout(request.paramAsTime("timeout", deleteLicenseRequest.timeout()));
        deleteLicenseRequest.masterNodeTimeout(request.paramAsTime("master_timeout",
deleteLicenseRequest.masterNodeTimeout()));

        return channel -> client.es().admin().cluster().execute(DeleteLicenseAction.INSTANCE, deleteLicenseRequest,
            new RestToXContentListener<>(channel));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
```

```
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.apache.lucene.util.BytesRef;
import org.apache.lucene.util.BytesRefIterator;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.core.internal.io.Streams;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.Signature;
import java.security.SignatureException;
import java.util.Arrays;
import java.util.Base64;
import java.util.Collections;

/**
* Responsible for verifying signed licenses
*/
public class LicenseVerifier {

  /**
   * verifies the license content with the signature using the packaged
   * public key
   * @param license to verify
   * @return true if valid, false otherwise
   */
  public static boolean verifyLicense(final License license, byte[] publicKeyData) {
      byte[] signedContent = null;
      byte[] publicKeyFingerprint = null;
      try {
          byte[] signatureBytes = Base64.getDecoder().decode(license.signature());
          ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
          int version = byteBuffer.getInt();
          int magicLen = byteBuffer.getInt();
          byte[] magic = new byte[magicLen];
          byteBuffer.get(magic);
          int hashLen = byteBuffer.getInt();
```

```
            publicKeyFingerprint = new byte[hashLen];
            byteBuffer.get(publicKeyFingerprint);
            int signedContentLen = byteBuffer.getInt();
            signedContent = new byte[signedContentLen];
            byteBuffer.get(signedContent);
            XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
            license.toXContent(contentBuilder, new
ToXContent.MapParams(Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true")));
            Signature rsa = Signature.getInstance("SHA512withRSA");
            rsa.initVerify(CryptUtils.readPublicKey(publicKeyData));
            BytesRefIterator iterator = BytesReference.bytes(contentBuilder).iterator();
            BytesRef ref;
            while((ref = iterator.next()) != null) {
                rsa.update(ref.bytes, ref.offset, ref.length);
            }
            return rsa.verify(signedContent);
        } catch (IOException | NoSuchAlgorithmException | SignatureException | InvalidKeyException e) {
            throw new IllegalStateException(e);
        } finally {
            if (signedContent != null) {
                Arrays.fill(signedContent, (byte) 0);
            }
        }
    }

    public static boolean verifyLicense(final License license) {
        final byte[] publicKeyBytes;
        try (InputStream is = LicenseVerifier.class.getResourceAsStream("/public.key")) {
            ByteArrayOutputStream out = new ByteArrayOutputStream();
            Streams.copy(is, out);
            publicKeyBytes = out.toByteArray();
        } catch (IOException ex) {
            throw new IllegalStateException(ex);
        }
        return verifyLicense(license, publicKeyBytes);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.LoggingDeprecationHandler;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
```

```java
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.common.xcontent.XContentType;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.util.Base64;
import java.util.Collections;

import static org.elasticsearch.license.CryptUtils.encryptV3Format;
import static org.elasticsearch.license.CryptUtils.encrypt;
import static org.elasticsearch.license.CryptUtils.decryptV3Format;
import static org.elasticsearch.license.CryptUtils.decrypt;

class SelfGeneratedLicense {

  public static License create(License.Builder specBuilder, DiscoveryNodes currentNodes) {
    return create(specBuilder, LicenseUtils.compatibleLicenseVersion(currentNodes));
  }

  public static License create(License.Builder specBuilder, int version) {
    License spec = specBuilder
        .issuer("elasticsearch")
        .version(version)
        .build();
    final String signature;
    try {
      XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
      spec.toXContent(contentBuilder, new
ToXContent.MapParams(Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true")));
      byte[] encrypt;
      if (version < License.VERSION_CRYPTO_ALGORITHMS) {
        encrypt = encryptV3Format(BytesReference.toBytes(BytesReference.bytes(contentBuilder)));
      } else {
        encrypt = encrypt(BytesReference.toBytes(BytesReference.bytes(contentBuilder)));
      }
      byte[] bytes = new byte[4 + 4 + encrypt.length];
      ByteBuffer byteBuffer = ByteBuffer.wrap(bytes);
      // Set -version in signature
      byteBuffer.putInt(-version)
          .putInt(encrypt.length)
          .put(encrypt);
      signature = Base64.getEncoder().encodeToString(bytes);
    } catch (IOException e) {
      throw new IllegalStateException(e);
    }
```

```java
            return License.builder().fromLicenseSpec(spec, signature).build();
        }

    public static boolean verify(final License license) {
        try {
            byte[] signatureBytes = Base64.getDecoder().decode(license.signature());
            ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
            int version = byteBuffer.getInt();
            int contentLen = byteBuffer.getInt();
            byte[] content = new byte[contentLen];
            byteBuffer.get(content);
            final License expectedLicense;
            // Version in signature is -version, so check for -(-version) < 4
            byte[] decryptedContent = (-version < License.VERSION_CRYPTO_ALGORITHMS) ?
decryptV3Format(content) : decrypt(content);
            // EMPTY is safe here because we don't call namedObject
            try (XContentParser parser = XContentFactory.xContent(XContentType.JSON)
                    .createParser(NamedXContentRegistry.EMPTY, LoggingDeprecationHandler.INSTANCE,
decryptedContent)) {
                parser.nextToken();
                expectedLicense = License.builder().fromLicenseSpec(License.fromXContent(parser),
                    license.signature()).version(-version).build();
            }
            return license.equals(expectedLicense);
        } catch (IOException e) {
            throw new IllegalStateException(e);
        }
    }

    public static boolean validSelfGeneratedType(String type) {
        return "basic".equals(type) || "trial".equals(type);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeReadAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
```

```java
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportGetTrialStatusAction extends TransportMasterNodeReadAction<GetTrialStatusRequest,
GetTrialStatusResponse> {

    @Inject
    public TransportGetTrialStatusAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                                ThreadPool threadPool, ActionFilters actionFilters,
                                IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, GetTrialStatusAction.NAME, transportService, clusterService, threadPool, actionFilters,
            GetTrialStatusRequest::new, indexNameExpressionResolver);
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.SAME;
    }

    @Override
    protected GetTrialStatusResponse newResponse() {
        return new GetTrialStatusResponse();
    }

    @Override
    protected void masterOperation(GetTrialStatusRequest request, ClusterState state,
                        ActionListener<GetTrialStatusResponse> listener) throws Exception {
        LicensesMetaData licensesMetaData = state.metaData().custom(LicensesMetaData.TYPE);
        listener.onResponse(new GetTrialStatusResponse(licensesMetaData == null ||
licensesMetaData.isEligibleForTrial()));

    }

    @Override
    protected ClusterBlockException checkBlock(GetTrialStatusRequest request, ClusterState state) {
        return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_READ);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;
```

```java
import org.elasticsearch.common.logging.LoggerMessageFormat;
import org.elasticsearch.common.unit.TimeValue;

import java.util.UUID;

abstract class ExpirationCallback {

  static final String EXPIRATION_JOB_PREFIX = ".license_expiration_job_";

  public enum Orientation {PRE, POST}

  /**
   * Callback that is triggered every <code>frequency</code> when
   * current time is between <code>max</code> and <code>min</code>
   * before license expiry.
   */
  public abstract static class Pre extends ExpirationCallback {

    /**
     * Callback schedule prior to license expiry
     *
     * @param min       latest relative time to execute before license expiry
     * @param max       earliest relative time to execute before license expiry
     * @param frequency interval between execution
     */
    Pre(TimeValue min, TimeValue max, TimeValue frequency) {
        super(Orientation.PRE, min, max, frequency);
    }
  }

  /**
   * Callback that is triggered every <code>frequency</code> when
   * current time is between <code>min</code> and <code>max</code>
   * after license expiry.
   */
  public abstract static class Post extends ExpirationCallback {

    /**
     * Callback schedule after license expiry
     *
     * @param min       earliest relative time to execute after license expiry
     * @param max       latest relative time to execute after license expiry
     * @param frequency interval between execution
     */
    Post(TimeValue min, TimeValue max, TimeValue frequency) {
        super(Orientation.POST, min, max, frequency);
    }
  }
```

```
private final String id;
private final Orientation orientation;
private final long min;
private final long max;
private final long frequency;

private ExpirationCallback(Orientation orientation, TimeValue min, TimeValue max, TimeValue frequency) {
   this.orientation = orientation;
   this.min = (min == null) ? 0 : min.getMillis();
   this.max = (max == null) ? Long.MAX_VALUE : max.getMillis();
   this.frequency = frequency.getMillis();
   this.id = String.join("", EXPIRATION_JOB_PREFIX, UUID.randomUUID().toString());
}

public final String getId() {
   return id;
}

public final long getFrequency() {
   return frequency;
}

/**
 * Calculates the delay for the next trigger time. When <code>now</code> is in a
 * valid time bracket with respect to <code>expirationDate</code>, the delay is 0.
 * When <code>now</code> is before the time bracket, than delay to the start of the
 * time bracket and when <code>now</code> is passed the valid time bracket, the delay
 * is <code>null</code>
 * @param expirationDate license expiry date in milliseconds
 * @param now current time in milliseconds
 * @return time delay
 */
final TimeValue delay(long expirationDate, long now) {
   final TimeValue delay;
   switch (orientation) {
      case PRE:
         if (expirationDate >= now) {
            // license not yet expired
            long preExpiryDuration = expirationDate - now;
            if (preExpiryDuration > max) {
               // license duration is longer than maximum duration, delay it to the first match time
               delay = TimeValue.timeValueMillis(preExpiryDuration - max);
            } else if (preExpiryDuration <= max && preExpiryDuration >= min) {
               // no delay in valid time bracket
               delay = TimeValue.timeValueMillis(0);
            } else {
               // passed last match time
```

```
                delay = null;
              }
            } else {
              // invalid after license expiry
              delay = null;
            }
            break;
        case POST:
            if (expirationDate >= now) {
              // license not yet expired, delay it to the first match time
              delay = TimeValue.timeValueMillis(expirationDate - now + min);
            } else {
              // license has expired
              long expiredDuration = now - expirationDate;
              if (expiredDuration < min) {
                // license expiry duration is shorter than minimum duration, delay it to the first match time
                delay = TimeValue.timeValueMillis(min - expiredDuration);
              } else if (expiredDuration >= min && expiredDuration <= max) {
                // no delay in valid time bracket
                delay = TimeValue.timeValueMillis(0);
              } else {
                // passed last match time
                delay = null;
              }
            }
            break;
        default:
            throw new IllegalStateException("orientation [" + orientation + "] unknown");
      }
    return delay;
}

/**
 * {@link SchedulerEngine.Schedule#nextScheduledTimeAfter(long, long)} with respect to
 * license expiry date
 */
public final long nextScheduledTimeForExpiry(long expiryDate, long startTime, long time) {
    TimeValue delay = delay(expiryDate, time);
    if (delay != null) {
      long delayInMillis = delay.getMillis();
      if (delayInMillis == 0L) {
        if (startTime == time) {
          // initial trigger and in time bracket, schedule immediately
          return time;
        } else {
          // in time bracket, add frequency
          return time + frequency;
        }
```

```java
        } else {
            // not in time bracket
            return time + delayInMillis;
        }
    }
    return -1;
}

/**
 * Code to execute when the expiry callback is triggered in a valid
 * time bracket
 * @param license license to operate on
 */
public abstract void on(License license);

public final String toString() {
    return LoggerMessageFormat.format(null, "ExpirationCallback:(orientation [{}],  min [{}], max [{}], freq
[{}])",
            orientation.name(), TimeValue.timeValueMillis(min), TimeValue.timeValueMillis(max),
            TimeValue.timeValueMillis(frequency));
}
}
```

```java
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.Version;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.AckedClusterStateUpdateTask;
import org.elasticsearch.cluster.ClusterChangedEvent;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateListener;
import org.elasticsearch.cluster.ack.ClusterStateUpdateResponse;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.component.AbstractLifecycleComponent;
import org.elasticsearch.common.component.Lifecycle;
import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.elasticsearch.common.logging.LoggerMessageFormat;
import org.elasticsearch.common.settings.Setting;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
```

```java
import org.elasticsearch.discovery.DiscoveryModule;
import org.elasticsearch.env.Environment;
import org.elasticsearch.gateway.GatewayService;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.elasticsearch.xpack.core.XPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;
import org.elasticsearch.xpack.core.scheduler.SchedulerEngine;

import java.time.Clock;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.Set;
import java.util.concurrent.atomic.AtomicReference;

/**
 * Service responsible for managing {@link LicensesMetaData}.
 * <p>
 * On the master node, the service handles updating the cluster state when a new license is registered.
 * It also listens on all nodes for cluster state updates, and updates {@link XPackLicenseState} when
 * the license changes are detected in the cluster state.
 */
public class LicenseService extends AbstractLifecycleComponent implements ClusterStateListener,
SchedulerEngine.Listener {

    public static final Setting<String> SELF_GENERATED_LICENSE_TYPE = new
Setting<>("xpack.license.self_generated.type",
        (s) -> "basic", (s) -> {
      if (SelfGeneratedLicense.validSelfGeneratedType(s)) {
        return s;
      } else {
        throw new IllegalArgumentException("Illegal self generated license type [" + s + "]. Must be trial or basic.");
      }
    }, Setting.Property.NodeScope);

    // pkg private for tests
    static final TimeValue NON_BASIC_SELF_GENERATED_LICENSE_DURATION =
TimeValue.timeValueHours(30 * 24);

    static final Set<String> VALID_TRIAL_TYPES = new HashSet<>(Arrays.asList("trial", "platinum", "gold"));

    /**
     * Duration of grace period after a license has expired
```

```java
        */
        static final TimeValue GRACE_PERIOD_DURATION = days(7);

        public static final long BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS =
        Long.MAX_VALUE - days(365).millis();

        private final ClusterService clusterService;

        /**
         * The xpack feature state to update when license changes are made.
         */
        private final XPackLicenseState licenseState;

        /**
         * Currently active license
         */
        private final AtomicReference<License> currentLicense = new AtomicReference<>();
        private SchedulerEngine scheduler;
        private final Clock clock;

        /**
         * File watcher for operation mode changes
         */
        private final OperationModeFileWatcher operationModeFileWatcher;

        /**
         * Callbacks to notify relative to license expiry
         */
        private List<ExpirationCallback> expirationCallbacks = new ArrayList<>();

        /**
         * Max number of nodes licensed by generated trial license
         */
        static final int SELF_GENERATED_LICENSE_MAX_NODES = 1000;

        public static final String LICENSE_JOB = "licenseJob";

        private static final FormatDateTimeFormatter DATE_FORMATTER = Joda.forPattern("EEEE, MMMMM dd,
        yyyy", Locale.ROOT);

        private static final String ACKNOWLEDGEMENT_HEADER = "This license update requires acknowledgement.
        To acknowledge the license, " +
                "please read the following messages and update the license again, this time with the \"acknowledge=true\"
        parameter:";

        public LicenseService(Settings settings, ClusterService clusterService, Clock clock, Environment env,
                        ResourceWatcherService resourceWatcherService, XPackLicenseState licenseState) {
            super(settings);
```

```java
        this.clusterService = clusterService;
        this.clock = clock;
        this.scheduler = new SchedulerEngine(clock);
        this.licenseState = licenseState;
        this.operationModeFileWatcher = new OperationModeFileWatcher(resourceWatcherService,
            XPackPlugin.resolveConfigFile(env, "license_mode"), logger, () -> updateLicenseState(getLicense()));
        this.scheduler.register(this);
        populateExpirationCallbacks();
    }

    private void logExpirationWarning(long expirationMillis, boolean expired) {
        String expiredMsg = expired ? "expired" : "will expire";
        String general = LoggerMessageFormat.format(null, "License [{}] on [{}].\n" +
            "# If you have a new license, please update it. Otherwise, please reach out to\n" +
            "# your support contact.\n" +
            "# ", expiredMsg, DATE_FORMATTER.printer().print(expirationMillis));
        if (expired) {
            general = general.toUpperCase(Locale.ROOT);
        }
        StringBuilder builder = new StringBuilder(general);
        builder.append(System.lineSeparator());
        if (expired) {
            builder.append("# COMMERCIAL PLUGINS OPERATING WITH REDUCED FUNCTIONALITY");
        } else {
            builder.append("# Commercial plugins operate with reduced functionality on license expiration:");
        }
        XPackLicenseState.EXPIRATION_MESSAGES.forEach((feature, messages) -> {
            if (messages.length > 0) {
                builder.append(System.lineSeparator());
                builder.append("# - ");
                builder.append(feature);
                for (String message : messages) {
                    builder.append(System.lineSeparator());
                    builder.append("#  - ");
                    builder.append(message);
                }
            }
        });
        logger.warn("{}", builder);
    }

    private void populateExpirationCallbacks() {
        expirationCallbacks.add(new ExpirationCallback.Pre(days(7), days(25), days(1)) {
            @Override
            public void on(License license) {
                logExpirationWarning(license.expiryDate(), false);
            }
        });
```

```java
        expirationCallbacks.add(new ExpirationCallback.Pre(days(0), days(7), TimeValue.timeValueMinutes(10)) {
            @Override
            public void on(License license) {
                logExpirationWarning(license.expiryDate(), false);
            }
        });
        expirationCallbacks.add(new ExpirationCallback.Post(days(0), null, TimeValue.timeValueMinutes(10)) {
            @Override
            public void on(License license) {
                // logged when grace period begins
                logExpirationWarning(license.expiryDate(), true);
            }
        });
    }

    /**
     * Registers new license in the cluster
     * Master only operation. Installs a new license on the master provided it is VALID
     */
    public void registerLicense(final PutLicenseRequest request, final ActionListener<PutLicenseResponse> listener) {
        final License newLicense = request.license();
        final long now = clock.millis();
        if (!LicenseVerifier.verifyLicense(newLicense) || newLicense.issueDate() > now || newLicense.startDate() > now) {
            listener.onResponse(new PutLicenseResponse(true, LicensesStatus.INVALID));
        } else if (newLicense.type().equals("basic")) {
            listener.onFailure(new IllegalArgumentException("Registering basic licenses is not allowed."));
        } else if (newLicense.expiryDate() < now) {
            listener.onResponse(new PutLicenseResponse(true, LicensesStatus.EXPIRED));
        } else {
            if (!request.acknowledged()) {
                // TODO: ack messages should be generated on the master, since another node's cluster state may be behind...
                final License currentLicense = getLicense();
                if (currentLicense != null) {
                    Map<String, String[]> acknowledgeMessages = getAckMessages(newLicense, currentLicense);
                    if (acknowledgeMessages.isEmpty() == false) {
                        // needs acknowledgement
                        listener.onResponse(new PutLicenseResponse(false, LicensesStatus.VALID, ACKNOWLEDGEMENT_HEADER,
                                acknowledgeMessages));
                        return;
                    }
                }
            }

            if (newLicense.isProductionLicense()
```

```java
                && XPackSettings.SECURITY_ENABLED.get(settings)
                && XPackSettings.TRANSPORT_SSL_ENABLED.get(settings) == false
                && isProductionMode(settings, clusterService.localNode())) {
            // security is on but TLS is not configured we gonna fail the entire request and throw an exception
            throw new IllegalStateException("Cannot install a [" + newLicense.operationMode() +
                    "] license unless TLS is configured or security is disabled");
            // TODO we should really validate that all nodes have xpack installed and are consistently configured but this
            // should happen on a different level and not in this code
        } else {
            clusterService.submitStateUpdateTask("register license [" + newLicense.uid() + "]", new
                    AckedClusterStateUpdateTask<PutLicenseResponse>(request, listener) {
                @Override
                protected PutLicenseResponse newResponse(boolean acknowledged) {
                    return new PutLicenseResponse(acknowledged, LicensesStatus.VALID);
                }

                @Override
                public ClusterState execute(ClusterState currentState) throws Exception {
                    XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
                    MetaData currentMetadata = currentState.metaData();
                    LicensesMetaData licensesMetaData = currentMetadata.custom(LicensesMetaData.TYPE);
                    Version trialVersion = null;
                    if (licensesMetaData != null) {
                        trialVersion = licensesMetaData.getMostRecentTrialVersion();
                    }
                    MetaData.Builder mdBuilder = MetaData.builder(currentMetadata);
                    mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(newLicense,
trialVersion));
                    return ClusterState.builder(currentState).metaData(mdBuilder).build();
                }
            });
        }
    }
}

public static Map<String, String[]> getAckMessages(License newLicense, License currentLicense) {
    Map<String, String[]> acknowledgeMessages = new HashMap<>();
    if (!License.isAutoGeneratedLicense(currentLicense.signature()) // current license is not auto-generated
            && currentLicense.issueDate() > newLicense.issueDate()) { // and has a later issue date
        acknowledgeMessages.put("license", new String[]{
                "The new license is older than the currently installed license. " +
                        "Are you sure you want to override the current license?"});
    }
    XPackLicenseState.ACKNOWLEDGMENT_MESSAGES.forEach((feature, ackMessages) -> {
        String[] messages = ackMessages.apply(currentLicense.operationMode(), newLicense.operationMode());
        if (messages.length > 0) {
            acknowledgeMessages.put(feature, messages);
```

```java
            }
        });
        return acknowledgeMessages;
    }


    private static TimeValue days(int days) {
        return TimeValue.timeValueHours(days * 24);
    }


    @Override
    public void triggered(SchedulerEngine.Event event) {
        final LicensesMetaData licensesMetaData =
clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        if (licensesMetaData != null) {
            final License license = licensesMetaData.getLicense();
            if (event.getJobName().equals(LICENSE_JOB)) {
                updateLicenseState(license);
            } else if (event.getJobName().startsWith(ExpirationCallback.EXPIRATION_JOB_PREFIX)) {
                expirationCallbacks.stream()
                    .filter(expirationCallback -> expirationCallback.getId().equals(event.getJobName()))
                    .forEach(expirationCallback -> expirationCallback.on(license));
            }
        }
    }


    /**
     * Remove license from the cluster state metadata
     */
    public void removeLicense(final DeleteLicenseRequest request, final
ActionListener<ClusterStateUpdateResponse> listener) {
        clusterService.submitStateUpdateTask("delete license",
            new AckedClusterStateUpdateTask<ClusterStateUpdateResponse>(request, listener) {
                @Override
                protected ClusterStateUpdateResponse newResponse(boolean acknowledged) {
                    return new ClusterStateUpdateResponse(acknowledged);
                }

                @Override
                public ClusterState execute(ClusterState currentState) throws Exception {
                    MetaData metaData = currentState.metaData();
                    final LicensesMetaData currentLicenses = metaData.custom(LicensesMetaData.TYPE);
                    if (currentLicenses.getLicense() != LicensesMetaData.LICENSE_TOMBSTONE) {
                        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
                        LicensesMetaData newMetadata = new
LicensesMetaData(LicensesMetaData.LICENSE_TOMBSTONE,
                            currentLicenses.getMostRecentTrialVersion());
                        mdBuilder.putCustom(LicensesMetaData.TYPE, newMetadata);
```

```java
                return ClusterState.builder(currentState).metaData(mdBuilder).build();
            } else {
                return currentState;
            }
        }
    });
}

public License getLicense() {
    final License license = getLicense(clusterService.state().metaData());
    return license == LicensesMetaData.LICENSE_TOMBSTONE ? null : license;
}

void startTrialLicense(PostStartTrialRequest request, final ActionListener<PostStartTrialResponse> listener) {
    if (VALID_TRIAL_TYPES.contains(request.getType()) == false) {
        throw new IllegalArgumentException("Cannot start trial of type [" + request.getType() + "]. Valid trial types
are "
            + VALID_TRIAL_TYPES + ".");
    }
    StartTrialClusterTask task = new StartTrialClusterTask(logger, clusterService.getClusterName().value(), clock,
request, listener);
    clusterService.submitStateUpdateTask("started trial license", task);
}

void startBasicLicense(PostStartBasicRequest request, final ActionListener<PostStartBasicResponse> listener) {
    StartBasicClusterTask task = new StartBasicClusterTask(logger, clusterService.getClusterName().value(),
clock, request, listener);
    clusterService.submitStateUpdateTask("start basic license", task);
}

/**
 * Master-only operation to generate a one-time global self generated license.
 * The self generated license is only generated and stored if the current cluster state metadata
 * has no existing license. If the cluster currently has a basic license that has an expiration date,
 * a new basic license with no expiration date is generated.
 */
private void registerOrUpdateSelfGeneratedLicense() {
    clusterService.submitStateUpdateTask("maybe generate license for cluster",
        new StartupSelfGeneratedLicenseTask(settings, clock, clusterService));
}

@Override
protected void doStart() throws ElasticsearchException {
    clusterService.addListener(this);
    scheduler.start(Collections.emptyList());
    logger.debug("initializing license state");
    if (clusterService.lifecycleState() == Lifecycle.State.STARTED) {
        final ClusterState clusterState = clusterService.state();
```

```java
            if (clusterState.blocks().hasGlobalBlock(GatewayService.STATE_NOT_RECOVERED_BLOCK) == false
&&
                clusterState.nodes().getMasterNode() != null &&
XPackPlugin.isReadyForXPackCustomMetadata(clusterState)) {
            final LicensesMetaData currentMetaData = clusterState.metaData().custom(LicensesMetaData.TYPE);
            boolean noLicense = currentMetaData == null || currentMetaData.getLicense() == null;
            if (clusterState.getNodes().isLocalNodeElectedMaster() &&
                (noLicense || LicenseUtils.licenseNeedsExtended(currentMetaData.getLicense()))) {
                // triggers a cluster changed event eventually notifying the current licensee
                registerOrUpdateSelfGeneratedLicense();
            }
        }
    }
}

    @Override
    protected void doStop() throws ElasticsearchException {
        clusterService.removeListener(this);
        scheduler.stop();
        // clear current license
        currentLicense.set(null);
    }

    @Override
    protected void doClose() throws ElasticsearchException {
    }

    /**
     * When there is no global block on {@link
org.elasticsearch.gateway.GatewayService#STATE_NOT_RECOVERED_BLOCK}
     * notify licensees and issue auto-generated license if no license has been installed/issued yet.
     */
    @Override
    public void clusterChanged(ClusterChangedEvent event) {
        final ClusterState previousClusterState = event.previousState();
        final ClusterState currentClusterState = event.state();
        if (!currentClusterState.blocks().hasGlobalBlock(GatewayService.STATE_NOT_RECOVERED_BLOCK)) {
            if (XPackPlugin.isReadyForXPackCustomMetadata(currentClusterState) == false) {
                logger.debug("cannot add license to cluster as the following nodes might not understand the license
metadata: {}",
                    () -> XPackPlugin.nodesNotReadyForXPackCustomMetadata(currentClusterState));
                return;
            }

            final LicensesMetaData prevLicensesMetaData =
previousClusterState.getMetaData().custom(LicensesMetaData.TYPE);
            final LicensesMetaData currentLicensesMetaData =
currentClusterState.getMetaData().custom(LicensesMetaData.TYPE);
```

```java
            if (logger.isDebugEnabled()) {
               logger.debug("previous [{}]", prevLicensesMetaData);
               logger.debug("current [{}]", currentLicensesMetaData);
            }
            // notify all interested plugins
            if (previousClusterState.blocks().hasGlobalBlock(GatewayService.STATE_NOT_RECOVERED_BLOCK)
                  || prevLicensesMetaData == null) {
               if (currentLicensesMetaData != null) {
                  onUpdate(currentLicensesMetaData);
               }
            } else if (!prevLicensesMetaData.equals(currentLicensesMetaData)) {
               onUpdate(currentLicensesMetaData);
            }

            License currentLicense = null;
            boolean noLicenseInPrevMetadata = prevLicensesMetaData == null || prevLicensesMetaData.getLicense()
== null;
            if (noLicenseInPrevMetadata == false) {
               currentLicense = prevLicensesMetaData.getLicense();
            }
            boolean noLicenseInCurrentMetadata = (currentLicensesMetaData == null ||
currentLicensesMetaData.getLicense() == null);
            if (noLicenseInCurrentMetadata == false) {
               currentLicense = currentLicensesMetaData.getLicense();
            }

            boolean noLicense = noLicenseInPrevMetadata && noLicenseInCurrentMetadata;
            // auto-generate license if no licenses ever existed or if the current license is basic and
            // needs extended or if the license signature needs to be updated. this will trigger a subsequent cluster changed
event
            if (currentClusterState.getNodes().isLocalNodeElectedMaster() &&
                  (noLicense || LicenseUtils.licenseNeedsExtended(currentLicense) ||
                     LicenseUtils.signatureNeedsUpdate(currentLicense, currentClusterState.nodes()))) {
               registerOrUpdateSelfGeneratedLicense();
            }
         } else if (logger.isDebugEnabled()) {
            logger.debug("skipped license notifications reason: [{}]",
GatewayService.STATE_NOT_RECOVERED_BLOCK);
         }
      }

      protected void updateLicenseState(final License license) {
         if (license == LicensesMetaData.LICENSE_TOMBSTONE) {
            // implies license has been explicitly deleted
            licenseState.update(License.OperationMode.MISSING, false);
            return;
         }
         if (license != null) {
```

```
        long time = clock.millis();
        boolean active;
        if (license.expiryDate() == BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS) {
            active = true;
        } else {
            // We subtract the grace period from the current time to avoid overflowing on an expiration
            // date that is near Long.MAX_VALUE
            active = time >= license.issueDate() && time - GRACE_PERIOD_DURATION.getMillis() <
license.expiryDate();
        }
        licenseState.update(license.operationMode(), active);

        if (active) {
            if (time < license.expiryDate()) {
                logger.debug("license [{}] - valid", license.uid());
            } else {
                logger.warn("license [{}] - grace", license.uid());
            }
        } else {
            logger.warn("license [{}] - expired", license.uid());
        }
    }
}

/**
 * Notifies registered licensees of license state change and/or new active license
 * based on the license in <code>currentLicensesMetaData</code>.
 * Additionally schedules license expiry notifications and event callbacks
 * relative to the current license's expiry
 */
private void onUpdate(final LicensesMetaData currentLicensesMetaData) {
    final License license = getLicense(currentLicensesMetaData);
    // license can be null if the trial license is yet to be auto-generated
    // in this case, it is a no-op
    if (license != null) {
        final License previousLicense = currentLicense.get();
        if (license.equals(previousLicense) == false) {
            currentLicense.set(license);
            license.setOperationModeFileWatcher(operationModeFileWatcher);
            scheduler.add(new SchedulerEngine.Job(LICENSE_JOB, nextLicenseCheck(license)));
            for (ExpirationCallback expirationCallback : expirationCallbacks) {
                scheduler.add(new SchedulerEngine.Job(expirationCallback.getId(),
                    (startTime, now) ->
                        expirationCallback.nextScheduledTimeForExpiry(license.expiryDate(), startTime, now)));
            }
            if (previousLicense != null) {
                // remove operationModeFileWatcher to gc the old license object
                previousLicense.removeOperationModeFileWatcher();
```

```
                }
                logger.info("license [{}] mode [{}] - valid", license.uid(),
                        license.operationMode().name().toLowerCase(Locale.ROOT));
            }
            updateLicenseState(license);
        }
    }

    // pkg private for tests
    static SchedulerEngine.Schedule nextLicenseCheck(License license) {
        return (startTime, time) -> {
            if (time < license.issueDate()) {
                // when we encounter a license with a future issue date
                // which can happen with autogenerated license,
                // we want to schedule a notification on the license issue date
                // so the license is notificed once it is valid
                // see https://github.com/elastic/x-plugins/issues/983
                return license.issueDate();
            } else if (time < license.expiryDate()) {
                return license.expiryDate();
            } else if (time < license.expiryDate() + GRACE_PERIOD_DURATION.getMillis()) {
                return license.expiryDate() + GRACE_PERIOD_DURATION.getMillis();
            }
            return -1; // license is expired, no need to check again
        };
    }

    public static License getLicense(final MetaData metaData) {
        final LicensesMetaData licensesMetaData = metaData.custom(LicensesMetaData.TYPE);
        return getLicense(licensesMetaData);
    }

    static License getLicense(final LicensesMetaData metaData) {
        if (metaData != null) {
            License license = metaData.getLicense();
            if (license == LicensesMetaData.LICENSE_TOMBSTONE) {
                return license;
            } else if (license != null) {
                boolean autoGeneratedLicense = License.isAutoGeneratedLicense(license.signature());
                if ((autoGeneratedLicense && SelfGeneratedLicense.verify(license))
                        || (!autoGeneratedLicense && LicenseVerifier.verifyLicense(license))) {
                    return license;
                }
            }
        }
        return null;
    }
```

```java
    private static boolean isProductionMode(Settings settings, DiscoveryNode localNode) {
        final boolean singleNodeDisco = "single-
node".equals(DiscoveryModule.DISCOVERY_TYPE_SETTING.get(settings));
        return singleNodeDisco == false && isBoundToLoopback(localNode) == false;
    }

    private static boolean isBoundToLoopback(DiscoveryNode localNode) {
        return localNode.getAddress().address().getAddress().isLoopbackAddress();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

public class DeleteLicenseRequestBuilder extends AcknowledgedRequestBuilder<DeleteLicenseRequest,
DeleteLicenseResponse,
        DeleteLicenseRequestBuilder> {

    public DeleteLicenseRequestBuilder(ElasticsearchClient client) {
        this(client, DeleteLicenseAction.INSTANCE);
    }

    /**
     * Creates new get licenses request builder
     *
     * @param client elasticsearch client
     */
    public DeleteLicenseRequestBuilder(ElasticsearchClient client, DeleteLicenseAction action) {
        super(client, action, new DeleteLicenseRequest());
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
```

```java
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.POST;

public class RestPostStartBasicLicense extends XPackRestHandler {

    RestPostStartBasicLicense(Settings settings, RestController controller) {
        super(settings);
        controller.registerHandler(POST, URI_BASE + "/license/start_basic", this);
    }

    @Override
    protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
{
        PostStartBasicRequest startBasicRequest = new PostStartBasicRequest();
        startBasicRequest.acknowledge(request.paramAsBoolean("acknowledge", false));
        startBasicRequest.timeout(request.paramAsTime("timeout", startBasicRequest.timeout()));
        startBasicRequest.masterNodeTimeout(request.paramAsTime("master_timeout",
startBasicRequest.masterNodeTimeout()));
        return channel -> client.licensing().postStartBasic(startBasicRequest, new
RestBuilderListener<PostStartBasicResponse>(channel) {
            @Override
            public RestResponse buildResponse(PostStartBasicResponse response, XContentBuilder builder) throws
Exception {
                PostStartBasicResponse.Status status = response.getStatus();
                response.toXContent(builder, ToXContent.EMPTY_PARAMS);
                return new BytesRestResponse(status.getRestStatus(), builder);
            }
        });
    }

    @Override
    public String getName() {
        return "xpack_start_basic_action";
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
```

```java
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.io.stream.Writeable;
import org.elasticsearch.common.xcontent.ToXContentObject;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.xpack.core.XPackBuild;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.Set;
import java.util.stream.Collectors;

public class XPackInfoResponse extends ActionResponse {

    @Nullable private BuildInfo buildInfo;
    @Nullable private LicenseInfo licenseInfo;
    @Nullable private FeatureSetsInfo featureSetsInfo;

    public XPackInfoResponse() {}

    public XPackInfoResponse(@Nullable BuildInfo buildInfo, @Nullable LicenseInfo licenseInfo, @Nullable
FeatureSetsInfo featureSetsInfo) {
        this.buildInfo = buildInfo;
        this.licenseInfo = licenseInfo;
        this.featureSetsInfo = featureSetsInfo;
    }

    /**
     * @return  The build info (incl. build hash and timestamp)
     */
    public BuildInfo getBuildInfo() {
        return buildInfo;
    }

    /**
     * @return  The current license info (incl. UID, type/mode. status and expiry date). May return {@code null}
when no
     *          license is currently installed.
```

```java
     */
    public LicenseInfo getLicenseInfo() {
        return licenseInfo;
    }

    /**
     * @return  The current status of the feature sets in X-Pack. Feature sets describe the features available/enabled in
    X-Pack.
     */
    public FeatureSetsInfo getFeatureSetsInfo() {
        return featureSetsInfo;
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        super.writeTo(out);
        out.writeOptionalWriteable(buildInfo);
        out.writeOptionalWriteable(licenseInfo);
        out.writeOptionalWriteable(featureSetsInfo);
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        this.buildInfo = in.readOptionalWriteable(BuildInfo::new);
        this.licenseInfo = in.readOptionalWriteable(LicenseInfo::new);
        this.featureSetsInfo = in.readOptionalWriteable(FeatureSetsInfo::new);
    }

    public static class LicenseInfo implements ToXContentObject, Writeable {

        private final String uid;
        private final String type;
        private final String mode;
        private final long expiryDate;
        private final License.Status status;

        public LicenseInfo(License license) {
            this(license.uid(), license.type(), license.operationMode().name().toLowerCase(Locale.ROOT),
                    license.status(), license.expiryDate());
        }

        public LicenseInfo(StreamInput in) throws IOException {
            this(in.readString(), in.readString(), in.readString(), License.Status.readFrom(in), in.readLong());
        }

        public LicenseInfo(String uid, String type, String mode, License.Status status, long expiryDate) {
            this.uid = uid;
            this.type = type;
```

```java
        this.mode = mode;
        this.status = status;
        this.expiryDate = expiryDate;
    }

    public String getUid() {
        return uid;
    }

    public String getType() {
        return type;
    }

    public String getMode() {
        return mode;
    }

    public long getExpiryDate() {
        return expiryDate;
    }

    public License.Status getStatus() {
        return status;
    }

    @Override
    public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
        builder.startObject()
            .field("uid", uid)
            .field("type", type)
            .field("mode", mode)
            .field("status", status.label());
        if (expiryDate != LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS) {
            builder.timeField("expiry_date_in_millis", "expiry_date", expiryDate);
        }
        return builder.endObject();
    }

    public void writeTo(StreamOutput out) throws IOException {
        out.writeString(uid);
        out.writeString(type);
        out.writeString(mode);
        status.writeTo(out);
        out.writeLong(expiryDate);
    }
}

public static class BuildInfo implements ToXContentObject, Writeable {
```

```java
    private final String hash;
    private final String timestamp;

    public BuildInfo(XPackBuild build) {
        this(build.shortHash(), build.date());
    }

    public BuildInfo(StreamInput input) throws IOException {
        this(input.readString(), input.readString());
    }

    public BuildInfo(String hash, String timestamp) {
        this.hash = hash;
        this.timestamp = timestamp;
    }

    public String getHash() {
        return hash;
    }

    public String getTimestamp() {
        return timestamp;
    }

    @Override
    public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
        return builder.startObject()
            .field("hash", hash)
            .field("date", timestamp)
            .endObject();
    }

    public void writeTo(StreamOutput output) throws IOException {
        output.writeString(hash);
        output.writeString(timestamp);
    }
}

public static class FeatureSetsInfo implements ToXContentObject, Writeable {

    private final Map<String, FeatureSet> featureSets;

    public FeatureSetsInfo(StreamInput in) throws IOException {
        int size = in.readVInt();
        Map<String, FeatureSet> featureSets = new HashMap<>(size);
        for (int i = 0; i < size; i++) {
            FeatureSet featureSet = new FeatureSet(in);
```

```java
                    featureSets.put(featureSet.name, featureSet);
                }
                this.featureSets = Collections.unmodifiableMap(featureSets);
            }

            public FeatureSetsInfo(Set<FeatureSet> featureSets) {
                Map<String, FeatureSet> map = new HashMap<>(featureSets.size());
                for (FeatureSet featureSet : featureSets) {
                    map.put(featureSet.name, featureSet);
                }
                this.featureSets = Collections.unmodifiableMap(map);
            }

            public Map<String, FeatureSet> getFeatureSets() {
                return featureSets;
            }

            @Override
            public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
                builder.startObject();
                List<String> names = new
ArrayList<>(this.featureSets.keySet()).stream().sorted().collect(Collectors.toList());
                for (String name : names) {
                    builder.field(name, featureSets.get(name), params);
                }
                return builder.endObject();
            }

            public void writeTo(StreamOutput out) throws IOException {
                out.writeVInt(featureSets.size());
                for (FeatureSet featureSet : featureSets.values()) {
                    featureSet.writeTo(out);
                }
            }

            public static class FeatureSet implements ToXContentObject, Writeable {

                private final String name;
                @Nullable private final String description;
                private final boolean available;
                private final boolean enabled;
                @Nullable private final Map<String, Object> nativeCodeInfo;

                public FeatureSet(StreamInput in) throws IOException {
                    this(in.readString(), in.readOptionalString(), in.readBoolean(), in.readBoolean(),
                        in.getVersion().onOrAfter(Version.V_5_4_0) ? in.readMap() : null);
                }
```

```java
public FeatureSet(String name, @Nullable String description, boolean available, boolean enabled,
        @Nullable Map<String, Object> nativeCodeInfo) {
    this.name = name;
    this.description = description;
    this.available = available;
    this.enabled = enabled;
    this.nativeCodeInfo = nativeCodeInfo;
}

public String name() {
    return name;
}

@Nullable
public String description() {
    return description;
}

public boolean available() {
    return available;
}

public boolean enabled() {
    return enabled;
}

@Nullable
public Map<String, Object> nativeCodeInfo() {
    return nativeCodeInfo;
}

public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
    builder.startObject();
    if (description != null) {
        builder.field("description", description);
    }
    builder.field("available", available);
    builder.field("enabled", enabled);
    if (nativeCodeInfo != null) {
        builder.field("native_code_info", nativeCodeInfo);
    }
    return builder.endObject();
}

public void writeTo(StreamOutput out) throws IOException {
    out.writeString(name);
    out.writeOptionalString(description);
    out.writeBoolean(available);
```

```
            out.writeBoolean(enabled);
            if (out.getVersion().onOrAfter(Version.V_5_4_0)) {
                out.writeMap(nativeCodeInfo);
            }
        }
    }

}
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;

class GetBasicStatusRequestBuilder extends ActionRequestBuilder<GetBasicStatusRequest,
GetBasicStatusResponse> {

    GetBasicStatusRequestBuilder(ElasticsearchClient client, GetBasicStatusAction action) {
        super(client, action, new GetBasicStatusRequest());
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;

import static org.elasticsearch.rest.RestRequest.Method.GET;
```

```java
public class RestGetBasicStatus extends XPackRestHandler {

  RestGetBasicStatus(Settings settings, RestController controller) {
    super(settings);
    controller.registerHandler(GET, URI_BASE + "/license/basic_status", this);
  }

  @Override
  protected RestChannelConsumer doPrepareRequest(RestRequest request, XPackClient client) throws IOException
{
    return channel -> client.licensing().prepareGetStartBasic().execute(
        new RestBuilderListener<GetBasicStatusResponse>(channel) {
          @Override
          public RestResponse buildResponse(GetBasicStatusResponse response, XContentBuilder builder)
throws Exception {
            builder.startObject();
            builder.field("eligible_to_start_basic", response.isEligibleToStartBasic());
            builder.endObject();
            return new BytesRestResponse(RestStatus.OK, builder);
          }
        });
  }

  @Override
  public String getName() {
    return "xpack_basic_status_action";
  }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.ValidateActions;
import org.elasticsearch.action.support.master.AcknowledgedRequest;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentType;

import java.io.IOException;


public class PutLicenseRequest extends AcknowledgedRequest<PutLicenseRequest> {
```

```java
private License license;
private boolean acknowledge = false;

public PutLicenseRequest() {
}

@Override
public ActionRequestValidationException validate() {
    return (license == null) ? ValidateActions.addValidationError("license is missing", null) : null;
}

/**
 * Parses license from json format to an instance of {@link License}
 *
 * @param licenseDefinition licenses definition
 * @param xContentType the content type of the license
 */
public PutLicenseRequest license(BytesReference licenseDefinition, XContentType xContentType) {
    try {
        return license(License.fromSource(licenseDefinition, xContentType));
    } catch (IOException e) {
        throw new IllegalArgumentException("failed to parse license source", e);
    }
}

public PutLicenseRequest license(License license) {
    this.license = license;
    return this;
}

public License license() {
    return license;
}

public PutLicenseRequest acknowledge(boolean acknowledge) {
    this.acknowledge = acknowledge;
    return this;
}

public boolean acknowledged() {
    return acknowledge;
}

@Override
public void readFrom(StreamInput in) throws IOException {
    super.readFrom(in);
    license = License.readLicense(in);
    acknowledge = in.readBoolean();
```

```java
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        super.writeTo(out);
        license.writeTo(out);
        out.writeBoolean(acknowledge);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.Action;

public class PostStartBasicAction extends Action<PostStartBasicResponse> {

    public static final PostStartBasicAction INSTANCE = new PostStartBasicAction();
    public static final String NAME = "cluster:admin/xpack/license/start_basic";

    private PostStartBasicAction() {
        super(NAME);
    }

    @Override
    public PostStartBasicResponse newResponse() {
        return new PostStartBasicResponse();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.AcknowledgedRequest;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

public class PostStartBasicRequest extends AcknowledgedRequest<PostStartBasicRequest> {
```

```java
    private boolean acknowledge = false;

    @Override
    public ActionRequestValidationException validate() {
        return null;
    }

    public PostStartBasicRequest acknowledge(boolean acknowledge) {
        this.acknowledge = acknowledge;
        return this;
    }

    public boolean isAcknowledged() {
        return acknowledge;
    }

    @Override
    public void readFrom(StreamInput in) throws IOException {
        super.readFrom(in);
        acknowledge = in.readBoolean();
    }

    @Override
    public void writeTo(StreamOutput out) throws IOException {
        super.writeTo(out);
        out.writeBoolean(acknowledge);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

class GetBasicStatusResponse extends ActionResponse {

    private boolean eligibleToStartBasic;

    GetBasicStatusResponse() {
    }
```

```java
GetBasicStatusResponse(boolean eligibleToStartBasic) {
    this.eligibleToStartBasic = eligibleToStartBasic;
}

boolean isEligibleToStartBasic() {
    return eligibleToStartBasic;
}

@Override
public void readFrom(StreamInput in) throws IOException {
    eligibleToStartBasic = in.readBoolean();
}

@Override
public void writeTo(StreamOutput out) throws IOException {
    out.writeBoolean(eligibleToStartBasic);
}
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.joda.time.MutableDateTime;
import org.joda.time.format.DateTimeFormatter;
import org.joda.time.format.ISODateTimeFormat;

public class DateUtils {

    private static final FormatDateTimeFormatter formatDateOnlyFormatter = Joda.forPattern("yyyy-MM-dd");

    private static final DateTimeFormatter dateOnlyFormatter = formatDateOnlyFormatter.parser().withZoneUTC();

    private static final DateTimeFormatter dateTimeFormatter = ISODateTimeFormat.dateTime().withZoneUTC();

    public static long endOfTheDay(String date) {
        try {
            // Try parsing using complete date/time format
            return dateTimeFormatter.parseDateTime(date).getMillis();
        } catch (IllegalArgumentException ex) {
            // Fall back to the date only format
            MutableDateTime dateTime = dateOnlyFormatter.parseMutableDateTime(date);
            dateTime.millisOfDay().set(dateTime.millisOfDay().getMaximumValue());
            return dateTime.getMillis();
```

```java
        }
    }

    public static long beginningOfTheDay(String date) {
        try {
            // Try parsing using complete date/time format
            return dateTimeFormatter.parseDateTime(date).getMillis();
        } catch (IllegalArgumentException ex) {
            // Fall back to the date only format
            return dateOnlyFormatter.parseDateTime(date).getMillis();
        }

    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

public enum LicensesStatus {
    VALID((byte) 0),
    INVALID((byte) 1),
    EXPIRED((byte) 2);

    private final byte id;

    LicensesStatus(byte id) {
        this.id = id;
    }

    public int id() {
        return id;
    }

    public static LicensesStatus fromId(int id) {
        if (id == 0) {
            return VALID;
        } else if (id == 1) {
            return INVALID;
        } else if (id == 2) {
            return EXPIRED;
        } else {
            throw new IllegalStateException("no valid LicensesStatus for id=" + id);
        }
    }
}
```

```java
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.rest.BytesRestResponse;
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestRequest;
import org.elasticsearch.rest.RestResponse;
import org.elasticsearch.rest.action.RestBuilderListener;
import org.elasticsearch.xpack.core.XPackClient;
import org.elasticsearch.xpack.core.rest.XPackRestHandler;

import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

import static org.elasticsearch.rest.RestRequest.Method.GET;
import static org.elasticsearch.rest.RestStatus.NOT_FOUND;
import static org.elasticsearch.rest.RestStatus.OK;

public class RestGetLicenseAction extends XPackRestHandler {

    @Inject
    public RestGetLicenseAction(Settings settings, RestController controller) {
        super(settings);
        controller.registerHandler(GET,  URI_BASE + "/license", this);
    }

    @Override
    public String getName() {
        return "xpack_get_license_action";
    }

    /**
     * There will be only one license displayed per feature, the selected license will have the latest expiry_date
     * out of all other licenses for the feature.
     * <p>
     * The licenses are sorted by latest issue_date
     */
    @Override
    public RestChannelConsumer doPrepareRequest(final RestRequest request, final XPackClient client) throws
```

```java
IOException {
    final Map<String, String> overrideParams = new HashMap<>(2);
    overrideParams.put(License.REST_VIEW_MODE, "true");
    overrideParams.put(License.LICENSE_VERSION_MODE, String.valueOf(License.VERSION_CURRENT));
    final ToXContent.Params params = new ToXContent.DelegatingMapParams(overrideParams, request);
    GetLicenseRequest getLicenseRequest = new GetLicenseRequest();
    getLicenseRequest.local(request.paramAsBoolean("local", getLicenseRequest.local()));
    return channel -> client.es().admin().cluster().execute(GetLicenseAction.INSTANCE, getLicenseRequest,
        new RestBuilderListener<GetLicenseResponse>(channel) {
            @Override
            public RestResponse buildResponse(GetLicenseResponse response, XContentBuilder builder) throws
Exception {
                // Default to pretty printing, but allow ?pretty=false to disable
                if (!request.hasParam("pretty")) {
                    builder.prettyPrint().lfAtEnd();
                }
                boolean hasLicense = response.license() != null;
                builder.startObject();
                if (hasLicense) {
                    builder.startObject("license");
                    response.license().toInnerXContent(builder, params);
                    builder.endObject();
                }
                builder.endObject();
                return new BytesRestResponse(hasLicense ? OK : NOT_FOUND, builder);
            }
        });
    }

}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import java.io.IOException;
import java.io.InputStream;
import java.nio.ByteBuffer;
import java.util.ArrayList;
import java.util.Base64;
import java.util.Comparator;
import java.util.List;
import java.util.Locale;

import org.apache.lucene.util.CollectionUtil;
import org.elasticsearch.ElasticsearchException;
```

```java
import org.elasticsearch.ElasticsearchParseException;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.LoggingDeprecationHandler;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.ToXContentObject;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.common.xcontent.XContentType;

/**
 * Data structure for license. Use {@link Builder} to build a license.
 * Provides serialization/deserialization & validation methods for license object
 */
public class License implements ToXContentObject {
  public static final int VERSION_START = 1;
  public static final int VERSION_NO_FEATURE_TYPE = 2;
  public static final int VERSION_START_DATE = 3;
  public static final int VERSION_CRYPTO_ALGORITHMS = 4;
  public static final int VERSION_CURRENT = VERSION_CRYPTO_ALGORITHMS;

  /**
   * XContent param name to deserialize license(s) with
   * an additional <code>status</code> field, indicating whether a
   * particular license is 'active' or 'expired' and no signature
   * and in a human readable format
   */
  public static final String REST_VIEW_MODE = "rest_view";
  /**
   * XContent param name to deserialize license(s) with
   * no signature
   */
  public static final String LICENSE_SPEC_VIEW_MODE = "license_spec_view";
  /**
   * XContent param name to deserialize licenses according
   * to a specific license version
   */
  public static final String LICENSE_VERSION_MODE = "license_version";

  public static final Comparator<License> LATEST_ISSUE_DATE_FIRST =
Comparator.comparing(License::issueDate).reversed();

  private final int version;
  private final String uid;
```

```java
    private final String issuer;
    private final String issuedTo;
    private final long issueDate;
    private final String type;
    private final String subscriptionType;
    private final String feature;
    private final String signature;
    private final long expiryDate;
    private final long startDate;
    private final int maxNodes;
    private final OperationMode operationMode;

    /**
     * Decouples operation mode of a license from the license type value.
     * <p>
     * Note: The mode indicates features that should be made available, but it does not indicate whether the license is
active!
     *
     * The id byte is used for ordering operation modes
     */
    public enum OperationMode {
        MISSING((byte) 0),
        TRIAL((byte) 1),
        BASIC((byte) 2),
        STANDARD((byte) 3),
        GOLD((byte) 4),
        PLATINUM((byte) 5);

        private final byte id;

        OperationMode(byte id) {
            this.id = id;
        }

        /** Returns non-zero positive number when <code>opMode1</code> is greater than <code>opMode2</code>
*/
        public static int compare(OperationMode opMode1, OperationMode opMode2) {
            return Integer.compare(opMode1.id, opMode2.id);
        }

        public static OperationMode resolve(String type) {
            switch (type.toLowerCase(Locale.ROOT)) {
                case "missing":
                    return MISSING;
                case "trial":
                case "none": // bwc for 1.x subscription_type field
                case "dev": // bwc for 1.x subscription_type field
                case "development": // bwc for 1.x subscription_type field
```

```java
                return TRIAL;
            case "basic":
                return BASIC;
            case "standard":
                return STANDARD;
            case "silver":
            case "gold":
                return GOLD;
            case "platinum":
            case "cloud_internal":
            case "internal": // bwc for 1.x subscription_type field
                return PLATINUM;
            default:
                throw new IllegalArgumentException("unknown type [" + type + "]");
        }
    }
}


private License(int version, String uid, String issuer, String issuedTo, long issueDate, String type,
        String subscriptionType, String feature, String signature, long expiryDate, int maxNodes, long startDate)
{
    this.version = version;
    this.uid = uid;
    this.issuer = issuer;
    this.issuedTo = issuedTo;
    this.issueDate = issueDate;
    this.type = type;
    this.subscriptionType = subscriptionType;
    this.feature = feature;
    this.signature = signature;
    // We will validate that only a basic license can have the
BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS
    // in the validate() method.
    if (expiryDate == -1) {
        this.expiryDate = LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
    } else {
        this.expiryDate = expiryDate;
    }
    this.maxNodes = maxNodes;
    this.startDate = startDate;
    if (version == VERSION_START) {
        // in 1.x: the acceptable values for 'subscription_type': none | dev | silver | gold | platinum
        this.operationMode = OperationMode.resolve(subscriptionType);
    } else {
        // in 2.x: the acceptable values for 'type': trial | basic | silver | dev | gold | platinum
        this.operationMode = OperationMode.resolve(type);
    }
    validate();
```

```java
    }

    /**
     * @return version of the license
     */
    public int version() {
        return version;
    }

    /**
     * @return a unique identifier for a license
     */
    public String uid() {
        return uid;
    }

    /**
     * @return type of the license [trial, subscription, internal]
     */
    public String type() {
        return type;
    }

    /**
     * @return the issueDate in milliseconds
     */
    public long issueDate() {
        return issueDate;
    }

    /**
     * @return the startDate in milliseconds
     */
    public long startDate() {
        return startDate;
    }

    /**
     * @return the expiry date in milliseconds
     */
    public long expiryDate() {
        return expiryDate;
    }

    /**
     * @return the maximum number of nodes this license has been issued for
     */
    public int maxNodes() {
```

```java
        return maxNodes;
    }

    /**
     * @return a string representing the entity this licenses has been issued to
     */
    public String issuedTo() {
        return issuedTo;
    }

    /**
     * @return a string representing the entity responsible for issuing this license (internal)
     */
    public String issuer() {
        return issuer;
    }

    /**
     * @return a string representing the signature of the license used for license verification
     */
    public String signature() {
        return signature;
    }

    /**
     * @return the operation mode of the license as computed from the license type or from
     * the license mode file
     */
    public OperationMode operationMode() {
        synchronized (this) {
            if (canReadOperationModeFromFile() && operationModeFileWatcher != null) {
                return operationModeFileWatcher.getCurrentOperationMode();
            }
        }
        return operationMode;
    }

    private boolean canReadOperationModeFromFile() {
        return type.equals("cloud_internal");
    }

    private volatile OperationModeFileWatcher operationModeFileWatcher;

    /**
     * Sets the operation mode file watcher for the license and initializes the
     * file watcher when the license type allows to override operation mode from file
     */
    public synchronized void setOperationModeFileWatcher(final OperationModeFileWatcher
```

```java
operationModeFileWatcher) {
    this.operationModeFileWatcher = operationModeFileWatcher;
    if (canReadOperationModeFromFile()) {
        this.operationModeFileWatcher.init();
    }
}

/**
 * Removes operation mode file watcher, so unused license objects can be gc'ed
 */
public synchronized void removeOperationModeFileWatcher() {
    this.operationModeFileWatcher = null;
}

/**
 * @return the current license's status
 */
public Status status() {
    long now = System.currentTimeMillis();
    if (issueDate > now) {
        return Status.INVALID;
    } else if (expiryDate < now) {
        return Status.EXPIRED;
    }
    return Status.ACTIVE;
}

private void validate() {
    if (issuer == null) {
        throw new IllegalStateException("issuer can not be null");
    } else if (issuedTo == null) {
        throw new IllegalStateException("issuedTo can not be null");
    } else if (issueDate == -1) {
        throw new IllegalStateException("issueDate has to be set");
    } else if (type == null) {
        throw new IllegalStateException("type can not be null");
    } else if (subscriptionType == null && version == VERSION_START) {
        throw new IllegalStateException("subscriptionType can not be null");
    } else if (uid == null) {
        throw new IllegalStateException("uid can not be null");
    } else if (feature == null && version == VERSION_START) {
        throw new IllegalStateException("feature can not be null");
    } else if (maxNodes == -1) {
        throw new IllegalStateException("maxNodes has to be set");
    } else if (expiryDate == -1) {
        throw new IllegalStateException("expiryDate has to be set");
    } else if (expiryDate == LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS
&& "basic".equals(type) == false) {
```

```java
            throw new IllegalStateException("only basic licenses are allowed to have no expiration");
        }
    }

    public static License readLicense(StreamInput in) throws IOException {
        int version = in.readVInt(); // Version for future extensibility
        if (version > VERSION_CURRENT) {
            throw new ElasticsearchException("Unknown license version found, please upgrade all nodes to the latest
elasticsearch-license" +
                    " plugin");
        }
        Builder builder = builder();
        builder.version(version);
        builder.uid(in.readString());
        builder.type(in.readString());
        if (version == VERSION_START) {
            builder.subscriptionType(in.readString());
        }
        builder.issueDate(in.readLong());
        if (version == VERSION_START) {
            builder.feature(in.readString());
        }
        builder.expiryDate(in.readLong());
        builder.maxNodes(in.readInt());
        builder.issuedTo(in.readString());
        builder.issuer(in.readString());
        builder.signature(in.readOptionalString());
        if (version >= VERSION_START_DATE) {
            builder.startDate(in.readLong());
        }
        return builder.build();
    }

    public void writeTo(StreamOutput out) throws IOException {
        out.writeVInt(version);
        out.writeString(uid);
        out.writeString(type);
        if (version == VERSION_START) {
            out.writeString(subscriptionType);
        }
        out.writeLong(issueDate);
        if (version == VERSION_START) {
            out.writeString(feature);
        }
        out.writeLong(expiryDate);
        out.writeInt(maxNodes);
        out.writeString(issuedTo);
        out.writeString(issuer);
```

```java
    out.writeOptionalString(signature);
    if (version >= VERSION_START_DATE) {
        out.writeLong(startDate);
    }
}

@Override
public String toString() {
    try {
        final XContentBuilder builder = XContentFactory.jsonBuilder();
        toXContent(builder, ToXContent.EMPTY_PARAMS);
        return Strings.toString(builder);
    } catch (IOException e) {
        return "";
    }
}

@Override
public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
    builder.startObject();
    toInnerXContent(builder, params);
    builder.endObject();
    return builder;
}

public XContentBuilder toInnerXContent(XContentBuilder builder, Params params) throws IOException {
    boolean licenseSpecMode = params.paramAsBoolean(LICENSE_SPEC_VIEW_MODE, false);
    boolean restViewMode = params.paramAsBoolean(REST_VIEW_MODE, false);
    boolean previouslyHumanReadable = builder.humanReadable();
    if (licenseSpecMode && restViewMode) {
        throw new IllegalArgumentException("can have either " + REST_VIEW_MODE + " or " +
LICENSE_SPEC_VIEW_MODE);
    } else if (restViewMode) {
        if (!previouslyHumanReadable) {
            builder.humanReadable(true);
        }
    }
    final int version;
    if (params.param(LICENSE_VERSION_MODE) != null && restViewMode) {
        version = Integer.parseInt(params.param(LICENSE_VERSION_MODE));
    } else {
        version = this.version;
    }
    if (restViewMode) {
        builder.field(Fields.STATUS, status().label());
    }
    builder.field(Fields.UID, uid);
    builder.field(Fields.TYPE, type);
```

```java
    if (version == VERSION_START) {
        builder.field(Fields.SUBSCRIPTION_TYPE, subscriptionType);
    }
    builder.timeField(Fields.ISSUE_DATE_IN_MILLIS, Fields.ISSUE_DATE, issueDate);
    if (version == VERSION_START) {
        builder.field(Fields.FEATURE, feature);
    }

    if (expiryDate != LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS) {
        builder.timeField(Fields.EXPIRY_DATE_IN_MILLIS, Fields.EXPIRY_DATE, expiryDate);
    }
    builder.field(Fields.MAX_NODES, maxNodes);
    builder.field(Fields.ISSUED_TO, issuedTo);
    builder.field(Fields.ISSUER, issuer);
    if (!licenseSpecMode && !restViewMode && signature != null) {
        builder.field(Fields.SIGNATURE, signature);
    }
    if (restViewMode) {
        builder.humanReadable(previouslyHumanReadable);
    }
    if (version >= VERSION_START_DATE) {
        builder.timeField(Fields.START_DATE_IN_MILLIS, Fields.START_DATE, startDate);
    }
    return builder;
}

public static License fromXContent(XContentParser parser) throws IOException {
    Builder builder = new Builder();
    XContentParser.Token token;
    while ((token = parser.nextToken()) != XContentParser.Token.END_OBJECT) {
        if (token == XContentParser.Token.FIELD_NAME) {
            String currentFieldName = parser.currentName();
            token = parser.nextToken();
            if (token.isValue()) {
                if (Fields.UID.equals(currentFieldName)) {
                    builder.uid(parser.text());
                } else if (Fields.TYPE.equals(currentFieldName)) {
                    builder.type(parser.text());
                } else if (Fields.SUBSCRIPTION_TYPE.equals(currentFieldName)) {
                    builder.subscriptionType(parser.text());
                } else if (Fields.ISSUE_DATE.equals(currentFieldName)) {
                    builder.issueDate(parseDate(parser, "issue", false));
                } else if (Fields.ISSUE_DATE_IN_MILLIS.equals(currentFieldName)) {
                    builder.issueDate(parser.longValue());
                } else if (Fields.FEATURE.equals(currentFieldName)) {
                    builder.feature(parser.text());
                } else if (Fields.EXPIRY_DATE.equals(currentFieldName)) {
                    builder.expiryDate(parseDate(parser, "expiration", true));
```

```java
                } else if (Fields.EXPIRY_DATE_IN_MILLIS.equals(currentFieldName)) {
                    builder.expiryDate(parser.longValue());
                } else if (Fields.START_DATE.equals(currentFieldName)) {
                    builder.startDate(parseDate(parser, "start", false));
                } else if (Fields.START_DATE_IN_MILLIS.equals(currentFieldName)) {
                    builder.startDate(parser.longValue());
                } else if (Fields.MAX_NODES.equals(currentFieldName)) {
                    builder.maxNodes(parser.intValue());
                } else if (Fields.ISSUED_TO.equals(currentFieldName)) {
                    builder.issuedTo(parser.text());
                } else if (Fields.ISSUER.equals(currentFieldName)) {
                    builder.issuer(parser.text());
                } else if (Fields.SIGNATURE.equals(currentFieldName)) {
                    builder.signature(parser.text());
                } else if (Fields.VERSION.equals(currentFieldName)) {
                    builder.version(parser.intValue());
                }
                // Ignore unknown elements - might be new version of license
            } else if (token == XContentParser.Token.START_ARRAY) {
                // It was probably created by newer version - ignoring
                parser.skipChildren();
            } else if (token == XContentParser.Token.START_OBJECT) {
                // It was probably created by newer version - ignoring
                parser.skipChildren();
            }
        }
    }
    // not a license spec
    if (builder.signature != null) {
        byte[] signatureBytes = Base64.getDecoder().decode(builder.signature);
        ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
        int version = byteBuffer.getInt();
        // we take the absolute version, because negative versions
        // mean that the license was generated by the cluster (see TrialLicense)
        // and positive version means that the license was signed
        if (version < 0) {
            version *= -1;
        }
        if (version == 0) {
            throw new ElasticsearchException("malformed signature for license [" + builder.uid + "]");
        } else if (version > VERSION_CURRENT) {
            throw new ElasticsearchException("Unknown license version found, please upgrade all nodes to the latest
" +
                "elasticsearch-license plugin");
        }
        // signature version is the source of truth
        builder.version(version);
    }
```

```java
        return builder.build();
    }

    /**
     * Returns true if the license was auto-generated (by license plugin),
     * false otherwise
     */
    public static boolean isAutoGeneratedLicense(String signature) {
        try {
            byte[] signatureBytes = Base64.getDecoder().decode(signature);
            ByteBuffer byteBuffer = ByteBuffer.wrap(signatureBytes);
            return byteBuffer.getInt() < 0;
        } catch (IllegalArgumentException e) {
            throw new IllegalStateException(e);
        }
    }


    public static License fromSource(BytesReference bytes, XContentType xContentType) throws IOException {
        if (bytes == null || bytes.length() == 0) {
            throw new ElasticsearchParseException("failed to parse license - no content provided");
        }
        if (xContentType == null) {
            throw new ElasticsearchParseException("failed to parse license - no content-type provided");
        }
        // EMPTY is safe here because we don't call namedObject
        try (InputStream byteStream = bytes.streamInput();
             XContentParser parser = xContentType.xContent()
                 .createParser(NamedXContentRegistry.EMPTY, LoggingDeprecationHandler.INSTANCE, byteStream))
        {
            License license = null;
            if (parser.nextToken() == XContentParser.Token.START_OBJECT) {
                if (parser.nextToken() == XContentParser.Token.FIELD_NAME) {
                    String currentFieldName = parser.currentName();
                    if (Fields.LICENSES.equals(currentFieldName)) {
                        final List<License> pre20Licenses = new ArrayList<>();
                        if (parser.nextToken() == XContentParser.Token.START_ARRAY) {
                            while (parser.nextToken() != XContentParser.Token.END_ARRAY) {
                                pre20Licenses.add(License.fromXContent(parser));
                            }
                            // take the latest issued unexpired license
                            CollectionUtil.timSort(pre20Licenses, LATEST_ISSUE_DATE_FIRST);
                            long now = System.currentTimeMillis();
                            for (License oldLicense : pre20Licenses) {
                                if (oldLicense.expiryDate() > now) {
                                    license = oldLicense;
                                    break;
                                }
                            }
```

```java
                    if (license == null && !pre20Licenses.isEmpty()) {
                        license = pre20Licenses.get(0);
                    }
                } else {
                    throw new ElasticsearchParseException("failed to parse licenses expected an array of licenses");
                }
            } else if (Fields.LICENSE.equals(currentFieldName)) {
                license = License.fromXContent(parser);
            }
            // Ignore all other fields - might be created with new version
        } else {
            throw new ElasticsearchParseException("failed to parse licenses expected field");
        }
    } else {
        throw new ElasticsearchParseException("failed to parse licenses expected start object");
    }
    return license;
}
}


@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;

    License license = (License) o;

    if (issueDate != license.issueDate) return false;
    if (expiryDate != license.expiryDate) return false;
    if (startDate!= license.startDate) return false;
    if (maxNodes != license.maxNodes) return false;
    if (version != license.version) return false;
    if (uid != null ? !uid.equals(license.uid) : license.uid != null) return false;
    if (issuer != null ? !issuer.equals(license.issuer) : license.issuer != null) return false;
    if (issuedTo != null ? !issuedTo.equals(license.issuedTo) : license.issuedTo != null) return false;
    if (type != null ? !type.equals(license.type) : license.type != null) return false;
    if (subscriptionType != null ? !subscriptionType.equals(license.subscriptionType) : license.subscriptionType !=
null)
        return false;
    if (feature != null ? !feature.equals(license.feature) : license.feature != null) return false;
    return !(signature != null ? !signature.equals(license.signature) : license.signature != null);

}

@Override
public int hashCode() {
    int result = uid != null ? uid.hashCode() : 0;
    result = 31 * result + (issuer != null ? issuer.hashCode() : 0);
```

```java
        result = 31 * result + (issuedTo != null ? issuedTo.hashCode() : 0);
        result = 31 * result + (int) (issueDate ^ (issueDate >>> 32));
        result = 31 * result + (type != null ? type.hashCode() : 0);
        result = 31 * result + (subscriptionType != null ? subscriptionType.hashCode() : 0);
        result = 31 * result + (feature != null ? feature.hashCode() : 0);
        result = 31 * result + (signature != null ? signature.hashCode() : 0);
        result = 31 * result + (int) (expiryDate ^ (expiryDate >>> 32));
        result = 31 * result + (int) (startDate ^ (startDate>>> 32));
        result = 31 * result + maxNodes;
        result = 31 * result + version;
        return result;
    }

    public static final class Fields {
        public static final String STATUS = "status";
        public static final String UID = "uid";
        public static final String TYPE = "type";
        public static final String SUBSCRIPTION_TYPE = "subscription_type";
        public static final String ISSUE_DATE_IN_MILLIS = "issue_date_in_millis";
        public static final String ISSUE_DATE = "issue_date";
        public static final String FEATURE = "feature";
        public static final String EXPIRY_DATE_IN_MILLIS = "expiry_date_in_millis";
        public static final String EXPIRY_DATE = "expiry_date";
        public static final String START_DATE_IN_MILLIS = "start_date_in_millis";
        public static final String START_DATE = "start_date";
        public static final String MAX_NODES = "max_nodes";
        public static final String ISSUED_TO = "issued_to";
        public static final String ISSUER = "issuer";
        public static final String VERSION = "version";
        public static final String SIGNATURE = "signature";

        public static final String LICENSES = "licenses";
        public static final String LICENSE = "license";

    }

    private static long parseDate(XContentParser parser, String description, boolean endOfTheDay) throws
IOException {
        if (parser.currentToken() == XContentParser.Token.VALUE_NUMBER) {
            return parser.longValue();
        } else {
            try {
                if (endOfTheDay) {
                    return DateUtils.endOfTheDay(parser.text());
                } else {
                    return DateUtils.beginningOfTheDay(parser.text());
                }
            } catch (IllegalArgumentException ex) {
```

```java
                throw new ElasticsearchParseException("invalid " + description + " date format " + parser.text());
            }
        }
    }

    public static Builder builder() {
        return new Builder();
    }

    public static class Builder {
        private int version = License.VERSION_CURRENT;
        private String uid;
        private String issuer;
        private String issuedTo;
        private long issueDate = -1;
        private String type;
        private String subscriptionType;
        private String feature;
        private String signature;
        private long expiryDate = -1;
        private long startDate = -1;
        private int maxNodes = -1;

        public Builder uid(String uid) {
            this.uid = uid;
            return this;
        }

        public Builder version(int version) {
            this.version = version;
            return this;
        }

        public Builder issuer(String issuer) {
            this.issuer = issuer;
            return this;
        }

        public Builder issuedTo(String issuedTo) {
            this.issuedTo = issuedTo;
            return this;
        }

        public Builder issueDate(long issueDate) {
            this.issueDate = issueDate;
            return this;
        }
```

```java
public Builder type(String type) {
    this.type = type;
    return this;
}

public Builder subscriptionType(String subscriptionType) {
    this.subscriptionType = subscriptionType;
    return this;
}

public Builder feature(String feature) {
    this.feature = feature;
    return this;
}

public Builder expiryDate(long expiryDate) {
    this.expiryDate = expiryDate;
    return this;
}

public Builder maxNodes(int maxNodes) {
    this.maxNodes = maxNodes;
    return this;
}

public Builder signature(String signature) {
    if (signature != null) {
        this.signature = signature;
    }
    return this;
}

public Builder startDate(long startDate) {
    this.startDate = startDate;
    return this;
}

public Builder fromLicenseSpec(License license, String signature) {
    return uid(license.uid())
        .version(license.version())
        .issuedTo(license.issuedTo())
        .issueDate(license.issueDate())
        .startDate(license.startDate())
        .type(license.type())
        .subscriptionType(license.subscriptionType)
        .feature(license.feature)
        .maxNodes(license.maxNodes())
        .expiryDate(license.expiryDate())
```

```java
        .issuer(license.issuer())
        .signature(signature);
    }

    /**
     * Returns a builder that converts pre 2.0 licenses
     * to the new license format
     */
    public Builder fromPre20LicenseSpec(License pre20License) {
        return uid(pre20License.uid())
            .issuedTo(pre20License.issuedTo())
            .issueDate(pre20License.issueDate())
            .maxNodes(pre20License.maxNodes())
            .expiryDate(pre20License.expiryDate());
    }

    public License build() {
        return new License(version, uid, issuer, issuedTo, issueDate, type,
            subscriptionType, feature, signature, expiryDate, maxNodes, startDate);
    }

    public Builder validate() {
        if (issuer == null) {
            throw new IllegalStateException("issuer can not be null");
        } else if (issuedTo == null) {
            throw new IllegalStateException("issuedTo can not be null");
        } else if (issueDate == -1) {
            throw new IllegalStateException("issueDate has to be set");
        } else if (type == null) {
            throw new IllegalStateException("type can not be null");
        } else if (uid == null) {
            throw new IllegalStateException("uid can not be null");
        } else if (signature == null) {
            throw new IllegalStateException("signature can not be null");
        } else if (maxNodes == -1) {
            throw new IllegalStateException("maxNodes has to be set");
        } else if (expiryDate == -1) {
            throw new IllegalStateException("expiryDate has to be set");
        }
        return this;
    }
}

public enum Status {

    ACTIVE("active"),
    INVALID("invalid"),
    EXPIRED("expired");
```

```java
    private final String label;

    Status(String label) {
        this.label = label;
    }

    public String label() {
        return label;
    }

    public void writeTo(StreamOutput out) throws IOException {
        out.writeString(label);
    }

    public static Status readFrom(StreamInput in) throws IOException {
        String value = in.readString();
        switch (value) {
            case "active":
                return ACTIVE;
            case "invalid":
                return INVALID;
            case "expired":
                return EXPIRED;
            default:
                throw new IllegalArgumentException("unknown license status [" + value + "]");
        }
    }
}

/**
 * Returns <code>true</code> iff the license is a production licnese
 */
public boolean isProductionLicense() {
    switch (operationMode()) {
        case MISSING:
        case TRIAL:
        case BASIC:
            return false;
        case STANDARD:
        case GOLD:
        case PLATINUM:
            return true;
        default:
            throw new AssertionError("unknown operation mode: " + operationMode());

    }
}
```

```
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeReadAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportGetLicenseAction extends TransportMasterNodeReadAction<GetLicenseRequest,
GetLicenseResponse> {

    private final LicenseService licenseService;

    @Inject
    public TransportGetLicenseAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                        LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                        IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, GetLicenseAction.NAME, transportService, clusterService, threadPool, actionFilters,
            GetLicenseRequest::new, indexNameExpressionResolver);
        this.licenseService = licenseService;
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.MANAGEMENT;
    }

    @Override
    protected GetLicenseResponse newResponse() {
        return new GetLicenseResponse();
    }

    @Override
```

```java
    protected ClusterBlockException checkBlock(GetLicenseRequest request, ClusterState state) {
        return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_READ);
    }

    @Override
    protected void masterOperation(final GetLicenseRequest request, ClusterState state,
                          final ActionListener<GetLicenseResponse> listener) throws ElasticsearchException {
        listener.onResponse(new GetLicenseResponse(licenseService.getLicense()));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.apache.logging.log4j.Logger;
import org.apache.logging.log4j.message.ParameterizedMessage;
import org.apache.logging.log4j.util.Supplier;
import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.logging.Loggers;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.time.Clock;
import java.util.UUID;

public class StartupSelfGeneratedLicenseTask extends ClusterStateUpdateTask {

    /**
     * Max number of nodes licensed by generated trial license
     */
    private int selfGeneratedLicenseMaxNodes = 1000;

    private final Settings settings;
    private final Clock clock;
    private final ClusterService clusterService;
    private final Logger logger;

    public StartupSelfGeneratedLicenseTask(Settings settings, Clock clock, ClusterService clusterService) {
        this.settings = settings;
        this.clock = clock;
```

```java
        this.clusterService = clusterService;
        this.logger = Loggers.getLogger(getClass(), settings);
    }

    @Override
    public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
        LicensesMetaData licensesMetaData = newState.metaData().custom(LicensesMetaData.TYPE);
        if (logger.isDebugEnabled()) {
            logger.debug("registered self generated license: {}", licensesMetaData);
        }
    }

    @Override
    public ClusterState execute(ClusterState currentState) throws Exception {
        XPackPlugin.checkReadyForXPackCustomMetadata(currentState);
        final MetaData metaData = currentState.metaData();
        final LicensesMetaData currentLicensesMetaData = metaData.custom(LicensesMetaData.TYPE);
        // do not generate a license if any license is present
        if (currentLicensesMetaData == null) {
            String type = LicenseService.SELF_GENERATED_LICENSE_TYPE.get(settings);
            if (SelfGeneratedLicense.validSelfGeneratedType(type) == false) {
                throw new IllegalArgumentException("Illegal self generated license type [" + type +
                        "]. Must be trial or basic.");
            }
            return updateWithLicense(currentState, type);
        } else if (LicenseUtils.signatureNeedsUpdate(currentLicensesMetaData.getLicense(), currentState.nodes())) {
            return updateLicenseSignature(currentState, currentLicensesMetaData);
        } else if (LicenseUtils.licenseNeedsExtended(currentLicensesMetaData.getLicense())) {
            return extendBasic(currentState, currentLicensesMetaData);
        } else {
            return currentState;
        }
    }

    private ClusterState updateLicenseSignature(ClusterState currentState, LicensesMetaData
currentLicenseMetaData) {
        License license = currentLicenseMetaData.getLicense();
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        String type = license.type();
        long issueDate = license.issueDate();
        long expiryDate = license.expiryDate();
        // extend the basic license expiration date if needed since extendBasic will not be called now
        if ("basic".equals(type) &&  expiryDate !=
LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS) {
            expiryDate = LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
        }
        License.Builder specBuilder = License.builder()
                .uid(license.uid())
```

```java
            .issuedTo(license.issuedTo())
            .maxNodes(selfGeneratedLicenseMaxNodes)
            .issueDate(issueDate)
            .type(type)
            .expiryDate(expiryDate);
        License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder, currentState.nodes());
        Version trialVersion = currentLicenseMetaData.getMostRecentTrialVersion();
        LicensesMetaData newLicenseMetadata = new LicensesMetaData(selfGeneratedLicense, trialVersion);
        mdBuilder.putCustom(LicensesMetaData.TYPE, newLicenseMetadata);
        logger.info("Updating existing license to the new version.\n\nOld license:\n {}\n\n New license:\n{}",
            license, newLicenseMetadata.getLicense());
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
    }


    @Override
    public void onFailure(String source, @Nullable Exception e) {
        logger.error((Supplier<?>) () -> new ParameterizedMessage("unexpected failure during [{}]", source), e);
    }


    private ClusterState extendBasic(ClusterState currentState, LicensesMetaData currentLicenseMetadata) {
        License license = currentLicenseMetadata.getLicense();
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        LicensesMetaData newLicenseMetadata = createBasicLicenseFromExistingLicense(currentLicenseMetadata);
        mdBuilder.putCustom(LicensesMetaData.TYPE, newLicenseMetadata);
        logger.info("Existing basic license has an expiration. Basic licenses no longer expire." +
            "Regenerating license.\n\nOld license:\n {}\n\n New license:\n{}", license,
newLicenseMetadata.getLicense());
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
    }


    private LicensesMetaData createBasicLicenseFromExistingLicense(LicensesMetaData currentLicenseMetadata) {
        License currentLicense = currentLicenseMetadata.getLicense();
        License.Builder specBuilder = License.builder()
            .uid(currentLicense.uid())
            .issuedTo(currentLicense.issuedTo())
            .maxNodes(selfGeneratedLicenseMaxNodes)
            .issueDate(currentLicense.issueDate())
            .type("basic")
            .expiryDate(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS);
        License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder, currentLicense.version());
        Version trialVersion = currentLicenseMetadata.getMostRecentTrialVersion();
        return new LicensesMetaData(selfGeneratedLicense, trialVersion);
    }


    private ClusterState updateWithLicense(ClusterState currentState, String type) {
        long issueDate = clock.millis();
        MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
        long expiryDate;
```

```java
        if ("basic".equals(type)) {
            expiryDate = LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
        } else {
            expiryDate = issueDate +
LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.getMillis();
        }
        License.Builder specBuilder = License.builder()
                .uid(UUID.randomUUID().toString())
                .issuedTo(clusterService.getClusterName().value())
                .maxNodes(selfGeneratedLicenseMaxNodes)
                .issueDate(issueDate)
                .type(type)
                .expiryDate(expiryDate);
        License selfGeneratedLicense = SelfGeneratedLicense.create(specBuilder, currentState.nodes());
        LicensesMetaData licensesMetaData;
        if ("trial".equals(type)) {
            licensesMetaData = new LicensesMetaData(selfGeneratedLicense, Version.CURRENT);
        } else {
            licensesMetaData = new LicensesMetaData(selfGeneratedLicense, null);
        }
        mdBuilder.putCustom(LicensesMetaData.TYPE, licensesMetaData);
        return ClusterState.builder(currentState).metaData(mdBuilder).build();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.AbstractNamedDiffable;
import org.elasticsearch.cluster.MergableCustomMetaData;
import org.elasticsearch.cluster.NamedDiff;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.io.IOException;
import java.util.EnumSet;

/**
```

```
* Contains metadata about registered licenses
*/
public class LicensesMetaData extends AbstractNamedDiffable<MetaData.Custom> implements
XPackPlugin.XPackMetaDataCustom,
    MergableCustomMetaData<LicensesMetaData> {

  public static final String TYPE = "licenses";

  /**
   * When license is explicitly removed by a user, LICENSE_TOMBSTONE
   * is used as a placeholder in the license metadata. This enables
   * us to distinguish between the scenario when a cluster never
   * had a license (null) and when a license was removed explicitly
   * (LICENSE_TOMBSTONE).
   * We rely on this to decide whether to generate a unsigned trial
   * license or not. we should only generate a license if no license
   * ever existed in the cluster state
   */
  public static final License LICENSE_TOMBSTONE = License.builder()
      .type("trial")
      .issuer("elasticsearch")
      .uid("TOMBSTONE")
      .issuedTo("")
      .maxNodes(0)
      .issueDate(0)
      .expiryDate(0)
      .build();

  private License license;

  // This field describes the version of x-pack for which this cluster has exercised a trial. If the field
  // is null, then no trial has been exercised. We keep the version to leave open the possibility that we
  // may eventually allow a cluster to exercise a trial every time they upgrade to a new major version.
  @Nullable
  private Version trialVersion;

  LicensesMetaData(License license, Version trialVersion) {
    this.license = license;
    this.trialVersion = trialVersion;
  }

  public License getLicense() {
    return license;
  }

  boolean isEligibleForTrial() {
    if (trialVersion == null) {
      return true;
```

```
      }
      return Version.CURRENT.major > trialVersion.major;
   }

   Version getMostRecentTrialVersion() {
      return trialVersion;
   }

   @Override
   public String toString() {
      return "LicensesMetaData{" +
            "license=" + license +
            ", trialVersion=" + trialVersion +
            '}';
   }

   @Override
   public boolean equals(Object o) {
      if (this == o) return true;
      if (o == null || getClass() != o.getClass()) return false;

      LicensesMetaData that = (LicensesMetaData) o;

      if (license != null ? !license.equals(that.license) : that.license != null) return false;
      return trialVersion != null ? trialVersion.equals(that.trialVersion) : that.trialVersion == null;
   }

   @Override
   public int hashCode() {
      int result = license != null ? license.hashCode() : 0;
      result = 31 * result + (trialVersion != null ? trialVersion.hashCode() : 0);
      return result;
   }

   @Override
   public String getWriteableName() {
      return TYPE;
   }

   @Override
   public Version getMinimalSupportedVersion() {
      return Version.CURRENT.minimumCompatibilityVersion();
   }

   @Override
   public EnumSet<MetaData.XContentContext> context() {
      return EnumSet.of(MetaData.XContentContext.GATEWAY);
   }
```

```java
public static LicensesMetaData fromXContent(XContentParser parser) throws IOException {
    License license = LICENSE_TOMBSTONE;
    Version trialLicense = null;
    XContentParser.Token token;
    while ((token = parser.nextToken()) != XContentParser.Token.END_OBJECT) {
        if (token == XContentParser.Token.FIELD_NAME) {
            String fieldName = parser.currentName();
            if (fieldName != null) {
                if (fieldName.equals(Fields.LICENSE)) {
                    token = parser.nextToken();
                    if (token == XContentParser.Token.START_OBJECT) {
                        license = License.fromXContent(parser);
                    } else if (token == XContentParser.Token.VALUE_NULL) {
                        license = LICENSE_TOMBSTONE;
                    }
                } else if (fieldName.equals(Fields.TRIAL_LICENSE)) {
                    parser.nextToken();
                    trialLicense = Version.fromString(parser.text());
                }
            }
        }
    }
    return new LicensesMetaData(license, trialLicense);
}

@Override
public XContentBuilder toXContent(XContentBuilder builder, Params params) throws IOException {
    if (license == LICENSE_TOMBSTONE) {
        builder.nullField(Fields.LICENSE);
    } else {
        builder.startObject(Fields.LICENSE);
        license.toInnerXContent(builder, params);
        builder.endObject();
    }
    if (trialVersion != null) {
        builder.field(Fields.TRIAL_LICENSE, trialVersion.toString());
    }
    return builder;
}

@Override
public void writeTo(StreamOutput streamOutput) throws IOException {
    if (license == LICENSE_TOMBSTONE) {
        streamOutput.writeBoolean(false); // no license
    } else {
        streamOutput.writeBoolean(true); // has a license
        license.writeTo(streamOutput);
```

```
        }
        if (streamOutput.getVersion().onOrAfter(Version.V_6_1_0)) {
            if (trialVersion == null) {
                streamOutput.writeBoolean(false);
            } else {
                streamOutput.writeBoolean(true);
                Version.writeVersion(trialVersion, streamOutput);
            }
        }
    }

    public LicensesMetaData(StreamInput streamInput) throws IOException {
        if (streamInput.readBoolean()) {
            license = License.readLicense(streamInput);
        } else {
            license = LICENSE_TOMBSTONE;
        }
        if (streamInput.getVersion().onOrAfter(Version.V_6_1_0)) {
            boolean hasExercisedTrial = streamInput.readBoolean();
            if (hasExercisedTrial) {
                this.trialVersion = Version.readVersion(streamInput);
            }
        }
    }

    public static NamedDiff<MetaData.Custom> readDiffFrom(StreamInput streamInput) throws IOException {
        return readDiffFrom(MetaData.Custom.class, TYPE, streamInput);
    }

    public static License extractLicense(LicensesMetaData licensesMetaData) {
        if (licensesMetaData != null) {
            License license = licensesMetaData.getLicense();
            if (license == LicensesMetaData.LICENSE_TOMBSTONE) {
                return null;
            } else {
                return license;
            }
        }
        return null;
    }

    @Override
    public LicensesMetaData merge(LicensesMetaData other) {
        if (other.license == null) {
            return this;
        } else if (license == null
                || OperationMode.compare(other.license.operationMode(), license.operationMode()) > 0) {
            return other;
```

```
      }
      return this;
   }

   private static final class Fields {
      private static final String LICENSE = "license";
      private static final String TRIAL_LICENSE = "trial_license";
   }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

public class GetLicenseResponse extends ActionResponse {

   private License license;

   GetLicenseResponse() {
   }

   GetLicenseResponse(License license) {
      this.license = license;
   }

   public License license() {
      return license;
   }

   @Override
   public void readFrom(StreamInput in) throws IOException {
      super.readFrom(in);
      if (in.readBoolean()) {
         license = License.readLicense(in);
      }
   }

   @Override
   public void writeTo(StreamOutput out) throws IOException {
      super.writeTo(out);
```

```java
        if (license == null) {
            out.writeBoolean(false);
        } else {
            out.writeBoolean(true);
            license.writeTo(out);
        }
    }

}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportPostStartTrialAction extends TransportMasterNodeAction<PostStartTrialRequest,
PostStartTrialResponse> {

    private final LicenseService licenseService;

    @Inject
    public TransportPostStartTrialAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                            LicenseService licenseService, ThreadPool threadPool, ActionFilters actionFilters,
                            IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, PostStartTrialAction.NAME, transportService, clusterService, threadPool, actionFilters,
            indexNameExpressionResolver, PostStartTrialRequest::new);
        this.licenseService = licenseService;
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.SAME;
    }
```

```java
        @Override
        protected PostStartTrialResponse newResponse() {
            return new PostStartTrialResponse();
        }

        @Override
        protected void masterOperation(PostStartTrialRequest request, ClusterState state,
                            ActionListener<PostStartTrialResponse> listener) throws Exception {
            licenseService.startTrialLicense(request, listener);
        }

        @Override
        protected ClusterBlockException checkBlock(PostStartTrialRequest request, ClusterState state) {
            return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_WRITE);
        }
    }
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeRequest;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;

import java.io.IOException;

public class PostStartTrialRequest extends MasterNodeRequest<PostStartTrialRequest> {

    private boolean acknowledge = false;
    private String type;

    @Override
    public ActionRequestValidationException validate() {
        return null;
    }

    public PostStartTrialRequest setType(String type) {
        this.type = type;
        return this;
    }

    public String getType() {
```

```java
      return type;
   }

   public PostStartTrialRequest acknowledge(boolean acknowledge) {
      this.acknowledge = acknowledge;
      return this;
   }

   public boolean isAcknowledged() {
      return acknowledge;
   }

   @Override
   public void readFrom(StreamInput in) throws IOException {
      super.readFrom(in);
      if (in.getVersion().onOrAfter(Version.V_6_3_0)) {
         type = in.readString();
         acknowledge = in.readBoolean();
      } else {
         type = "trial";
         acknowledge = true;
      }
   }

   @Override
   public void writeTo(StreamOutput out) throws IOException {
      Version version = Version.V_6_3_0;
      if (out.getVersion().onOrAfter(version)) {
         super.writeTo(out);
         out.writeString(type);
         out.writeBoolean(acknowledge);
      } else {
         if ("trial".equals(type) == false) {
            throw new IllegalArgumentException("All nodes in cluster must be version [" + version
                  + "] or newer to start trial with a different type than 'trial'. Attempting to write to " +
                  "a node with version [" + out.getVersion() + "] with trial type [" + type + "].");
         } else if (acknowledge == false) {
            throw new IllegalArgumentException("Request must be acknowledged to send to a node with a version " +
                  "prior to [" + version + "]. Attempting to send request to node with version [" + out.getVersion() + "] "
+
                  "without acknowledgement.");
         } else {
            super.writeTo(out);
         }
      }
   }
}
```

```java
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.support.master.AcknowledgedRequestBuilder;
import org.elasticsearch.client.ElasticsearchClient;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.XContentType;

/**
 * Register license request builder
 */
public class PutLicenseRequestBuilder extends AcknowledgedRequestBuilder<PutLicenseRequest,
PutLicenseResponse, PutLicenseRequestBuilder> {

  public PutLicenseRequestBuilder(ElasticsearchClient client) {
      this(client, PutLicenseAction.INSTANCE);
  }

  /**
   * Constructs register license request
   *
   * @param client elasticsearch client
   */
  public PutLicenseRequestBuilder(ElasticsearchClient client, PutLicenseAction action) {
      super(client, action, new PutLicenseRequest());
  }

  /**
   * Sets the license
   *
   * @param license license
   * @return this builder
   */
  public PutLicenseRequestBuilder setLicense(License license) {
      request.license(license);
      return this;
  }

  public PutLicenseRequestBuilder setLicense(BytesReference licenseSource, XContentType xContentType) {
      request.license(licenseSource, xContentType);
      return this;
  }

  public PutLicenseRequestBuilder setAcknowledge(boolean acknowledge) {
```

```
    request.acknowledge(acknowledge);
    return this;
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeReadRequest;
import org.elasticsearch.common.io.stream.StreamInput;

import java.io.IOException;

public class GetBasicStatusRequest extends MasterNodeReadRequest<GetBasicStatusRequest> {

  public GetBasicStatusRequest() {
  }

  public GetBasicStatusRequest(StreamInput in) throws IOException {
    super(in);
  }

  @Override
  public ActionRequestValidationException validate() {
    return null;
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.action.support.ActionFilters;
import org.elasticsearch.action.support.master.TransportMasterNodeReadAction;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlockException;
import org.elasticsearch.cluster.block.ClusterBlockLevel;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.inject.Inject;
import org.elasticsearch.common.settings.Settings;
```

```java
import org.elasticsearch.threadpool.ThreadPool;
import org.elasticsearch.transport.TransportService;

public class TransportGetBasicStatusAction extends TransportMasterNodeReadAction<GetBasicStatusRequest,
GetBasicStatusResponse> {

    @Inject
    public TransportGetBasicStatusAction(Settings settings, TransportService transportService, ClusterService
clusterService,
                        ThreadPool threadPool, ActionFilters actionFilters,
                        IndexNameExpressionResolver indexNameExpressionResolver) {
        super(settings, GetBasicStatusAction.NAME, transportService, clusterService, threadPool, actionFilters,
            GetBasicStatusRequest::new, indexNameExpressionResolver);
    }

    @Override
    protected String executor() {
        return ThreadPool.Names.SAME;
    }

    @Override
    protected GetBasicStatusResponse newResponse() {
        return new GetBasicStatusResponse();
    }

    @Override
    protected void masterOperation(GetBasicStatusRequest request, ClusterState state,
                        ActionListener<GetBasicStatusResponse> listener) throws Exception {
        LicensesMetaData licensesMetaData = state.metaData().custom(LicensesMetaData.TYPE);
        if (licensesMetaData == null) {
            listener.onResponse(new GetBasicStatusResponse(true));
        } else {
            License license = licensesMetaData.getLicense();
            listener.onResponse(new GetBasicStatusResponse(license == null || license.type().equals("basic") == false));
        }

    }

    @Override
    protected ClusterBlockException checkBlock(GetBasicStatusRequest request, ClusterState state) {
        return state.blocks().globalBlockedException(ClusterBlockLevel.METADATA_READ);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
```

```java
package org.elasticsearch.license;

import org.elasticsearch.action.Action;

public class GetTrialStatusAction extends Action<GetTrialStatusResponse> {

  public static final GetTrialStatusAction INSTANCE = new GetTrialStatusAction();
  public static final String NAME = "cluster:admin/xpack/license/trial_status";

  private GetTrialStatusAction() {
    super(NAME);
  }

  @Override
  public GetTrialStatusResponse newResponse() {
    return new GetTrialStatusResponse();
  }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.client.ElasticsearchClient;

public class LicensingClient {

  private final ElasticsearchClient client;

  public LicensingClient(ElasticsearchClient client) {
    this.client = client;
  }

  public PutLicenseRequestBuilder preparePutLicense(License license) {
    return new PutLicenseRequestBuilder(client).setLicense(license);
  }

  public void putLicense(PutLicenseRequest request, ActionListener<PutLicenseResponse> listener) {
    client.execute(PutLicenseAction.INSTANCE, request, listener);
  }

  public GetLicenseRequestBuilder prepareGetLicense() {
    return new GetLicenseRequestBuilder(client);
  }
```

```java
    public void getLicense(GetLicenseRequest request, ActionListener<GetLicenseResponse> listener) {
        client.execute(GetLicenseAction.INSTANCE, request, listener);
    }

    public DeleteLicenseRequestBuilder prepareDeleteLicense() {
        return new DeleteLicenseRequestBuilder(client);
    }

    public void deleteLicense(DeleteLicenseRequest request, ActionListener<DeleteLicenseResponse> listener) {
        client.execute(DeleteLicenseAction.INSTANCE, request, listener);
    }

    public PostStartTrialRequestBuilder preparePostStartTrial() {
        return new PostStartTrialRequestBuilder(client, PostStartTrialAction.INSTANCE);
    }

    public GetTrialStatusRequestBuilder prepareGetStartTrial() {
        return new GetTrialStatusRequestBuilder(client, GetTrialStatusAction.INSTANCE);
    }

    public void postStartTrial(PostStartTrialRequest request, ActionListener<PostStartTrialResponse> listener) {
        client.execute(PostStartTrialAction.INSTANCE, request, listener);
    }

    public void postStartBasic(PostStartBasicRequest request, ActionListener<PostStartBasicResponse> listener) {
        client.execute(PostStartBasicAction.INSTANCE, request, listener);
    }

    public PostStartBasicRequestBuilder preparePostStartBasic() {
        return new PostStartBasicRequestBuilder(client, PostStartBasicAction.INSTANCE);
    }

    public GetBasicStatusRequestBuilder prepareGetStartBasic() {
        return new GetBasicStatusRequestBuilder(client, GetBasicStatusAction.INSTANCE);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.io.stream.StreamOutput;
import org.elasticsearch.rest.RestStatus;
```

```java
import java.io.IOException;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;

class PostStartTrialResponse extends ActionResponse {

  // Nodes Prior to 6.3 did not have NEED_ACKNOWLEDGEMENT as part of status
  enum Pre63Status {
    UPGRADED_TO_TRIAL,
    TRIAL_ALREADY_ACTIVATED;
  }
  enum Status {
    UPGRADED_TO_TRIAL(true, null, RestStatus.OK),
    TRIAL_ALREADY_ACTIVATED(false, "Operation failed: Trial was already activated.",
RestStatus.FORBIDDEN),
    NEED_ACKNOWLEDGEMENT(false,"Operation failed: Needs acknowledgement.", RestStatus.OK);

    private final boolean isTrialStarted;

    private final String errorMessage;
    private final RestStatus restStatus;
    Status(boolean isTrialStarted, String errorMessage, RestStatus restStatus) {
      this.isTrialStarted = isTrialStarted;
      this.errorMessage = errorMessage;
      this.restStatus = restStatus;
    }

    boolean isTrialStarted() {
      return isTrialStarted;
    }

    String getErrorMessage() {
      return errorMessage;
    }

    RestStatus getRestStatus() {
      return restStatus;
    }

  }

  private Status status;
  private Map<String, String[]> acknowledgeMessages;
  private String acknowledgeMessage;

  PostStartTrialResponse() {
```

```java
  }

  PostStartTrialResponse(Status status) {
     this(status, Collections.emptyMap(), null);
  }

  PostStartTrialResponse(Status status, Map<String, String[]> acknowledgeMessages, String acknowledgeMessage)
{
     this.status = status;
     this.acknowledgeMessages = acknowledgeMessages;
     this.acknowledgeMessage = acknowledgeMessage;
  }

  public Status getStatus() {
     return status;
  }

  @Override
  public void readFrom(StreamInput in) throws IOException {
     status = in.readEnum(Status.class);
     if (in.getVersion().onOrAfter(Version.V_6_3_0)) {
        acknowledgeMessage = in.readOptionalString();
        int size = in.readVInt();
        Map<String, String[]> acknowledgeMessages = new HashMap<>(size);
        for (int i = 0; i < size; i++) {
           String feature = in.readString();
           int nMessages = in.readVInt();
           String[] messages = new String[nMessages];
           for (int j = 0; j < nMessages; j++) {
              messages[j] = in.readString();
           }
           acknowledgeMessages.put(feature, messages);
        }
        this.acknowledgeMessages = acknowledgeMessages;
     } else {
        this.acknowledgeMessages = Collections.emptyMap();
     }
  }

  @Override
  public void writeTo(StreamOutput out) throws IOException {
     Version version = Version.V_6_3_0;
     if (out.getVersion().onOrAfter(version)) {
        out.writeEnum(status);
        out.writeOptionalString(acknowledgeMessage);
        out.writeVInt(acknowledgeMessages.size());
        for (Map.Entry<String, String[]> entry : acknowledgeMessages.entrySet()) {
           out.writeString(entry.getKey());
```

```java
            out.writeVInt(entry.getValue().length);
            for (String message : entry.getValue()) {
                out.writeString(message);
            }
        }
    } else {
        if (status == Status.UPGRADED_TO_TRIAL) {
            out.writeEnum(Pre63Status.UPGRADED_TO_TRIAL);
        } else if (status == Status.TRIAL_ALREADY_ACTIVATED) {
            out.writeEnum(Pre63Status.TRIAL_ALREADY_ACTIVATED);
        } else {
            throw new IllegalArgumentException("Starting trial on node with version [" + Version.CURRENT + "]
requires " +
                "acknowledgement parameter.");
        }
    }
}

Map<String, String[]> getAcknowledgementMessages() {
    return acknowledgeMessages;
}

String getAcknowledgementMessage() {
    return acknowledgeMessage;
}
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequest;
import org.elasticsearch.action.ActionResponse;
import org.elasticsearch.cluster.NamedDiff;
import org.elasticsearch.cluster.metadata.IndexNameExpressionResolver;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.common.ParseField;
import org.elasticsearch.common.io.stream.NamedWriteableRegistry;
import org.elasticsearch.common.settings.ClusterSettings;
import org.elasticsearch.common.settings.IndexScopedSettings;
import org.elasticsearch.common.settings.Setting;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.settings.SettingsFilter;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.plugins.ActionPlugin;
```

```java
import org.elasticsearch.rest.RestController;
import org.elasticsearch.rest.RestHandler;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
import java.util.function.Supplier;


public class Licensing implements ActionPlugin {

  public static final String NAME = "license";
  protected final Settings settings;

  // Until this is moved out to its own plugin (its currently in XPackPlugin.java, we need to make sure that any edits to this file
  // are also carried out in XPackClientPlugin.java
  public List<NamedWriteableRegistry.Entry> getNamedWriteables() {
    List<NamedWriteableRegistry.Entry> entries = new ArrayList<>();
    entries.add(new NamedWriteableRegistry.Entry(MetaData.Custom.class, LicensesMetaData.TYPE,
LicensesMetaData::new));
    entries.add(new NamedWriteableRegistry.Entry(NamedDiff.class, LicensesMetaData.TYPE,
LicensesMetaData::readDiffFrom));
    return entries;
  }

  // Until this is moved out to its own plugin (its currently in XPackPlugin.java, we need to make sure that any edits to this file
  // are also carried out in XPackClientPlugin.java
  public List<NamedXContentRegistry.Entry> getNamedXContent() {
    List<NamedXContentRegistry.Entry> entries = new ArrayList<>();
    // Metadata
    entries.add(new NamedXContentRegistry.Entry(MetaData.Custom.class, new
ParseField(LicensesMetaData.TYPE),
        LicensesMetaData::fromXContent));
    return entries;
  }

  public Licensing(Settings settings) {
    this.settings = settings;
  }

  @Override
  public List<ActionHandler<? extends ActionRequest, ? extends ActionResponse>> getActions() {
    return Arrays.asList(new ActionHandler<>(PutLicenseAction.INSTANCE, TransportPutLicenseAction.class),
        new ActionHandler<>(GetLicenseAction.INSTANCE, TransportGetLicenseAction.class),
        new ActionHandler<>(DeleteLicenseAction.INSTANCE, TransportDeleteLicenseAction.class),
```

```java
        new ActionHandler<>(PostStartTrialAction.INSTANCE, TransportPostStartTrialAction.class),
        new ActionHandler<>(GetTrialStatusAction.INSTANCE, TransportGetTrialStatusAction.class),
        new ActionHandler<>(PostStartBasicAction.INSTANCE, TransportPostStartBasicAction.class),
        new ActionHandler<>(GetBasicStatusAction.INSTANCE, TransportGetBasicStatusAction.class));
    }

    @Override
    public List<RestHandler> getRestHandlers(Settings settings, RestController restController, ClusterSettings clusterSettings,
            IndexScopedSettings indexScopedSettings, SettingsFilter settingsFilter, IndexNameExpressionResolver indexNameExpressionResolver,
            Supplier<DiscoveryNodes> nodesInCluster) {
        List<RestHandler> handlers = new ArrayList<>();
        handlers.add(new RestGetLicenseAction(settings, restController));
        handlers.add(new RestPutLicenseAction(settings, restController));
        handlers.add(new RestDeleteLicenseAction(settings, restController));
        handlers.add(new RestGetTrialStatus(settings, restController));
        handlers.add(new RestGetBasicStatus(settings, restController));
        handlers.add(new RestPostStartTrialLicense(settings, restController));
        handlers.add(new RestPostStartBasicLicense(settings, restController));
        return handlers;
    }

    // Until this is moved out to its own plugin (its currently in XPackPlugin.java, we need to make sure that any edits to this file
    // are also carried out in XPackClientPlugin.java
    public List<Setting<?>> getSettings() {
        // TODO convert this wildcard to a real setting
        return Collections.singletonList(Setting.groupSetting("license.", Setting.Property.NodeScope));
    }

}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionRequestValidationException;
import org.elasticsearch.action.support.master.MasterNodeReadRequest;
import org.elasticsearch.common.io.stream.StreamInput;

import java.io.IOException;


public class GetLicenseRequest extends MasterNodeReadRequest<GetLicenseRequest> {
```

```java
    public GetLicenseRequest() {
    }

    public GetLicenseRequest(StreamInput in) throws IOException {
        super(in);
    }

    @Override
    public ActionRequestValidationException validate() {
        return null;
    }
}
```

```java
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.apache.lucene.index.DirectoryReader;
import org.apache.lucene.util.Accountable;
import org.apache.lucene.util.RamUsageEstimator;
import org.apache.lucene.util.automaton.Automata;
import org.apache.lucene.util.automaton.Automaton;
import org.apache.lucene.util.automaton.CharacterRunAutomaton;
import org.apache.lucene.util.automaton.MinimizationOperations;
import org.apache.lucene.util.automaton.Operations;
import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.regex.Regex;
import org.elasticsearch.xpack.core.security.authz.accesscontrol.FieldSubsetReader;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissionsDefinition.FieldGrantExcludeGroup;
import org.elasticsearch.xpack.core.security.support.Automatons;

import java.io.IOException;
import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.stream.Collectors;

import static org.apache.lucene.util.automaton.Operations.subsetOf;

/**
* Stores patterns to fields which access is granted or denied to and maintains an automaton that can be used to check
if permission is
* allowed for a specific field.
```

```
 * Field permissions are configured via a list of strings that are patterns a field has to match. Two lists determine
whether or
 * not a field is granted access to:
 * 1. It has to match the patterns in grantedFieldsArray
 * 2. it must not match the patterns in deniedFieldsArray
 */
public final class FieldPermissions implements Accountable {

  public static final FieldPermissions DEFAULT = new FieldPermissions();

  private static final long BASE_FIELD_PERM_DEF_BYTES = RamUsageEstimator.shallowSizeOf(new
FieldPermissionsDefinition(null, null));
  private static final long BASE_FIELD_GROUP_BYTES = RamUsageEstimator.shallowSizeOf(new
FieldGrantExcludeGroup(null, null));
  private static final long BASE_HASHSET_SIZE = RamUsageEstimator.shallowSizeOfInstance(HashSet.class);
  private static final long BASE_HASHSET_ENTRY_SIZE;
  static {
    HashMap<String, Object> map = new HashMap<>();
    map.put(FieldPermissions.class.getName(), new Object());
    long mapEntryShallowSize = RamUsageEstimator.shallowSizeOf(map.entrySet().iterator().next());
    // assume a load factor of 50%
    // for each entry, we need two object refs, one for the entry itself
    // and one for the free space that is due to the fact hash tables can
    // not be fully loaded
    BASE_HASHSET_ENTRY_SIZE = mapEntryShallowSize + 2 *
RamUsageEstimator.NUM_BYTES_OBJECT_REF;
  }

  private final FieldPermissionsDefinition fieldPermissionsDefinition;
  // an automaton that represents a union of one more sets of permitted and denied fields
  private final CharacterRunAutomaton permittedFieldsAutomaton;
  private final boolean permittedFieldsAutomatonIsTotal;
  private final Automaton originalAutomaton;

  private final long ramBytesUsed;

  /** Constructor that does not enable field-level security: all fields are accepted. */
  public FieldPermissions() {
    this(new FieldPermissionsDefinition(null, null), Automatons.MATCH_ALL);
  }

  /** Constructor that enables field-level security based on include/exclude rules. Exclude rules
   *  have precedence over include rules. */
  public FieldPermissions(FieldPermissionsDefinition fieldPermissionsDefinition) {
    this(fieldPermissionsDefinition, initializePermittedFieldsAutomaton(fieldPermissionsDefinition));
  }

  /** Constructor that enables field-level security based on include/exclude rules. Exclude rules
```

```java
 *  have precedence over include rules. */
FieldPermissions(FieldPermissionsDefinition fieldPermissionsDefinition, Automaton permittedFieldsAutomaton)
{
    if (permittedFieldsAutomaton.isDeterministic() == false && permittedFieldsAutomaton.getNumStates() > 1) {
        // we only accept deterministic automata so that the CharacterRunAutomaton constructor
        // directly wraps the provided automaton
        throw new IllegalArgumentException("Only accepts deterministic automata");
    }
    this.fieldPermissionsDefinition = fieldPermissionsDefinition;
    this.originalAutomaton = permittedFieldsAutomaton;
    this.permittedFieldsAutomaton = new CharacterRunAutomaton(permittedFieldsAutomaton);
    // we cache the result of isTotal since this might be a costly operation
    this.permittedFieldsAutomatonIsTotal = Operations.isTotal(permittedFieldsAutomaton);

    long ramBytesUsed = BASE_FIELD_PERM_DEF_BYTES;

    for (FieldGrantExcludeGroup group : fieldPermissionsDefinition.getFieldGrantExcludeGroups()) {
        ramBytesUsed += BASE_FIELD_GROUP_BYTES + BASE_HASHSET_ENTRY_SIZE;
        if (group.getGrantedFields() != null) {
            ramBytesUsed += RamUsageEstimator.shallowSizeOf(group.getGrantedFields());
        }
        if (group.getExcludedFields() != null) {
            ramBytesUsed += RamUsageEstimator.shallowSizeOf(group.getExcludedFields());
        }
    }
    ramBytesUsed += permittedFieldsAutomaton.ramBytesUsed();
    ramBytesUsed += runAutomatonRamBytesUsed(permittedFieldsAutomaton);
    this.ramBytesUsed = ramBytesUsed;
}

/**
 * Return an estimation of the ram bytes used by a {@link CharacterRunAutomaton}
 * that wraps the given automaton.
 */
private static long runAutomatonRamBytesUsed(Automaton a) {
    return a.getNumStates() * 5; // wild guess, better than 0
}

public static Automaton initializePermittedFieldsAutomaton(FieldPermissionsDefinition
fieldPermissionsDefinition) {
    Set<FieldGrantExcludeGroup> groups = fieldPermissionsDefinition.getFieldGrantExcludeGroups();
    assert groups.size() > 0 : "there must always be a single group for field inclusion/exclusion";
    List<Automaton> automatonList =
            groups.stream()
                    .map(g -> FieldPermissions.initializePermittedFieldsAutomaton(g.getGrantedFields(),
g.getExcludedFields()))
                    .collect(Collectors.toList());
    return Automatons.unionAndMinimize(automatonList);
```

```
        }

    private static Automaton initializePermittedFieldsAutomaton(final String[] grantedFields, final String[]
deniedFields) {
        Automaton grantedFieldsAutomaton;
        if (grantedFields == null || Arrays.stream(grantedFields).anyMatch(Regex::isMatchAllPattern)) {
            grantedFieldsAutomaton = Automatons.MATCH_ALL;
        } else {
            // an automaton that includes metadata fields, including join fields created by the _parent field such
            // as _parent#type
            Automaton metaFieldsAutomaton = Operations.concatenate(Automata.makeChar('_'),
Automata.makeAnyString());
            grantedFieldsAutomaton = Operations.union(Automatons.patterns(grantedFields), metaFieldsAutomaton);
        }

        Automaton deniedFieldsAutomaton;
        if (deniedFields == null || deniedFields.length == 0) {
            deniedFieldsAutomaton = Automatons.EMPTY;
        } else {
            deniedFieldsAutomaton = Automatons.patterns(deniedFields);
        }

        grantedFieldsAutomaton = MinimizationOperations.minimize(grantedFieldsAutomaton,
Operations.DEFAULT_MAX_DETERMINIZED_STATES);
        deniedFieldsAutomaton = MinimizationOperations.minimize(deniedFieldsAutomaton,
Operations.DEFAULT_MAX_DETERMINIZED_STATES);

        if (subsetOf(deniedFieldsAutomaton, grantedFieldsAutomaton) == false) {
            throw new ElasticsearchSecurityException("Exceptions for field permissions must be a subset of the " +
                    "granted fields but " + Strings.arrayToCommaDelimitedString(deniedFields) + " is not a subset of " +
                    Strings.arrayToCommaDelimitedString(grantedFields));
        }

        grantedFieldsAutomaton = Automatons.minusAndMinimize(grantedFieldsAutomaton,
deniedFieldsAutomaton);
        return grantedFieldsAutomaton;
    }

    /**
     * Returns true if this field permission policy allows access to the field and false if not.
     * fieldName can be a wildcard.
     */
    public boolean grantsAccessTo(String fieldName) {
        return permittedFieldsAutomatonIsTotal || permittedFieldsAutomaton.run(fieldName);
    }

    FieldPermissionsDefinition getFieldPermissionsDefinition() {
        return fieldPermissionsDefinition;
```

```java
    }

    /** Return whether field-level security is enabled, ie. whether any field might be filtered out. */
    public boolean hasFieldLevelSecurity() {
        return permittedFieldsAutomatonIsTotal == false;
    }

    /** Return a wrapped reader that only exposes allowed fields. */
    public DirectoryReader filter(DirectoryReader reader) throws IOException {
        if (hasFieldLevelSecurity() == false) {
            return reader;
        }
        return FieldSubsetReader.wrap(reader, permittedFieldsAutomaton);
    }

    // for testing only
    Automaton getIncludeAutomaton() {
        return originalAutomaton;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;

        FieldPermissions that = (FieldPermissions) o;

        if (permittedFieldsAutomatonIsTotal != that.permittedFieldsAutomatonIsTotal) return false;
        return fieldPermissionsDefinition != null ?
                fieldPermissionsDefinition.equals(that.fieldPermissionsDefinition) : that.fieldPermissionsDefinition ==
null;
    }

    @Override
    public int hashCode() {
        int result = fieldPermissionsDefinition != null ? fieldPermissionsDefinition.hashCode() : 0;
        result = 31 * result + (permittedFieldsAutomatonIsTotal ? 1 : 0);
        return result;
    }

    @Override
    public long ramBytesUsed() {
        return ramBytesUsed;
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
```

package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.xpack.core.security.authz.privilege.ClusterPrivilege;

import java.util.function.Predicate;

/**
* A permission that is based on privileges for cluster wide actions
*/
public final class ClusterPermission {

    public static final ClusterPermission NONE = new ClusterPermission(ClusterPrivilege.NONE);

    private final ClusterPrivilege privilege;
    private final Predicate<String> predicate;

    ClusterPermission(ClusterPrivilege privilege) {
        this.privilege = privilege;
        this.predicate = privilege.predicate();
    }

    public ClusterPrivilege privilege() {
        return privilege;
    }

    public boolean check(String action) {
        return predicate.test(action);
    }
}
package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.util.set.Sets;
import org.elasticsearch.xpack.core.security.authz.RoleDescriptor;
import org.elasticsearch.xpack.core.security.authz.accesscontrol.IndicesAccessControl;
import org.elasticsearch.xpack.core.security.authz.privilege.ClusterPrivilege;
import org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege;
import org.elasticsearch.xpack.core.security.authz.privilege.Privilege;

```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.Set;

public final class Role {

  public static final Role EMPTY = Role.builder("__empty").build();

  private final String[] names;
  private final ClusterPermission cluster;
  private final IndicesPermission indices;
  private final RunAsPermission runAs;

  Role(String[] names, ClusterPermission cluster, IndicesPermission indices, RunAsPermission runAs) {
    this.names = names;
    this.cluster = Objects.requireNonNull(cluster);
    this.indices = Objects.requireNonNull(indices);
    this.runAs = Objects.requireNonNull(runAs);
  }

  public String[] names() {
    return names;
  }

  public ClusterPermission cluster() {
    return cluster;
  }

  public IndicesPermission indices() {
    return indices;
  }

  public RunAsPermission runAs() {
    return runAs;
  }

  public static Builder builder(String... names) {
    return new Builder(names, null);
  }

  public static Builder builder(String[] names, FieldPermissionsCache fieldPermissionsCache) {
    return new Builder(names, fieldPermissionsCache);
  }

  public static Builder builder(RoleDescriptor rd, FieldPermissionsCache fieldPermissionsCache) {
```

```java
            return new Builder(rd, fieldPermissionsCache);
        }


        /**
         * Returns whether at least one group encapsulated by this indices permissions is authorized to execute the
         * specified action with the requested indices/aliases. At the same time if field and/or document level security
         * is configured for any group also the allowed fields and role queries are resolved.
         */
        public IndicesAccessControl authorize(String action, Set<String> requestedIndicesOrAliases, MetaData metaData,
                                    FieldPermissionsCache fieldPermissionsCache) {
            Map<String, IndicesAccessControl.IndexAccessControl> indexPermissions = indices.authorize(
                action, requestedIndicesOrAliases, metaData, fieldPermissionsCache
            );

            // At least one role / indices permission set need to match with all the requested indices/aliases:
            boolean granted = true;
            for (Map.Entry<String, IndicesAccessControl.IndexAccessControl> entry : indexPermissions.entrySet()) {
                if (!entry.getValue().isGranted()) {
                    granted = false;
                    break;
                }
            }
            return new IndicesAccessControl(granted, indexPermissions);
        }

        public static class Builder {

            private final String[] names;
            private ClusterPermission cluster = ClusterPermission.NONE;
            private RunAsPermission runAs = RunAsPermission.NONE;
            private List<IndicesPermission.Group> groups = new ArrayList<>();
            private FieldPermissionsCache fieldPermissionsCache = null;

            private Builder(String[] names, FieldPermissionsCache fieldPermissionsCache) {
                this.names = names;
                this.fieldPermissionsCache = fieldPermissionsCache;
            }

            private Builder(RoleDescriptor rd, @Nullable FieldPermissionsCache fieldPermissionsCache) {
                this.names = new String[] { rd.getName() };
                this.fieldPermissionsCache = fieldPermissionsCache;
                if (rd.getClusterPrivileges().length == 0) {
                    cluster = ClusterPermission.NONE;
                } else {
                    this.cluster(ClusterPrivilege.get(Sets.newHashSet(rd.getClusterPrivileges())));
                }
                groups.addAll(convertFromIndicesPrivileges(rd.getIndicesPrivileges(), fieldPermissionsCache));
```

```
        String[] rdRunAs = rd.getRunAs();
        if (rdRunAs != null && rdRunAs.length > 0) {
            this.runAs(new Privilege(Sets.newHashSet(rdRunAs), rdRunAs));
        }
    }

    public Builder cluster(ClusterPrivilege privilege) {
        cluster = new ClusterPermission(privilege);
        return this;
    }

    public Builder runAs(Privilege privilege) {
        runAs = new RunAsPermission(privilege);
        return this;
    }

    public Builder add(IndexPrivilege privilege, String... indices) {
        groups.add(new IndicesPermission.Group(privilege, FieldPermissions.DEFAULT, null, indices));
        return this;
    }

    public Builder add(FieldPermissions fieldPermissions, Set<BytesReference> query, IndexPrivilege privilege,
String... indices) {
        groups.add(new IndicesPermission.Group(privilege, fieldPermissions, query, indices));
        return this;
    }

    public Role build() {
        IndicesPermission indices = groups.isEmpty() ? IndicesPermission.NONE :
            new IndicesPermission(groups.toArray(new IndicesPermission.Group[groups.size()]));
        return new Role(names, cluster, indices, runAs);
    }

    static List<IndicesPermission.Group> convertFromIndicesPrivileges(RoleDescriptor.IndicesPrivileges[]
indicesPrivileges,
                                                @Nullable FieldPermissionsCache fieldPermissionsCache) {
        List<IndicesPermission.Group> list = new ArrayList<>(indicesPrivileges.length);
        for (RoleDescriptor.IndicesPrivileges privilege : indicesPrivileges) {
            final FieldPermissions fieldPermissions;
            if (fieldPermissionsCache != null) {
                fieldPermissions = fieldPermissionsCache.getFieldPermissions(privilege.getGrantedFields(),
privilege.getDeniedFields());
            } else {
                fieldPermissions = new FieldPermissions(
                    new FieldPermissionsDefinition(privilege.getGrantedFields(), privilege.getDeniedFields()));
            }
            final Set<BytesReference> query = privilege.getQuery() == null ? null :
Collections.singleton(privilege.getQuery());
```

```
            list.add(new IndicesPermission.Group(IndexPrivilege.get(Sets.newHashSet(privilege.getPrivileges())),
                fieldPermissions,
                query,
                privilege.getIndices())));

        }
        return list;
    }
  }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import org.apache.lucene.util.automaton.Automaton;
import org.apache.lucene.util.automaton.TooComplexToDeterminizeException;
import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.cluster.metadata.AliasOrIndex;
import org.elasticsearch.cluster.metadata.IndexMetaData;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.logging.Loggers;
import org.elasticsearch.xpack.core.security.authz.accesscontrol.IndicesAccessControl;
import org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege;
import org.elasticsearch.xpack.core.security.support.Automatons;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.Set;
import java.util.SortedMap;
import java.util.concurrent.ConcurrentHashMap;
import java.util.function.Function;
import java.util.function.Predicate;

import static java.util.Collections.unmodifiableMap;
import static java.util.Collections.unmodifiableSet;
```

```java
/**
 * A permission that is based on privileges for index related actions executed
 * on specific indices
 */
public final class IndicesPermission implements Iterable<IndicesPermission.Group> {

    public static final IndicesPermission NONE = new IndicesPermission();

    private final Function<String, Predicate<String>> loadingFunction;

    private final ConcurrentHashMap<String, Predicate<String>> allowedIndicesMatchersForAction = new
ConcurrentHashMap<>();

    private final Group[] groups;

    public IndicesPermission(Group... groups) {
        this.groups = groups;
        loadingFunction = (action) -> {
            List<String> indices = new ArrayList<>();
            for (Group group : groups) {
                if (group.actionMatcher.test(action)) {
                    indices.addAll(Arrays.asList(group.indices));
                }
            }
            return indexMatcher(indices);
        };
    }

    static Predicate<String> indexMatcher(List<String> indices) {
        try {
            return Automatons.predicate(indices);
        } catch (TooComplexToDeterminizeException e) {
            Loggers.getLogger(IndicesPermission.class).debug("Index pattern automaton [{}] is too complex", indices);
            String description = Strings.collectionToCommaDelimitedString(indices);
            if (description.length() > 80) {
                description = Strings.cleanTruncate(description, 80) + "...";
            }
            throw new ElasticsearchSecurityException("The set of permitted index patterns [{}] is too complex to
evaluate", e, description);
        }
    }

    @Override
    public Iterator<Group> iterator() {
        return Arrays.asList(groups).iterator();
    }

    public Group[] groups() {
```

```java
        return groups;
    }

    /**
     * @return A predicate that will match all the indices that this permission
     * has the privilege for executing the given action on.
     */
    public Predicate<String> allowedIndicesMatcher(String action) {
        return allowedIndicesMatchersForAction.computeIfAbsent(action, loadingFunction);
    }

    /**
     * Checks if the permission matches the provided action, without looking at indices.
     * To be used in very specific cases where indices actions need to be authorized regardless of their indices.
     * The usecase for this is composite actions that are initially only authorized based on the action name (indices are not
     * checked on the coordinating node), and properly authorized later at the shard level checking their indices as well.
     */
    public boolean check(String action) {
        for (Group group : groups) {
            if (group.check(action)) {
                return true;
            }
        }
        return false;
    }

    public Automaton allowedActionsMatcher(String index) {
        List<Automaton> automatonList = new ArrayList<>();
        for (Group group : groups) {
            if (group.indexNameMatcher.test(index)) {
                automatonList.add(group.privilege.getAutomaton());
            }
        }
        return automatonList.isEmpty() ? Automatons.EMPTY : Automatons.unionAndMinimize(automatonList);
    }

    /**
     * Authorizes the provided action against the provided indices, given the current cluster metadata
     */
    public Map<String, IndicesAccessControl.IndexAccessControl> authorize(String action, Set<String> requestedIndicesOrAliases,
                                                   MetaData metaData, FieldPermissionsCache fieldPermissionsCache) {
        // now... every index that is associated with the request, must be granted
        // by at least one indices permission group

        SortedMap<String, AliasOrIndex> allAliasesAndIndices = metaData.getAliasAndIndexLookup();
```

```java
        Map<String, Set<FieldPermissions>> fieldPermissionsByIndex = new HashMap<>();
        Map<String, DocumentLevelPermissions> roleQueriesByIndex = new HashMap<>();
        Map<String, Boolean> grantedBuilder = new HashMap<>();

        for (String indexOrAlias : requestedIndicesOrAliases) {
            boolean granted = false;
            Set<String> concreteIndices = new HashSet<>();
            AliasOrIndex aliasOrIndex = allAliasesAndIndices.get(indexOrAlias);
            if (aliasOrIndex != null) {
                for (IndexMetaData indexMetaData : aliasOrIndex.getIndices()) {
                    concreteIndices.add(indexMetaData.getIndex().getName());
                }
            }


            for (Group group : groups) {
                if (group.check(action, indexOrAlias)) {
                    granted = true;
                    for (String index : concreteIndices) {
                        Set<FieldPermissions> fieldPermissions = fieldPermissionsByIndex.computeIfAbsent(index, (k) ->
new HashSet<>());
                        fieldPermissionsByIndex.put(indexOrAlias, fieldPermissions);
                        fieldPermissions.add(group.getFieldPermissions());
                        DocumentLevelPermissions permissions =
                            roleQueriesByIndex.computeIfAbsent(index, (k) -> new DocumentLevelPermissions());
                        roleQueriesByIndex.putIfAbsent(indexOrAlias, permissions);
                        if (group.hasQuery()) {
                            permissions.addAll(group.getQuery());
                        } else {
                            // if more than one permission matches for a concrete index here and if
                            // a single permission doesn't have a role query then DLS will not be
                            // applied even when other permissions do have a role query
                            permissions.setAllowAll(true);
                        }
                    }
                }
            }


            if (concreteIndices.isEmpty()) {
                grantedBuilder.put(indexOrAlias, granted);
            } else {
                grantedBuilder.put(indexOrAlias, granted);
                for (String concreteIndex : concreteIndices) {
                    grantedBuilder.put(concreteIndex, granted);
                }
            }
        }

        Map<String, IndicesAccessControl.IndexAccessControl> indexPermissions = new HashMap<>();
```

```java
        for (Map.Entry<String, Boolean> entry : grantedBuilder.entrySet()) {
            String index = entry.getKey();
            DocumentLevelPermissions permissions = roleQueriesByIndex.get(index);
            final Set<BytesReference> roleQueries;
            if (permissions != null && permissions.isAllowAll() == false) {
                roleQueries = unmodifiableSet(permissions.queries);
            } else {
                roleQueries = null;
            }

            final FieldPermissions fieldPermissions;
            final Set<FieldPermissions> indexFieldPermissions = fieldPermissionsByIndex.get(index);
            if (indexFieldPermissions != null && indexFieldPermissions.isEmpty() == false) {
                fieldPermissions = indexFieldPermissions.size() == 1 ? indexFieldPermissions.iterator().next() :
                        fieldPermissionsCache.getFieldPermissions(indexFieldPermissions);
            } else {
                fieldPermissions = FieldPermissions.DEFAULT;
            }
            indexPermissions.put(index, new IndicesAccessControl.IndexAccessControl(entry.getValue(),
fieldPermissions, roleQueries));
        }
        return unmodifiableMap(indexPermissions);
    }

    public static class Group {
        private final IndexPrivilege privilege;
        private final Predicate<String> actionMatcher;
        private final String[] indices;
        private final Predicate<String> indexNameMatcher;

        public FieldPermissions getFieldPermissions() {
            return fieldPermissions;
        }

        private final FieldPermissions fieldPermissions;
        private final Set<BytesReference> query;

        public Group(IndexPrivilege privilege, FieldPermissions fieldPermissions, @Nullable Set<BytesReference>
query, String... indices) {
            assert indices.length != 0;
            this.privilege = privilege;
            this.actionMatcher = privilege.predicate();
            this.indices = indices;
            this.indexNameMatcher = indexMatcher(Arrays.asList(indices));
            this.fieldPermissions = Objects.requireNonNull(fieldPermissions);
            this.query = query;
        }
```

```java
public IndexPrivilege privilege() {
    return privilege;
}

public String[] indices() {
    return indices;
}

@Nullable
public Set<BytesReference> getQuery() {
    return query;
}

private boolean check(String action) {
    return actionMatcher.test(action);
}

private boolean check(String action, String index) {
    assert index != null;
    return check(action) && indexNameMatcher.test(index);
}

boolean hasQuery() {
    return query != null;
}
}

private static class DocumentLevelPermissions {

    private Set<BytesReference> queries = null;
    private boolean allowAll = false;

    private void addAll(Set<BytesReference> query) {
        if (allowAll == false) {
            if (queries == null) {
                queries = new HashSet<>();
            }
            queries.addAll(query);
        }
    }

    private boolean isAllowAll() {
        return allowAll;
    }

    private void setAllowAll(boolean allowAll) {
        this.allowAll = allowAll;
    }
```

```
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.xpack.core.security.authz.permission;

import java.util.Arrays;
import java.util.Collections;
import java.util.Set;

/**
* Represents the definition of a {@link FieldPermissions}. Field permissions are defined as a
* collections of grant and exclude definitions where the exclude definition must be a subset of
* the grant definition.
*/
public final class FieldPermissionsDefinition {

  private final Set<FieldGrantExcludeGroup> fieldGrantExcludeGroups;

  public FieldPermissionsDefinition(String[] grant, String[] exclude) {
      this(Collections.singleton(new FieldGrantExcludeGroup(grant, exclude)));
  }

  public FieldPermissionsDefinition(Set<FieldGrantExcludeGroup> fieldGrantExcludeGroups) {
      this.fieldGrantExcludeGroups = Collections.unmodifiableSet(fieldGrantExcludeGroups);
  }

  public Set<FieldGrantExcludeGroup> getFieldGrantExcludeGroups() {
      return fieldGrantExcludeGroups;
  }

  @Override
  public boolean equals(Object o) {
      if (this == o) return true;
      if (o == null || getClass() != o.getClass()) return false;

      FieldPermissionsDefinition that = (FieldPermissionsDefinition) o;

      return fieldGrantExcludeGroups != null ?
            fieldGrantExcludeGroups.equals(that.fieldGrantExcludeGroups) :
            that.fieldGrantExcludeGroups == null;
  }

  @Override
  public int hashCode() {
```

```java
        return fieldGrantExcludeGroups != null ? fieldGrantExcludeGroups.hashCode() : 0;
    }

    public static final class FieldGrantExcludeGroup {
        private final String[] grantedFields;
        private final String[] excludedFields;

        public FieldGrantExcludeGroup(String[] grantedFields, String[] excludedFields) {
            this.grantedFields = grantedFields;
            this.excludedFields = excludedFields;
        }

        public String[] getGrantedFields() {
            return grantedFields;
        }

        public String[] getExcludedFields() {
            return excludedFields;
        }

        @Override
        public boolean equals(Object o) {
            if (this == o) return true;
            if (o == null || getClass() != o.getClass()) return false;

            FieldGrantExcludeGroup that = (FieldGrantExcludeGroup) o;

            if (!Arrays.equals(grantedFields, that.grantedFields)) return false;
            return Arrays.equals(excludedFields, that.excludedFields);
        }

        @Override
        public int hashCode() {
            int result = Arrays.hashCode(grantedFields);
            result = 31 * result + Arrays.hashCode(excludedFields);
            return result;
        }
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.xpack.core.security.authz.privilege.Privilege;
```

```java
import java.util.function.Predicate;

/**
 * A permissions that is based on a general privilege that contains patterns of users that this
 * user can execute a request as
 */
public final class RunAsPermission {

    public static final RunAsPermission NONE = new RunAsPermission(Privilege.NONE);

    private final Predicate<String> predicate;

    RunAsPermission(Privilege privilege) {
        this.predicate = privilege.predicate();
    }

    /**
     * Checks if this permission grants run as to the specified user
     */
    public boolean check(String username) {
        return predicate.test(username);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.xpack.core.security.authz.permission;

import org.apache.lucene.util.automaton.Automaton;
import org.elasticsearch.ElasticsearchException;
import org.elasticsearch.common.cache.Cache;
import org.elasticsearch.common.cache.CacheBuilder;
import org.elasticsearch.common.settings.Setting;
import org.elasticsearch.common.settings.Setting.Property;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissionsDefinition.FieldGrantExcludeGroup;
import org.elasticsearch.xpack.core.security.support.Automatons;

import java.util.Collection;
import java.util.List;
import java.util.Optional;
import java.util.Set;
import java.util.concurrent.ExecutionException;
import java.util.function.Predicate;
import java.util.stream.Collectors;
```

```java
import static org.elasticsearch.xpack.core.security.SecurityField.setting;

/**
 * A service for managing the caching of {@link FieldPermissions} as these may often need to be combined or
created and internally they
 * use an {@link org.apache.lucene.util.automaton.Automaton}, which can be costly to create once you account for
minimization
 */
public final class FieldPermissionsCache {

    public static final Setting<Long> CACHE_SIZE_SETTING = Setting.longSetting(
            setting("authz.store.roles.field_permissions.cache.max_size_in_bytes"), 100 * 1024 * 1024, -1L,
Property.NodeScope);
    private final Cache<FieldPermissionsDefinition, FieldPermissions> cache;

    public FieldPermissionsCache(Settings settings) {
        this.cache = CacheBuilder.<FieldPermissionsDefinition, FieldPermissions>builder()
            .setMaximumWeight(CACHE_SIZE_SETTING.get(settings))
            .weigher((key, fieldPermissions) -> fieldPermissions.ramBytesUsed())
            .build();
    }

    /**
     * Gets a {@link FieldPermissions} instance that corresponds to the granted and denied parameters. The instance
may come from the cache
     * or if it gets created, the instance will be cached
     */
    FieldPermissions getFieldPermissions(String[] granted, String[] denied) {
        return getFieldPermissions(new FieldPermissionsDefinition(granted, denied));
    }

    /**
     * Gets a {@link FieldPermissions} instance that corresponds to the granted and denied parameters. The instance
may come from the cache
     * or if it gets created, the instance will be cached
     */
    public FieldPermissions getFieldPermissions(FieldPermissionsDefinition fieldPermissionsDefinition) {
        try {
            return cache.computeIfAbsent(fieldPermissionsDefinition,
                    (key) -> new FieldPermissions(key, FieldPermissions.initializePermittedFieldsAutomaton(key)));
        } catch (ExecutionException e) {
            throw new ElasticsearchException("unable to compute field permissions", e);
        }
    }

    /**
     * Returns a field permissions object that corresponds to the merging of the given field permissions and caches the
instance if one was
```

```
     * not found in the cache.
     */
    FieldPermissions getFieldPermissions(Collection<FieldPermissions> fieldPermissionsCollection) {
        Optional<FieldPermissions> allowAllFieldPermissions = fieldPermissionsCollection.stream()
            .filter(((Predicate<FieldPermissions>) (FieldPermissions::hasFieldLevelSecurity)).negate())
            .findFirst();
        return allowAllFieldPermissions.orElseGet(() -> {
            final Set<FieldGrantExcludeGroup> fieldGrantExcludeGroups = fieldPermissionsCollection.stream()
                .flatMap(fieldPermission ->
fieldPermission.getFieldPermissionsDefinition().getFieldGrantExcludeGroups().stream())
                .collect(Collectors.toSet());
            final FieldPermissionsDefinition combined = new FieldPermissionsDefinition(fieldGrantExcludeGroups);
            try {
                return cache.computeIfAbsent(combined, (key) -> {
                    List<Automaton> automatonList = fieldPermissionsCollection.stream()
                            .map(FieldPermissions::getIncludeAutomaton)
                            .collect(Collectors.toList());
                    return new FieldPermissions(key, Automatons.unionAndMinimize(automatonList));
                });
            } catch (ExecutionException e) {
                throw new ElasticsearchException("unable to compute field permissions", e);
            }
        });
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.analysis.common.CommonAnalysisPlugin;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.env.Environment;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase.ClusterScope;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackPlugin;

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.Arrays;
import java.util.Collection;
```

```java
import static org.elasticsearch.test.ESIntegTestCase.Scope.TEST;
import static org.hamcrest.CoreMatchers.equalTo;
import static org.hamcrest.CoreMatchers.nullValue;

@ClusterScope(scope = TEST, numDataNodes = 0, numClientNodes = 0, maxNumDataNodes = 0,
transportClientRatio = 0)
public class LicenseServiceClusterTests extends AbstractLicensesIntegrationTestCase {

  @Override
  protected Settings nodeSettings(int nodeOrdinal) {
    return nodeSettingsBuilder(nodeOrdinal).build();
  }

  @Override
  protected boolean addMockHttpTransport() {
    return false; // enable http
  }

  private Settings.Builder nodeSettingsBuilder(int nodeOrdinal) {
    return Settings.builder()
        .put(super.nodeSettings(nodeOrdinal))
        .put("node.data", true)
        .put("resource.reload.interval.high", "500ms"); // for license mode file watcher
  }

  @Override
  protected Collection<Class<? extends Plugin>> nodePlugins() {
    return Arrays.asList(LocalStateCompositeXPackPlugin.class, CommonAnalysisPlugin.class,
Netty4Plugin.class);
  }

  @Override
  protected Collection<Class<? extends Plugin>> transportClientPlugins() {
    return nodePlugins();
  }

  public void testClusterRestartWithLicense() throws Exception {
    wipeAllLicenses();

    int numNodes = randomIntBetween(1, 5);
    logger.info("--> starting {} node(s)", numNodes);
    internalCluster().startNodes(numNodes);
    ensureGreen();

    logger.info("--> put signed license");
    LicensingClient licensingClient = new LicensingClient(client());
    License license = TestUtils.generateSignedLicense(TimeValue.timeValueMinutes(1));
    putLicense(license);
```

```java
        assertThat(licensingClient.prepareGetLicense().get().license(), equalTo(license));
        assertOperationMode(license.operationMode());

        logger.info("--> restart all nodes");
        internalCluster().fullRestart();
        ensureYellow();
        licensingClient = new LicensingClient(client());
        logger.info("--> get and check signed license");
        assertThat(licensingClient.prepareGetLicense().get().license(), equalTo(license));
        logger.info("--> remove licenses");
        licensingClient.prepareDeleteLicense().get();
        assertOperationMode(License.OperationMode.MISSING);

        logger.info("--> restart all nodes");
        internalCluster().fullRestart();
        licensingClient = new LicensingClient(client());
        ensureYellow();
        assertThat(licensingClient.prepareGetLicense().get().license(), nullValue());
        assertOperationMode(License.OperationMode.MISSING);


        wipeAllLicenses();
    }

    public void testCloudInternalLicense() throws Exception {
        wipeAllLicenses();

        int numNodes = randomIntBetween(1, 5);
        logger.info("--> starting {} node(s)", numNodes);
        internalCluster().startNodes(numNodes);
        ensureGreen();

        logger.info("--> put signed license");
        LicensingClient licensingClient = new LicensingClient(client());
        License license = TestUtils.generateSignedLicense("cloud_internal", License.VERSION_CURRENT,
System.currentTimeMillis(),
            TimeValue.timeValueMinutes(1));
        putLicense(license);
        assertThat(licensingClient.prepareGetLicense().get().license(), equalTo(license));
        assertOperationMode(License.OperationMode.PLATINUM);
        writeCloudInternalMode("gold");
        assertOperationMode(License.OperationMode.GOLD);
        writeCloudInternalMode("basic");
        assertOperationMode(License.OperationMode.BASIC);
    }

    public void testClusterRestartWhileEnabled() throws Exception {
        wipeAllLicenses();
```

```
        internalCluster().startNode();
        ensureGreen();
        assertLicenseActive(true);
        logger.info("--> restart node");
        internalCluster().fullRestart();
        ensureYellow();
        logger.info("--> await node for enabled");
        assertLicenseActive(true);
    }

    public void testClusterRestartWhileGrace() throws Exception {
        wipeAllLicenses();
        internalCluster().startNode();
        assertLicenseActive(true);
        putLicense(TestUtils.generateSignedLicense(TimeValue.timeValueMillis(0)));
        ensureGreen();
        assertLicenseActive(true);
        logger.info("--> restart node");
        internalCluster().fullRestart();
        ensureYellow();
        logger.info("--> await node for grace_period");
        assertLicenseActive(true);
    }

    public void testClusterRestartWhileExpired() throws Exception {
        wipeAllLicenses();
        internalCluster().startNode();
        ensureGreen();
        assertLicenseActive(true);
        putLicense(TestUtils.generateExpiredNonBasicLicense(System.currentTimeMillis() -
LicenseService.GRACE_PERIOD_DURATION.getMillis()));
        assertLicenseActive(false);
        logger.info("--> restart node");
        internalCluster().fullRestart();
        ensureYellow();
        logger.info("--> await node for disabled");
        assertLicenseActive(false);
    }

    public void testClusterRestartWithOldSignature() throws Exception {
        wipeAllLicenses();
        internalCluster().startNode();
        ensureGreen();
        assertLicenseActive(true);
        putLicense(TestUtils.generateSignedLicenseOldSignature());
        LicensingClient licensingClient = new LicensingClient(client());
        assertThat(licensingClient.prepareGetLicense().get().license().version(),
equalTo(License.VERSION_START_DATE));
```

```java
        logger.info("--> restart node");
        internalCluster().fullRestart(); // restart so that license is updated
        ensureYellow();
        logger.info("--> await node for enabled");
        assertLicenseActive(true);
        licensingClient = new LicensingClient(client());
        assertThat(licensingClient.prepareGetLicense().get().license().version(),
    equalTo(License.VERSION_CURRENT)); //license updated
        internalCluster().fullRestart(); // restart once more and verify updated license is active
        ensureYellow();
        logger.info("--> await node for enabled");
        assertLicenseActive(true);
    }

    private void assertOperationMode(License.OperationMode operationMode) throws InterruptedException {
        boolean success = awaitBusy(() -> {
            for (XPackLicenseState licenseState : internalCluster().getDataNodeInstances(XPackLicenseState.class)) {
                if (licenseState.getOperationMode() == operationMode) {
                    return true;
                }
            }
            return false;
        });
        assertTrue(success);
    }

    private void writeCloudInternalMode(String mode) throws Exception {
        for (Environment environment : internalCluster().getDataOrMasterNodeInstances(Environment.class)) {
            Path licenseModePath = XPackPlugin.resolveConfigFile(environment, "license_mode");
            Files.createDirectories(licenseModePath.getParent());
            Files.write(licenseModePath, mode.getBytes(StandardCharsets.UTF_8));
        }
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.threadpool.TestThreadPool;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.junit.After;
import org.junit.Before;
```

```java
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.StandardOpenOption;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.TimeUnit;
import java.util.concurrent.atomic.AtomicReference;

import static org.hamcrest.Matchers.equalTo;

public class OperationModeFileWatcherTests extends ESTestCase {
    private ResourceWatcherService watcherService;
    private TestThreadPool threadPool;
    private Path licenseModePath;
    private OperationModeFileWatcher operationModeFileWatcher;
    private AtomicReference<CountDownLatch> onChangeCounter;

    @Before
    public void setup() throws Exception {
        threadPool = new TestThreadPool("license mode file watcher tests");
        Settings settings = Settings.builder()
                .put("resource.reload.interval.high", "10ms")
                .build();
        watcherService = new ResourceWatcherService(settings,
                threadPool);
        watcherService.start();
        licenseModePath = createTempFile();
        onChangeCounter = new AtomicReference<>(new CountDownLatch(1));
        operationModeFileWatcher = new OperationModeFileWatcher(watcherService, licenseModePath, logger,
                () -> onChangeCounter.get().countDown());
    }

    @After
    public void shutdown() throws InterruptedException {
        terminate(threadPool);
        watcherService.stop();
    }

    public void testInit() throws Exception {
        onChangeCounter.set(new CountDownLatch(2));
        writeMode("gold");
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
        operationModeFileWatcher.init();
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
```

```java
    }

    public void testUpdateModeFromFile() throws Exception {
        Files.delete(licenseModePath);
        operationModeFileWatcher.init();
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
            equalTo(License.OperationMode.PLATINUM));
        writeMode("gold");
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
        onChangeCounter.set(new CountDownLatch(1));
        writeMode("basic");
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.BASIC));
    }

    public void testDeleteModeFromFile() throws Exception {
        Files.delete(licenseModePath);
        operationModeFileWatcher.init();
        writeMode("gold");
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
        onChangeCounter.set(new CountDownLatch(1));

        Files.delete(licenseModePath);
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
            equalTo(License.OperationMode.PLATINUM));
    }

    public void testInvalidModeFromFile() throws Exception {
        writeMode("invalid");
        operationModeFileWatcher.init();
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
            equalTo(License.OperationMode.PLATINUM));
        operationModeFileWatcher.onFileChanged(licenseModePath);
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
            equalTo(License.OperationMode.PLATINUM));
    }

    public void testLicenseModeFileIsDirectory() throws Exception {
        licenseModePath = createTempDir();
        operationModeFileWatcher.init();
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
            equalTo(License.OperationMode.PLATINUM));
        operationModeFileWatcher.onFileChanged(licenseModePath);
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
            equalTo(License.OperationMode.PLATINUM));
```

```
        }

    public void testLicenseModeFileCreatedAfterInit() throws Exception {
        Files.delete(licenseModePath);
        operationModeFileWatcher.init();
        assertThat(operationModeFileWatcher.getCurrentOperationMode(),
equalTo(License.OperationMode.PLATINUM));
        onChangeCounter.set(new CountDownLatch(1));
        Path tempFile = createTempFile();
        writeMode("gold", tempFile);
        licenseModePath = tempFile;
        assertTrue(onChangeCounter.get().await(5, TimeUnit.SECONDS));
        assertThat(operationModeFileWatcher.getCurrentOperationMode(), equalTo(License.OperationMode.GOLD));
    }

    private void writeMode(String mode) throws IOException {
        writeMode(mode, licenseModePath);
    }

    static void writeMode(String mode, Path file) throws IOException {
        Files.write(file, mode.getBytes(StandardCharsets.UTF_8), StandardOpenOption.CREATE);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.io.stream.BytesStreamOutput;
import org.elasticsearch.common.io.stream.StreamInput;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentHelper;
import org.elasticsearch.test.ESTestCase;

import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;

import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.not;
import static org.hamcrest.Matchers.sameInstance;
```

```java
public class PutLicenseResponseTests extends ESTestCase {
  @SuppressWarnings("unchecked")
  public void testSerialization() throws Exception {
    boolean acknowledged = randomBoolean();
    LicensesStatus status = randomFrom(LicensesStatus.VALID, LicensesStatus.INVALID,
LicensesStatus.EXPIRED);
    Map<String, String[]> ackMessages = randomAckMessages();

    PutLicenseResponse response = new PutLicenseResponse(acknowledged, status, "", ackMessages);

    XContentBuilder contentBuilder = XContentFactory.jsonBuilder();
    response.toXContent(contentBuilder, ToXContent.EMPTY_PARAMS);

    Map<String, Object> map = XContentHelper.convertToMap(BytesReference.bytes(contentBuilder), false,
        contentBuilder.contentType()).v2();
    assertThat(map.containsKey("acknowledged"), equalTo(true));
    boolean actualAcknowledged = (boolean) map.get("acknowledged");
    assertThat(actualAcknowledged, equalTo(acknowledged));

    assertThat(map.containsKey("license_status"), equalTo(true));
    String actualStatus = (String) map.get("license_status");
    assertThat(actualStatus, equalTo(status.name().toLowerCase(Locale.ROOT)));

    assertThat(map.containsKey("acknowledge"), equalTo(true));
    Map<String, List<String>> actualAckMessages = (Map<String, List<String>>) map.get("acknowledge");
    assertTrue(actualAckMessages.containsKey("message"));
    actualAckMessages.remove("message");
    assertThat(actualAckMessages.keySet(), equalTo(ackMessages.keySet()));
    for (Map.Entry<String, List<String>> entry : actualAckMessages.entrySet()) {
      assertArrayEquals(entry.getValue().toArray(), ackMessages.get(entry.getKey()));
    }
  }

  public void testStreamSerialization() throws IOException {
    boolean acknowledged = randomBoolean();
    LicensesStatus status = randomFrom(LicensesStatus.VALID, LicensesStatus.INVALID,
LicensesStatus.EXPIRED);
    Map<String, String[]> ackMessages = randomAckMessages();

    // write the steam so that we can attempt to read it back
    BytesStreamOutput output = new BytesStreamOutput();

    PutLicenseResponse response = new PutLicenseResponse(acknowledged, status, "", ackMessages);
    // write it out
    response.writeTo(output);

    StreamInput input = output.bytes().streamInput();
```

```java
        // read it back in
        response.readFrom(input);

        assertThat(response.isAcknowledged(), equalTo(acknowledged));
        assertThat(response.status(), equalTo(status));
        assertThat(response.acknowledgeMessages(), not(sameInstance(ackMessages)));
        assertThat(response.acknowledgeMessages().size(), equalTo(ackMessages.size()));

        for (String key : ackMessages.keySet()) {
            assertArrayEquals(ackMessages.get(key), response.acknowledgeMessages().get(key));
        }
    }

    private static Map<String, String[]> randomAckMessages() {
        int nFeatures = randomIntBetween(1, 5);

        Map<String, String[]> ackMessages = new HashMap<>();

        for (int i = 0; i < nFeatures; i++) {
            String feature = randomAlphaOfLengthBetween(9, 15);
            int nMessages = randomIntBetween(1, 5);
            String[] messages = new String[nMessages];
            for (int j = 0; j < nMessages; j++) {
                messages[j] = randomAlphaOfLengthBetween(10, 30);
            }
            ackMessages.put(feature, messages);
        }

        return ackMessages;
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.analysis.common.CommonAnalysisPlugin;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;

import java.util.Arrays;
import java.util.Collection;
```

```java
import static org.elasticsearch.test.ESIntegTestCase.Scope.TEST;

@ESIntegTestCase.ClusterScope(scope = TEST, numDataNodes = 0, numClientNodes = 0, maxNumDataNodes =
0, transportClientRatio = 0,
    autoMinMasterNodes = false)
public class LicenseServiceClusterNotRecoveredTests extends AbstractLicensesIntegrationTestCase {

    @Override
    protected Settings nodeSettings(int nodeOrdinal) {
        return nodeSettingsBuilder(nodeOrdinal).build();
    }

    @Override
    protected boolean addMockHttpTransport() {
        return false;
    }

    private Settings.Builder nodeSettingsBuilder(int nodeOrdinal) {
        return Settings.builder()
            .put(super.nodeSettings(nodeOrdinal))
            .put("node.data", true)
            .put("resource.reload.interval.high", "500ms"); // for license mode file watcher
    }

    @Override
    protected Collection<Class<? extends Plugin>> nodePlugins() {
        return Arrays.asList(LocalStateCompositeXPackPlugin.class, CommonAnalysisPlugin.class,
Netty4Plugin.class);
    }

    @Override
    protected Collection<Class<? extends Plugin>> transportClientPlugins() {
        return nodePlugins();
    }

    public void testClusterNotRecovered() throws Exception {
        logger.info("--> start one master out of two [recovery state]");
        internalCluster().startNode(nodeSettingsBuilder(0).put("discovery.zen.minimum_master_nodes",
2).put("node.master", true));
        logger.info("--> start second master out of two [recovered state]");
        internalCluster().startNode(nodeSettingsBuilder(1).put("discovery.zen.minimum_master_nodes",
2).put("node.master", true));
        assertLicenseActive(true);
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
```

```java
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.ack.ClusterStateUpdateResponse;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESSingleNodeTestCase;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;
import org.junit.Before;

import java.util.Collection;
import java.util.Collections;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.atomic.AtomicBoolean;

import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.not;

public class LicensesManagerServiceTests extends ESSingleNodeTestCase {

    @Override
    protected Collection<Class<? extends Plugin>> getPlugins() {
        return Collections.singletonList(LocalStateCompositeXPackPlugin.class);
    }

    @Override
    protected Settings nodeSettings() {
        return Settings.builder()
            .put(super.nodeSettings())
            .put(XPackSettings.SECURITY_ENABLED.getKey(), false)
            .put(XPackSettings.MONITORING_ENABLED.getKey(), false)
            .put(XPackSettings.WATCHER_ENABLED.getKey(), false)
            .put(XPackSettings.GRAPH_ENABLED.getKey(), false)
            .put(XPackSettings.MACHINE_LEARNING_ENABLED.getKey(), false)
            .build();
    }

    @Override
    protected boolean resetNodeAfterTest() {
        return true;
    }
```

```java
    @Before
    public void waitForTrialLicenseToBeGenerated() throws Exception {
        assertBusy(() ->
assertNotNull(getInstanceFromNode(ClusterService.class).state().metaData().custom(LicensesMetaData.TYPE)));
    }


    public void testStoreAndGetLicenses() throws Exception {
        LicenseService licenseService = getInstanceFromNode(LicenseService.class);
        ClusterService clusterService = getInstanceFromNode(ClusterService.class);
        License goldLicense = TestUtils.generateSignedLicense("gold", TimeValue.timeValueHours(1));
        TestUtils.registerAndAckSignedLicenses(licenseService, goldLicense, LicensesStatus.VALID);
        License silverLicense = TestUtils.generateSignedLicense("silver", TimeValue.timeValueHours(2));
        TestUtils.registerAndAckSignedLicenses(licenseService, silverLicense, LicensesStatus.VALID);
        License platinumLicense = TestUtils.generateSignedLicense("platinum", TimeValue.timeValueHours(1));
        TestUtils.registerAndAckSignedLicenses(licenseService, platinumLicense, LicensesStatus.VALID);
        LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        assertThat(licensesMetaData.getLicense(), equalTo(platinumLicense));
        final License getLicenses = licenseService.getLicense();
        assertThat(getLicenses, equalTo(platinumLicense));
    }

    // TODO: Add test/feature blocking the registration of basic license

    public void testEffectiveLicenses() throws Exception {
        final LicenseService licenseService = getInstanceFromNode(LicenseService.class);
        final ClusterService clusterService = getInstanceFromNode(ClusterService.class);
        License goldLicense = TestUtils.generateSignedLicense("gold", TimeValue.timeValueSeconds(5));
        // put gold license
        TestUtils.registerAndAckSignedLicenses(licenseService, goldLicense, LicensesStatus.VALID);
        LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        assertThat(LicenseService.getLicense(licensesMetaData), equalTo(goldLicense));

        License platinumLicense = TestUtils.generateSignedLicense("platinum", TimeValue.timeValueSeconds(3));
        // put platinum license
        TestUtils.registerAndAckSignedLicenses(licenseService, platinumLicense, LicensesStatus.VALID);
        licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        assertThat(LicenseService.getLicense(licensesMetaData), equalTo(platinumLicense));
    }

    public void testInvalidLicenseStorage() throws Exception {
        LicenseService licenseService = getInstanceFromNode(LicenseService.class);
        ClusterService clusterService = getInstanceFromNode(ClusterService.class);
        License signedLicense = TestUtils.generateSignedLicense(TimeValue.timeValueMinutes(2));

        // modify content of signed license
        License tamperedLicense = License.builder()
            .fromLicenseSpec(signedLicense, signedLicense.signature())
            .expiryDate(signedLicense.expiryDate() + 10 * 24 * 60 * 60 * 1000L)
```

```
            .validate()
            .build();

        TestUtils.registerAndAckSignedLicenses(licenseService, tamperedLicense, LicensesStatus.INVALID);

        // ensure that the invalid license never made it to cluster state
        LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        assertThat(licensesMetaData.getLicense(), not(equalTo(tamperedLicense)));
    }

    public void testRemoveLicenses() throws Exception {
        LicenseService licenseService = getInstanceFromNode(LicenseService.class);
        ClusterService clusterService = getInstanceFromNode(ClusterService.class);

        // generate signed licenses
        License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(1));
        TestUtils.registerAndAckSignedLicenses(licenseService, license, LicensesStatus.VALID);
        LicensesMetaData licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        assertThat(licensesMetaData.getLicense(), not(LicensesMetaData.LICENSE_TOMBSTONE));

        // remove signed licenses
        removeAndAckSignedLicenses(licenseService);
        licensesMetaData = clusterService.state().metaData().custom(LicensesMetaData.TYPE);
        assertThat(licensesMetaData.getLicense(), equalTo(LicensesMetaData.LICENSE_TOMBSTONE));
    }

    private void removeAndAckSignedLicenses(final LicenseService licenseService) {
        final CountDownLatch latch = new CountDownLatch(1);
        final AtomicBoolean success = new AtomicBoolean(false);
        licenseService.removeLicense(new DeleteLicenseRequest(), new
ActionListener<ClusterStateUpdateResponse>() {
            @Override
            public void onResponse(ClusterStateUpdateResponse clusterStateUpdateResponse) {
                if (clusterStateUpdateResponse.isAcknowledged()) {
                    success.set(true);
                }
                latch.countDown();
            }

            @Override
            public void onFailure(Exception throwable) {
                latch.countDown();
            }
        });
        try {
            latch.await();
        } catch (InterruptedException e) {
            fail(e.getMessage());
```

```java
        }
        assertThat("remove license(s) failed", success.get(), equalTo(true));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentHelper;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.test.ESTestCase;

import java.nio.charset.StandardCharsets;
import java.util.Collections;
import java.util.Map;
import java.util.UUID;

import static org.hamcrest.core.IsEqual.equalTo;
import static org.hamcrest.core.IsNull.notNullValue;
import static org.hamcrest.core.IsNull.nullValue;

public class LicenseSerializationTests extends ESTestCase {
    public void testSimpleIssueExpiryDate() throws Exception {
        long now = System.currentTimeMillis();
        String issueDate = TestUtils.dateMathString("now", now);
        String expiryDate = TestUtils.dateMathString("now+10d/d", now);
        String licenseSpecs = TestUtils.generateLicenseSpecString(new TestUtils.LicenseSpec(issueDate, expiryDate));
        License generatedLicense = License.fromSource(new
BytesArray(licenseSpecs.getBytes(StandardCharsets.UTF_8)), XContentType.JSON);
        assertThat(generatedLicense.issueDate(), equalTo(DateUtils.beginningOfTheDay(issueDate)));
        assertThat(generatedLicense.expiryDate(), equalTo(DateUtils.endOfTheDay(expiryDate)));
    }

    public void testLicensesFields() throws Exception {
        TestUtils.LicenseSpec randomLicenseSpec =
TestUtils.generateRandomLicenseSpec(License.VERSION_START);
        String licenseSpecsSource = TestUtils.generateLicenseSpecString(randomLicenseSpec);
        final License fromSource =
            License.fromSource(new BytesArray(licenseSpecsSource.getBytes(StandardCharsets.UTF_8)),
XContentType.JSON);
```

```java
        TestUtils.assertLicenseSpec(randomLicenseSpec, fromSource);
    }

  public void testLicenseRestView() throws Exception {
      long now = System.currentTimeMillis();
      String expiredLicenseExpiryDate = TestUtils.dateMathString("now-1d/d", now);
      String validLicenseIssueDate = TestUtils.dateMathString("now-10d/d", now);
      String invalidLicenseIssueDate = TestUtils.dateMathString("now+1d/d", now);
      String validLicenseExpiryDate = TestUtils.dateMathString("now+2d/d", now);

      License license = TestUtils.generateLicenses(new TestUtils.LicenseSpec(validLicenseIssueDate,
expiredLicenseExpiryDate));
      XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
      license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
      builder.flush();
      Map<String, Object> map = XContentHelper.convertToMap(BytesReference.bytes(builder), false,
builder.contentType()).v2();

      // should have an extra status field, human readable issue_data and expiry_date
      assertThat(map.get("status"), notNullValue());
      assertThat(map.get("issue_date"), notNullValue());
      assertThat(map.get("expiry_date"), notNullValue());
      assertThat(map.get("status"), equalTo("expired"));
      builder = XContentFactory.contentBuilder(XContentType.JSON);
      license.toXContent(builder, ToXContent.EMPTY_PARAMS);
      builder.flush();
      map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
      assertThat(map.get("status"), nullValue());

      license = TestUtils.generateLicenses(new TestUtils.LicenseSpec(validLicenseIssueDate,
validLicenseExpiryDate));
      builder = XContentFactory.contentBuilder(XContentType.JSON);
      license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
      builder.flush();
      map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();

      // should have an extra status field, human readable issue_data and expiry_date
      assertThat(map.get("status"), notNullValue());
      assertThat(map.get("issue_date"), notNullValue());
      assertThat(map.get("expiry_date"), notNullValue());
      assertThat(map.get("status"), equalTo("active"));
      builder = XContentFactory.contentBuilder(XContentType.JSON);
      license.toXContent(builder, ToXContent.EMPTY_PARAMS);
      builder.flush();
      map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
      assertThat(map.get("status"), nullValue());
```

```java
        license = TestUtils.generateLicenses(new TestUtils.LicenseSpec(invalidLicenseIssueDate,
validLicenseExpiryDate));
        builder = XContentFactory.contentBuilder(XContentType.JSON);
        license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
        builder.flush();
        map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();

        // should have an extra status field, human readable issue_data and expiry_date
        assertThat(map.get("status"), notNullValue());
        assertThat(map.get("issue_date"), notNullValue());
        assertThat(map.get("expiry_date"), notNullValue());
        assertThat(map.get("status"), equalTo("invalid"));
        builder = XContentFactory.contentBuilder(XContentType.JSON);
        license.toXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.flush();
        map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
        assertThat(map.get("status"), nullValue());
    }

    public void testLicenseRestViewNonExpiringBasic() throws Exception {
        long now = System.currentTimeMillis();

        License.Builder specBuilder = License.builder()
            .uid(UUID.randomUUID().toString())
            .issuedTo("test")
            .maxNodes(1000)
            .issueDate(now)
            .type("basic")
            .expiryDate(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS);
        License license = SelfGeneratedLicense.create(specBuilder, License.VERSION_CURRENT);
        XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
        license.toXContent(builder, new
ToXContent.MapParams(Collections.singletonMap(License.REST_VIEW_MODE, "true")));
        builder.flush();
        Map<String, Object> map = XContentHelper.convertToMap(BytesReference.bytes(builder), false,
builder.contentType()).v2();

        // should have an extra status field, human readable issue_data and no expiry_date
        assertThat(map.get("status"), notNullValue());
        assertThat(map.get("type"), equalTo("basic"));
        assertThat(map.get("issue_date"), notNullValue());
        assertThat(map.get("expiry_date"), nullValue());
        assertThat(map.get("expiry_date_in_millis"), nullValue());
        assertThat(map.get("status"), equalTo("active"));
        builder = XContentFactory.contentBuilder(XContentType.JSON);
        license.toXContent(builder, ToXContent.EMPTY_PARAMS);
```

```java
        builder.flush();
        map = XContentHelper.convertToMap(BytesReference.bytes(builder), false, builder.contentType()).v2();
        assertThat(map.get("status"), nullValue());
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.watcher.FileWatcher;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.junit.Before;

import java.nio.file.Path;

import static org.hamcrest.Matchers.equalTo;
import static org.mockito.Matchers.any;
import static org.mockito.Matchers.eq;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;
import static org.mockito.Mockito.verifyZeroInteractions;

public class LicenseOperationModeUpdateTests extends ESTestCase {

    private OperationModeFileWatcher operationModeFileWatcher;
    private Path licenseModeFile;
    private ResourceWatcherService resourceWatcherService;

    @Before
    public void init() throws Exception {
        licenseModeFile = createTempFile();
        resourceWatcherService = mock(ResourceWatcherService.class);
        operationModeFileWatcher = new OperationModeFileWatcher(resourceWatcherService, licenseModeFile,
logger, () -> {});
    }

    public void testLicenseOperationModeUpdate() throws Exception {
        String type = randomFrom("trial", "basic", "standard", "gold", "platinum");
        License license = License.builder()
                .uid("id")
                .expiryDate(0)
                .issueDate(0)
                .issuedTo("elasticsearch")
```

```
                .issuer("issuer")
                .type(type)
                .maxNodes(1)
                .build();

        assertThat(license.operationMode(), equalTo(License.OperationMode.resolve(type)));
        OperationModeFileWatcherTests.writeMode("gold", licenseModeFile);
        license.setOperationModeFileWatcher(operationModeFileWatcher);
        verifyZeroInteractions(resourceWatcherService);
        assertThat(license.operationMode(), equalTo(License.OperationMode.resolve(type)));
    }

    public void testCloudInternalLicenseOperationModeUpdate() throws Exception {
        License license = License.builder()
                .uid("id")
                .expiryDate(0)
                .issueDate(0)
                .issuedTo("elasticsearch")
                .issuer("issuer")
                .type("cloud_internal")
                .maxNodes(1)
                .build();

        assertThat(license.operationMode(), equalTo(License.OperationMode.PLATINUM));
        OperationModeFileWatcherTests.writeMode("gold", licenseModeFile);
        license.setOperationModeFileWatcher(operationModeFileWatcher);
        verify(resourceWatcherService, times(1)).add(any(FileWatcher.class),
eq(ResourceWatcherService.Frequency.HIGH));
        assertThat(license.operationMode(), equalTo(License.OperationMode.GOLD));
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.analysis.common.CommonAnalysisPlugin;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.Nullable;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
```

```java
import org.elasticsearch.xpack.core.XPackClientPlugin;
import org.elasticsearch.xpack.core.XPackSettings;

import java.util.Arrays;
import java.util.Collection;
import java.util.concurrent.CountDownLatch;

public abstract class AbstractLicensesIntegrationTestCase extends ESIntegTestCase {

    @Override
    protected Settings nodeSettings(int nodeOrdinal) {
        return Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), false).build();
    }

    @Override
    protected Collection<Class<? extends Plugin>> nodePlugins() {
        return Arrays.asList(LocalStateCompositeXPackPlugin.class, CommonAnalysisPlugin.class);
    }

    @Override
    protected Collection<Class<? extends Plugin>> transportClientPlugins() {
        return Arrays.asList(XPackClientPlugin.class, CommonAnalysisPlugin.class);
    }

    @Override
    protected Settings transportClientSettings() {
        // Plugin should be loaded on the transport client as well
        return nodeSettings(0);
    }

    protected void putLicense(final License license) throws InterruptedException {
        final CountDownLatch latch = new CountDownLatch(1);
        ClusterService clusterService = internalCluster().getInstance(ClusterService.class,
internalCluster().getMasterName());
        clusterService.submitStateUpdateTask("putting license", new ClusterStateUpdateTask() {
            @Override
            public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
                latch.countDown();
            }

            @Override
            public ClusterState execute(ClusterState currentState) throws Exception {
                MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
                mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
                return ClusterState.builder(currentState).metaData(mdBuilder).build();
            }

            @Override
```

```java
        public void onFailure(String source, @Nullable Exception e) {
            logger.error("error on metaData cleanup after test", e);
        }
    });
    latch.await();
}


protected void putLicenseTombstone() throws InterruptedException {
    putLicense(LicensesMetaData.LICENSE_TOMBSTONE);
}


protected void wipeAllLicenses() throws InterruptedException {
    final CountDownLatch latch = new CountDownLatch(1);
    ClusterService clusterService = internalCluster().getInstance(ClusterService.class,
internalCluster().getMasterName());
    clusterService.submitStateUpdateTask("delete licensing metadata", new ClusterStateUpdateTask() {
        @Override
        public void clusterStateProcessed(String source, ClusterState oldState, ClusterState newState) {
            latch.countDown();
        }

        @Override
        public ClusterState execute(ClusterState currentState) throws Exception {
            MetaData.Builder mdBuilder = MetaData.builder(currentState.metaData());
            mdBuilder.removeCustom(LicensesMetaData.TYPE);
            return ClusterState.builder(currentState).metaData(mdBuilder).build();
        }

        @Override
        public void onFailure(String source, @Nullable Exception e) {
            logger.error("error on metaData cleanup after test", e);
        }
    });
    latch.await();
}


protected void assertLicenseActive(boolean active) throws InterruptedException {
    boolean success = awaitBusy(() -> {
        for (XPackLicenseState licenseState : internalCluster().getDataNodeInstances(XPackLicenseState.class)) {
            if (licenseState.isActive() == active) {
                return true;
            }
        }
        return false;
    });
    assertTrue(success);
}
```

```java
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.XPackSettings;

import java.util.Arrays;
import java.util.function.Predicate;
import java.util.stream.Collectors;

import static org.elasticsearch.license.License.OperationMode.BASIC;
import static org.elasticsearch.license.License.OperationMode.GOLD;
import static org.elasticsearch.license.License.OperationMode.MISSING;
import static org.elasticsearch.license.License.OperationMode.PLATINUM;
import static org.elasticsearch.license.License.OperationMode.STANDARD;
import static org.elasticsearch.license.License.OperationMode.TRIAL;
import static org.hamcrest.Matchers.is;

/**
 * Unit tests for the {@link XPackLicenseState}
 */
public class XPackLicenseStateTests extends ESTestCase {

    /** Creates a license state with the given license type and active state, and checks the given method returns
     expected. */
    void assertAllowed(OperationMode mode, boolean active, Predicate<XPackLicenseState> predicate, boolean
    expected) {
        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        licenseState.update(mode, active);
        assertEquals(expected, predicate.test(licenseState));
    }

    /**
     * Checks the ack message going from the  {@code from} license type to {@code to} license type.
     * TODO: check the actual messages, not just the number of them! This was copied from previous license tests...
     */
    void assertAckMesssages(String feature, OperationMode from, OperationMode to, int expectedMessages) {
        String[] gotMessages = XPackLicenseState.ACKNOWLEDGMENT_MESSAGES.get(feature).apply(from, to);
        assertEquals(expectedMessages, gotMessages.length);
    }
```

```java
static <T> T randomFrom(T[] values, Predicate<T> filter) {
    return randomFrom(Arrays.stream(values).filter(filter).collect(Collectors.toList()));
}

static OperationMode randomMode() {
    return randomFrom(OperationMode.values());
}

public static OperationMode randomTrialStandardGoldOrPlatinumMode() {
    return randomFrom(TRIAL, STANDARD, GOLD, PLATINUM);
}

public static OperationMode randomTrialOrPlatinumMode() {
    return randomFrom(TRIAL, PLATINUM);
}

public static OperationMode randomTrialBasicStandardGoldOrPlatinumMode() {
    return randomFrom(TRIAL, BASIC, STANDARD, GOLD, PLATINUM);
}

public static OperationMode randomBasicStandardOrGold() {
    return randomFrom(BASIC, STANDARD, GOLD);
}

public void testSecurityDefaults() {
    XPackLicenseState licenseState =
            new XPackLicenseState(Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(),
true).build());
    assertThat(licenseState.isAuthAllowed(), is(true));
    assertThat(licenseState.isIpFilteringAllowed(), is(true));
    assertThat(licenseState.isAuditingAllowed(), is(true));
    assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
    assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
    assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
    assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));

    licenseState =
        new XPackLicenseState(Settings.builder().put(XPackSettings.TRANSPORT_SSL_ENABLED.getKey(),
true).build());
    assertThat(licenseState.isAuthAllowed(), is(true));
    assertThat(licenseState.isIpFilteringAllowed(), is(true));
    assertThat(licenseState.isAuditingAllowed(), is(true));
    assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
    assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
    assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
    assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));
```

```
        licenseState = new XPackLicenseState(Settings.EMPTY);
        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));
    }

    public void testSecurityBasic() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
            Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(BASIC, true);

        assertThat(licenseState.isAuthAllowed(), is(false));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NONE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityBasicExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
            Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(BASIC, false);

        assertThat(licenseState.isAuthAllowed(), is(false));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NONE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityStandard() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
            Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(STANDARD, true);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
```

```java
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NATIVE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityStandardExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
            Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(STANDARD, false);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(false));
        assertThat(licenseState.isAuditingAllowed(), is(false));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.NATIVE));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityGold() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
            Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(GOLD, true);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.DEFAULT));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityGoldExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
            Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(GOLD, false);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(false));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.DEFAULT));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityPlatinum() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
```

```java
            Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(PLATINUM, true);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(true));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(true));
    }

    public void testSecurityPlatinumExpired() {
        XPackLicenseState licenseState = new XPackLicenseState(randomFrom(Settings.EMPTY,
                Settings.builder().put(XPackSettings.SECURITY_ENABLED.getKey(), true).build()));
        licenseState.update(PLATINUM, false);

        assertThat(licenseState.isAuthAllowed(), is(true));
        assertThat(licenseState.isIpFilteringAllowed(), is(true));
        assertThat(licenseState.isAuditingAllowed(), is(true));
        assertThat(licenseState.isStatsAndHealthAllowed(), is(false));
        assertThat(licenseState.isDocumentAndFieldLevelSecurityAllowed(), is(true));
        assertThat(licenseState.allowedRealmType(), is(XPackLicenseState.AllowedRealmType.ALL));
        assertThat(licenseState.isCustomRoleProvidersAllowed(), is(false));
    }

    public void testSecurityAckBasicToNotGoldOrStandard() {
        OperationMode toMode = randomFrom(OperationMode.values(), mode -> mode != GOLD && mode !=
STANDARD);
        assertAckMesssages(XPackField.SECURITY, BASIC, toMode, 0);
    }

    public void testSecurityAckAnyToTrialOrPlatinum() {
        assertAckMesssages(XPackField.SECURITY, randomMode(), randomTrialOrPlatinumMode(), 0);
    }

    public void testSecurityAckTrialStandardGoldOrPlatinumToBasic() {
        assertAckMesssages(XPackField.SECURITY, randomTrialStandardGoldOrPlatinumMode(), BASIC, 3);
    }

    public void testSecurityAckAnyToStandard() {
        OperationMode from = randomFrom(BASIC, GOLD, PLATINUM, TRIAL);
        assertAckMesssages(XPackField.SECURITY, from, STANDARD, 4);
    }

    public void testSecurityAckBasicStandardTrialOrPlatinumToGold() {
        OperationMode from = randomFrom(BASIC, PLATINUM, TRIAL, STANDARD);
        assertAckMesssages(XPackField.SECURITY, from, GOLD, 2);
```

```
    }

    public void testMonitoringAckBasicToAny() {
        assertAckMesssages(XPackField.MONITORING, BASIC, randomMode(), 0);
    }

    public void testMonitoringAckAnyToTrialGoldOrPlatinum() {
        assertAckMesssages(XPackField.MONITORING, randomMode(),
randomTrialStandardGoldOrPlatinumMode(), 0);
    }

    public void testMonitoringAckNotBasicToBasic() {
        OperationMode from = randomFrom(STANDARD, GOLD, PLATINUM, TRIAL);
        assertAckMesssages(XPackField.MONITORING, from, BASIC, 2);
    }

    public void testMonitoringAllowed() {
        assertAllowed(randomMode(), true, XPackLicenseState::isMonitoringAllowed, true);
        assertAllowed(randomMode(), false, XPackLicenseState::isMonitoringAllowed, false);
    }

    public void testMonitoringUpdateRetention() {
        assertAllowed(STANDARD, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(GOLD, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(PLATINUM, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(TRIAL, true, XPackLicenseState::isUpdateRetentionAllowed, true);
        assertAllowed(BASIC, true, XPackLicenseState::isUpdateRetentionAllowed, false);
        assertAllowed(MISSING, false, XPackLicenseState::isUpdateRetentionAllowed, false);
    }

    public void testWatcherPlatinumGoldTrialStandard() throws Exception {
        assertAllowed(TRIAL, true, XPackLicenseState::isWatcherAllowed, true);
        assertAllowed(GOLD, true, XPackLicenseState::isWatcherAllowed, true);
        assertAllowed(PLATINUM, true, XPackLicenseState::isWatcherAllowed, true);
        assertAllowed(STANDARD, true, XPackLicenseState::isWatcherAllowed, true);
    }

    public void testWatcherBasicLicense() throws Exception {
        assertAllowed(BASIC, true, XPackLicenseState::isWatcherAllowed, false);
    }

    public void testWatcherInactive() {
        assertAllowed(BASIC, false, XPackLicenseState::isWatcherAllowed, false);
    }

    public void testWatcherInactivePlatinumGoldTrial() throws Exception {
        assertAllowed(TRIAL, false, XPackLicenseState::isWatcherAllowed, false);
        assertAllowed(GOLD, false, XPackLicenseState::isWatcherAllowed, false);
```

```java
        assertAllowed(PLATINUM, false, XPackLicenseState::isWatcherAllowed, false);
        assertAllowed(STANDARD, false, XPackLicenseState::isWatcherAllowed, false);
    }

    public void testGraphPlatinumTrial() throws Exception {
        assertAllowed(TRIAL, true, XPackLicenseState::isGraphAllowed, true);
        assertAllowed(PLATINUM, true, XPackLicenseState::isGraphAllowed, true);
    }

    public void testGraphBasic() throws Exception {
        assertAllowed(BASIC, true, XPackLicenseState::isGraphAllowed, false);
    }

    public void testGraphStandard() throws Exception {
        assertAllowed(STANDARD, true, XPackLicenseState::isGraphAllowed, false);
    }

    public void testGraphInactiveBasic() {
        assertAllowed(BASIC, false, XPackLicenseState::isGraphAllowed, false);
    }

    public void testGraphInactivePlatinumTrial() throws Exception {
        assertAllowed(TRIAL, false, XPackLicenseState::isMachineLearningAllowed, false);
        assertAllowed(PLATINUM, false, XPackLicenseState::isMachineLearningAllowed, false);
    }

    public void testMachineLearningPlatinumTrial() throws Exception {
        assertAllowed(TRIAL, true, XPackLicenseState::isMachineLearningAllowed, true);
        assertAllowed(PLATINUM, true, XPackLicenseState::isMachineLearningAllowed, true);
    }

    public void testMachineLearningBasic() throws Exception {
        assertAllowed(BASIC, true, XPackLicenseState::isMachineLearningAllowed, false);
    }

    public void testMachineLearningStandard() throws Exception {
        assertAllowed(STANDARD, true, XPackLicenseState::isMachineLearningAllowed, false);
    }

    public void testMachineLearningInactiveBasic() {
        assertAllowed(BASIC, false, XPackLicenseState::isMachineLearningAllowed, false);
    }

    public void testMachineLearningInactivePlatinumTrial() throws Exception {
        assertAllowed(TRIAL, false, XPackLicenseState::isMachineLearningAllowed, false);
        assertAllowed(PLATINUM, false, XPackLicenseState::isMachineLearningAllowed, false);
    }
```

```java
public void testLogstashPlatinumGoldTrialStandard() throws Exception {
    assertAllowed(TRIAL, true, XPackLicenseState::isLogstashAllowed, true);
    assertAllowed(GOLD, true, XPackLicenseState::isLogstashAllowed, true);
    assertAllowed(PLATINUM, true, XPackLicenseState::isLogstashAllowed, true);
    assertAllowed(STANDARD, true, XPackLicenseState::isLogstashAllowed, true);
}

public void testLogstashBasicLicense() throws Exception {
    assertAllowed(BASIC, true, XPackLicenseState::isLogstashAllowed, false);
}

public void testLogstashInactive() {
    assertAllowed(BASIC, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(TRIAL, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(GOLD, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(PLATINUM, false, XPackLicenseState::isLogstashAllowed, false);
    assertAllowed(STANDARD, false, XPackLicenseState::isLogstashAllowed, false);
}

public void testSqlDefaults() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(true));
}

public void testSqlBasic() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(BASIC, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlBasicExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(BASIC, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlStandard() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(STANDARD, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}
```

```java
public void testSqlStandardExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(STANDARD, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlGold() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(GOLD, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlGoldExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(GOLD, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlPlatinum() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(PLATINUM, true);

    assertThat(licenseState.isSqlAllowed(), is(true));
    assertThat(licenseState.isJdbcAllowed(), is(true));
}

public void testSqlPlatinumExpired() {
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    licenseState.update(PLATINUM, false);

    assertThat(licenseState.isSqlAllowed(), is(false));
    assertThat(licenseState.isJdbcAllowed(), is(false));
}

public void testSqlAckAnyToTrialOrPlatinum() {
    assertAckMesssages(XPackField.SQL, randomMode(), randomTrialOrPlatinumMode(), 0);
}

public void testSqlAckTrialOrPlatinumToNotTrialOrPlatinum() {
    assertAckMesssages(XPackField.SQL, randomTrialOrPlatinumMode(), randomBasicStandardOrGold(), 1);
}
```

```java
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.scheduler.SchedulerEngine;
import org.junit.Before;

import static org.hamcrest.Matchers.equalTo;

public class LicenseScheduleTests extends ESTestCase {

    private License license;
    private SchedulerEngine.Schedule schedule;

    @Before
    public void setuo() throws Exception {
        license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(12));
        schedule = LicenseService.nextLicenseCheck(license);
    }

    public void testEnabledLicenseSchedule() throws Exception {
        int expiryDuration = (int) (license.expiryDate() - license.issueDate());
        long triggeredTime = license.issueDate() + between(0, expiryDuration);
        assertThat(schedule.nextScheduledTimeAfter(license.issueDate(), triggeredTime),
equalTo(license.expiryDate()));
    }

    public void testGraceLicenseSchedule() throws Exception {
        long triggeredTime = license.expiryDate() + between(1,
            ((int) LicenseService.GRACE_PERIOD_DURATION.getMillis()));
        assertThat(schedule.nextScheduledTimeAfter(license.issueDate(), triggeredTime),
            equalTo(license.expiryDate() + LicenseService.GRACE_PERIOD_DURATION.getMillis()));
    }

    public void testExpiredLicenseSchedule() throws Exception {
        long triggeredTime = license.expiryDate() + LicenseService.GRACE_PERIOD_DURATION.getMillis() +
            randomIntBetween(1, 1000);
        assertThat(schedule.nextScheduledTimeAfter(license.issueDate(), triggeredTime),
            equalTo(-1L));
    }
```

```java
    public void testInvalidLicenseSchedule() throws Exception {
        long triggeredTime = license.issueDate() - randomIntBetween(1, 1000);
        assertThat(schedule.nextScheduledTimeAfter(triggeredTime, triggeredTime),
            equalTo(license.issueDate()));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.client.Response;
import org.elasticsearch.client.ResponseException;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.common.io.Streams;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackClientPlugin;

import java.io.InputStreamReader;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.test.ESIntegTestCase.Scope.SUITE;

@ESIntegTestCase.ClusterScope(scope = SUITE)
public class StartBasicLicenseTests extends AbstractLicensesIntegrationTestCase {

    @Override
    protected boolean addMockHttpTransport() {
        return false; // enable http
    }

    @Override
    protected Settings nodeSettings(int nodeOrdinal) {
        return Settings.builder()
            .put(super.nodeSettings(nodeOrdinal))
            .put("node.data", true)
            .put(LicenseService.SELF_GENERATED_LICENSE_TYPE.getKey(), "basic").build();
    }
```

```java
    @Override
    protected Collection<Class<? extends Plugin>> nodePlugins() {
        return Arrays.asList(LocalStateCompositeXPackPlugin.class, Netty4Plugin.class);
    }

    @Override
    protected Collection<Class<? extends Plugin>> transportClientPlugins() {
        return Arrays.asList(XPackClientPlugin.class, Netty4Plugin.class);
    }

    public void testStartBasicLicense() throws Exception {
        LicensingClient licensingClient = new LicensingClient(client());
        License license = TestUtils.generateSignedLicense("trial",  License.VERSION_CURRENT, -1,
TimeValue.timeValueHours(24));
        licensingClient.preparePutLicense(license).get();

        assertBusy(() -> {
            GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();
            assertEquals("trial", getLicenseResponse.license().type());
        });

        // Testing that you can start a basic license when you have no license
        if (randomBoolean()) {
            licensingClient.prepareDeleteLicense().get();
            assertNull(licensingClient.prepareGetLicense().get().license());
        }

        RestClient restClient = getRestClient();
        Response response = restClient.performRequest("GET", "/_xpack/license/basic_status");
        String body = Streams.copyToString(new InputStreamReader(response.getEntity().getContent(),
StandardCharsets.UTF_8));
        assertEquals(200, response.getStatusLine().getStatusCode());
        assertEquals("{\"eligible_to_start_basic\":true}", body);

        Response response2 = restClient.performRequest("POST", "/_xpack/license/start_basic?acknowledge=true");
        String body2 = Streams.copyToString(new InputStreamReader(response2.getEntity().getContent(),
StandardCharsets.UTF_8));
        assertEquals(200, response2.getStatusLine().getStatusCode());
        assertTrue(body2.contains("\"acknowledged\":true"));
        assertTrue(body2.contains("\"basic_was_started\":true"));

        assertBusy(() -> {
            GetLicenseResponse currentLicense = licensingClient.prepareGetLicense().get();
            assertEquals("basic", currentLicense.license().type());
        });

        long expirationMillis = licensingClient.prepareGetLicense().get().license().expiryDate();
        assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS,
```

```
expirationMillis);

    Response response3 = restClient.performRequest("GET", "/_xpack/license");
    String body3 = Streams.copyToString(new InputStreamReader(response3.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertTrue(body3.contains("\"type\" : \"basic\""));
    assertFalse(body3.contains("expiry_date"));
    assertFalse(body3.contains("expiry_date_in_millis"));


    Response response4 = restClient.performRequest("GET", "/_xpack/license/basic_status");
    String body4 = Streams.copyToString(new InputStreamReader(response4.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response3.getStatusLine().getStatusCode());
    assertEquals("{\"eligible_to_start_basic\":false}", body4);

    ResponseException ex = expectThrows(ResponseException.class,
        () -> restClient.performRequest("POST", "/_xpack/license/start_basic"));
    Response response5 = ex.getResponse();
    String body5 = Streams.copyToString(new InputStreamReader(response5.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(403, response5.getStatusLine().getStatusCode());
    assertTrue(body5.contains("\"basic_was_started\":false"));
    assertTrue(body5.contains("\"acknowledged\":true"));
    assertTrue(body5.contains("\"error_message\":\"Operation failed: Current license is basic.\""));
  }

  public void testUnacknowledgedStartBasicLicense() throws Exception {
    LicensingClient licensingClient = new LicensingClient(client());
    License license = TestUtils.generateSignedLicense("trial",  License.VERSION_CURRENT, -1,
TimeValue.timeValueHours(24));
    licensingClient.preparePutLicense(license).get();

    assertBusy(() -> {
      GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();
      assertEquals("trial", getLicenseResponse.license().type());
    });

    Response response2 = getRestClient().performRequest("POST", "/_xpack/license/start_basic");
    String body2 = Streams.copyToString(new InputStreamReader(response2.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response2.getStatusLine().getStatusCode());
    assertTrue(body2.contains("\"acknowledged\":false"));
    assertTrue(body2.contains("\"basic_was_started\":false"));
    assertTrue(body2.contains("\"error_message\":\"Operation failed: Needs acknowledgement.\""));
    assertTrue(body2.contains("\"message\":\"This license update requires acknowledgement. To acknowledge the
license, " +
        "please read the following messages and call /start_basic again, this time with the
```

```
\\\"acknowledge=true\\\""));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import com.carrotsearch.randomizedtesting.RandomizedTest;
import org.elasticsearch.action.ActionListener;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.Strings;
import org.elasticsearch.common.joda.DateMathParser;
import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.licensor.LicenseSigner;
import org.hamcrest.MatcherAssert;
import org.joda.time.format.DateTimeFormatter;
import org.junit.Assert;

import java.io.IOException;
import java.io.InputStream;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.StandardCopyOption;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.atomic.AtomicReference;

import static com.carrotsearch.randomizedtesting.RandomizedTest.randomBoolean;
import static com.carrotsearch.randomizedtesting.RandomizedTest.randomInt;
import static org.apache.lucene.util.LuceneTestCase.createTempFile;
import static org.elasticsearch.common.xcontent.XContentFactory.jsonBuilder;
import static org.elasticsearch.test.ESTestCase.randomAlphaOfLength;
import static org.elasticsearch.test.ESTestCase.randomFrom;
import static org.elasticsearch.test.ESTestCase.randomIntBetween;
import static org.hamcrest.core.IsEqual.equalTo;
import static org.junit.Assert.assertThat;
```

```java
public class TestUtils {

  private static final FormatDateTimeFormatter formatDateTimeFormatter = Joda.forPattern("yyyy-MM-dd");
  private static final DateMathParser dateMathParser = new DateMathParser(formatDateTimeFormatter);
  private static final DateTimeFormatter dateTimeFormatter = formatDateTimeFormatter.printer();

  public static String dateMathString(String time, final long now) {
    return dateTimeFormatter.print(dateMathParser.parse(time, () -> now));
  }

  public static long dateMath(String time, final long now) {
    return dateMathParser.parse(time, () -> now);
  }

  public static LicenseSpec generateRandomLicenseSpec(int version) {
    boolean datesInMillis = randomBoolean();
    long now = System.currentTimeMillis();
    String uid = UUID.randomUUID().toString();
    String feature = "feature__" + randomInt();
    String issuer = "issuer__"  + randomInt();
    String issuedTo = "issuedTo__" + randomInt();
    final String type;
    final String subscriptionType;
    if (version < License.VERSION_NO_FEATURE_TYPE) {
      subscriptionType = randomFrom("gold", "silver", "platinum");
      type = "subscription";//randomFrom("subscription", "internal", "development");
    } else {
      subscriptionType = null;
      type = randomFrom("basic", "dev", "gold", "silver", "platinum");
    }
    int maxNodes = RandomizedTest.randomIntBetween(5, 100);
    if (datesInMillis) {
      long issueDateInMillis = dateMath("now", now);
      long expiryDateInMillis = dateMath("now+10d/d", now);
      return new LicenseSpec(version, uid, feature, issueDateInMillis, expiryDateInMillis, type, subscriptionType, issuedTo, issuer,
          maxNodes);
    } else {
      String issueDate = dateMathString("now", now);
      String expiryDate = dateMathString("now+10d/d", now);
      return new LicenseSpec(version, uid, feature, issueDate, expiryDate, type, subscriptionType, issuedTo,
issuer, maxNodes);
    }
  }

  public static String generateLicenseSpecString(LicenseSpec licenseSpec) throws IOException {
    XContentBuilder licenses = jsonBuilder();
```

```java
            licenses.startObject();
            licenses.startArray("licenses");
            licenses.startObject()
                .field("uid", licenseSpec.uid)
                .field("type", licenseSpec.type)
                .field("subscription_type", licenseSpec.subscriptionType)
                .field("issued_to", licenseSpec.issuedTo)
                .field("issuer", licenseSpec.issuer)
                .field("feature", licenseSpec.feature)
                .field("max_nodes", licenseSpec.maxNodes);

            if (licenseSpec.issueDate != null) {
                licenses.field("issue_date", licenseSpec.issueDate);
            } else {
                licenses.field("issue_date_in_millis", licenseSpec.issueDateInMillis);
            }
            if (licenseSpec.expiryDate != null) {
                licenses.field("expiry_date", licenseSpec.expiryDate);
            } else {
                licenses.field("expiry_date_in_millis", licenseSpec.expiryDateInMillis);
            }
            licenses.field("version", licenseSpec.version);
            licenses.endObject();
            licenses.endArray();
            licenses.endObject();
            return Strings.toString(licenses);
        }

        public static License generateLicenses(LicenseSpec spec) {
            License.Builder builder = License.builder()
                .uid(spec.uid)
                .feature(spec.feature)
                .type(spec.type)
                .subscriptionType(spec.subscriptionType)
                .issuedTo(spec.issuedTo)
                .issuer(spec.issuer)
                .maxNodes(spec.maxNodes);

            if (spec.expiryDate != null) {
                builder.expiryDate(DateUtils.endOfTheDay(spec.expiryDate));
            } else {
                builder.expiryDate(spec.expiryDateInMillis);
            }
            if (spec.issueDate != null) {
                builder.issueDate(DateUtils.beginningOfTheDay(spec.issueDate));
            } else {
                builder.issueDate(spec.issueDateInMillis);
            }
```

```java
        return builder.build();
    }

    public static void assertLicenseSpec(LicenseSpec spec, License license) {
        MatcherAssert.assertThat(license.uid(), equalTo(spec.uid));
        MatcherAssert.assertThat(license.issuedTo(), equalTo(spec.issuedTo));
        MatcherAssert.assertThat(license.issuer(), equalTo(spec.issuer));
        MatcherAssert.assertThat(license.type(), equalTo(spec.type));
        MatcherAssert.assertThat(license.maxNodes(), equalTo(spec.maxNodes));
        if (spec.issueDate != null) {
            MatcherAssert.assertThat(license.issueDate(), equalTo(DateUtils.beginningOfTheDay(spec.issueDate)));
        } else {
            MatcherAssert.assertThat(license.issueDate(), equalTo(spec.issueDateInMillis));
        }
        if (spec.expiryDate != null) {
            MatcherAssert.assertThat(license.expiryDate(), equalTo(DateUtils.endOfTheDay(spec.expiryDate)));
        } else {
            MatcherAssert.assertThat(license.expiryDate(), equalTo(spec.expiryDateInMillis));
        }
    }

    public static class LicenseSpec {
        public final int version;
        public final String feature;
        public final String issueDate;
        public final long issueDateInMillis;
        public final String expiryDate;
        public final long expiryDateInMillis;
        public final String uid;
        public final String type;
        public final String subscriptionType;
        public final String issuedTo;
        public final String issuer;
        public final int maxNodes;

        public LicenseSpec(String issueDate, String expiryDate) {
            this(License.VERSION_CURRENT, UUID.randomUUID().toString(), "feature", issueDate, expiryDate,
"trial", "none", "customer",
                    "elasticsearch", 5);
        }

        public LicenseSpec(int version, String uid, String feature, long issueDateInMillis, long expiryDateInMillis,
String type,
                        String subscriptionType, String issuedTo, String issuer, int maxNodes) {
            this.version = version;
            this.feature = feature;
            this.issueDateInMillis = issueDateInMillis;
            this.issueDate = null;
```

```java
            this.expiryDateInMillis = expiryDateInMillis;
            this.expiryDate = null;
            this.uid = uid;
            this.type = type;
            this.subscriptionType = subscriptionType;
            this.issuedTo = issuedTo;
            this.issuer = issuer;
            this.maxNodes = maxNodes;
        }

        public LicenseSpec(int version, String uid, String feature, String issueDate, String expiryDate, String type,
                    String subscriptionType, String issuedTo, String issuer, int maxNodes) {
            this.version = version;
            this.feature = feature;
            this.issueDate = issueDate;
            this.issueDateInMillis = -1;
            this.expiryDate = expiryDate;
            this.expiryDateInMillis = -1;
            this.uid = uid;
            this.type = type;
            this.subscriptionType = subscriptionType;
            this.issuedTo = issuedTo;
            this.issuer = issuer;
            this.maxNodes = maxNodes;
        }
    }
    private static Path getTestPriKeyPath() throws Exception {
        return getResourcePath("/private.key");
    }

    private static Path getTestPubKeyPath() throws Exception {
        return getResourcePath("/public.key");
    }

    public static String dumpLicense(License license) throws Exception {
        XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
        builder.startObject();
        builder.startObject("license");
        license.toInnerXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        return Strings.toString(builder);
    }

    public static License generateSignedLicense(TimeValue expiryDuration) throws Exception {
        return generateSignedLicense(null, -1, expiryDuration);
    }
```

```java
    public static License generateSignedLicense(String type, TimeValue expiryDuration) throws Exception {
        return generateSignedLicense(type, -1, expiryDuration);
    }

    public static License generateSignedLicense(long issueDate, TimeValue expiryDuration) throws Exception {
        return generateSignedLicense(null, issueDate, expiryDuration);
    }

    public static License generateSignedLicense(String type, long issueDate, TimeValue expiryDuration) throws
Exception {
        return generateSignedLicense(type, randomIntBetween(License.VERSION_START,
License.VERSION_CURRENT), issueDate, expiryDuration);
    }

    public static License generateSignedLicenseOldSignature() {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
            .uid(UUID.randomUUID().toString())
            .version(License.VERSION_START_DATE)
            .issuedTo("customer")
            .maxNodes(5)
            .type("trial")
            .issueDate(issueDate)
            .expiryDate(issueDate + TimeValue.timeValueHours(24).getMillis());
        return SelfGeneratedLicense.create(specBuilder, License.VERSION_START_DATE);
    }

    /**
     * This method which chooses the license type randomly if the type is null. However, it will not randomly
     * choose trial or basic types as those types can only be self-generated.
     */
    public static License generateSignedLicense(String type, int version, long issueDate, TimeValue expiryDuration)
throws Exception {
        long issue = (issueDate != -1L) ? issueDate : System.currentTimeMillis() -
TimeValue.timeValueHours(2).getMillis();
        final String licenseType;
        if (version < License.VERSION_NO_FEATURE_TYPE) {
            licenseType = randomFrom("subscription", "internal", "development");
        } else {
            licenseType = (type != null) ? type : randomFrom("silver", "dev", "gold", "platinum");
        }
        final License.Builder builder = License.builder()
            .uid(UUID.randomUUID().toString())
            .version(version)
            .expiryDate(System.currentTimeMillis() + expiryDuration.getMillis())
            .issueDate(issue)
            .type(licenseType)
            .issuedTo("customer")
```

```java
        .issuer("elasticsearch")
        .maxNodes(5);
    if (version == License.VERSION_START) {
        builder.subscriptionType((type != null) ? type : randomFrom("dev", "gold", "platinum", "silver"));
        builder.feature(randomAlphaOfLength(10));
    }
    final LicenseSigner signer = new LicenseSigner(getTestPriKeyPath(), getTestPubKeyPath());
    return signer.sign(builder.build());
}

public static License generateSignedLicense(License.Builder builder) throws Exception {
    LicenseSigner signer = new LicenseSigner(getTestPriKeyPath(), getTestPubKeyPath());
    return signer.sign(builder.build());
}

public static License generateExpiredNonBasicLicense(long expiryDate) throws Exception {
    return generateExpiredNonBasicLicense(randomFrom("silver", "dev", "gold", "platinum"), expiryDate);
}

public static License generateExpiredNonBasicLicense() throws Exception {
    return generateExpiredNonBasicLicense(randomFrom("silver", "dev", "gold", "platinum"));
}

public static License generateExpiredNonBasicLicense(String type) throws Exception {
    return generateExpiredNonBasicLicense(type,
        System.currentTimeMillis() - TimeValue.timeValueHours(randomIntBetween(1, 10)).getMillis());
}

public static License generateExpiredNonBasicLicense(String type, long expiryDate) throws Exception {
    final License.Builder builder = License.builder()
        .uid(UUID.randomUUID().toString())
        .version(License.VERSION_CURRENT)
        .expiryDate(expiryDate)
        .issueDate(expiryDate - TimeValue.timeValueMinutes(10).getMillis())
        .type(type)
        .issuedTo("customer")
        .issuer("elasticsearch")
        .maxNodes(5);
    LicenseSigner signer = new LicenseSigner(getTestPriKeyPath(), getTestPubKeyPath());
    return signer.sign(builder.build());
}

private static Path getResourcePath(String resource) throws Exception {
    Path resourceFile = createTempFile();
    try (InputStream resourceInput = TestUtils.class.getResourceAsStream(resource)) {
        Files.copy(resourceInput, resourceFile, StandardCopyOption.REPLACE_EXISTING);
    }
    return resourceFile;
```

```
    }

    public static void registerAndAckSignedLicenses(final LicenseService licenseService, License license,
                                final LicensesStatus expectedStatus) {
        PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(license).acknowledge(true);
        final CountDownLatch latch = new CountDownLatch(1);
        final AtomicReference<LicensesStatus> status = new AtomicReference<>();
        licenseService.registerLicense(putLicenseRequest, new ActionListener<PutLicenseResponse>() {
            @Override
            public void onResponse(PutLicenseResponse licensesUpdateResponse) {
                status.set(licensesUpdateResponse.status());
                latch.countDown();
            }

            @Override
            public void onFailure(Exception e) {
                latch.countDown();
            }
        });
        try {
            latch.await();
        } catch (InterruptedException e) {
            Assert.fail(e.getMessage());
        }
        assertThat(status.get(), equalTo(expectedStatus));
    }

    public static class AssertingLicenseState extends XPackLicenseState {
        public final List<License.OperationMode> modeUpdates = new ArrayList<>();
        public final List<Boolean> activeUpdates = new ArrayList<>();

        public AssertingLicenseState() {
            super(Settings.EMPTY);
        }

        @Override
        void update(License.OperationMode mode, boolean active) {
            modeUpdates.add(mode);
            activeUpdates.add(active);
        }
    }

    /**
     * A license state that makes the {@link #update(License.OperationMode, boolean)}
     * method public for use in tests.
     */
    public static class UpdatableLicenseState extends XPackLicenseState {
        public UpdatableLicenseState() {
```

```java
            this(Settings.EMPTY);
        }

        public UpdatableLicenseState(Settings settings) {
            super(settings);
        }

        @Override
        public void update(License.OperationMode mode, boolean active) {
            super.update(mode, active);
        }
    }

    public static void putLicense(MetaData.Builder builder, License license) {
        builder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.cluster.ClusterName;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.common.settings.Settings;
import org.mockito.ArgumentCaptor;
import org.mockito.Mockito;

import java.util.UUID;

import static org.elasticsearch.license.TestUtils.dateMath;
import static org.mockito.Matchers.any;
import static org.mockito.Mockito.verify;
import static org.mockito.Mockito.when;

public class LicenseRegistrationTests extends AbstractLicenseServiceTestCase {

    public void testSelfGeneratedTrialLicense() throws Exception {
        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(null, licenseState, Settings.EMPTY, "trial");
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
        licenseService.start();

        ClusterState state = ClusterState.builder(new ClusterName("a")).build();
```

```java
        ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
        verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
        ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
        LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
        assertNotNull(licenseMetaData);
        assertNotNull(licenseMetaData.getLicense());
        assertFalse(licenseMetaData.isEligibleForTrial());
        assertEquals("trial", licenseMetaData.getLicense().type());
        assertEquals(clock.millis() +
LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.millis(),
            licenseMetaData.getLicense().expiryDate());
    }

    public void testSelfGeneratedBasicLicense() throws Exception {
        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(null, licenseState, Settings.EMPTY, "basic");
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
        licenseService.start();

        ClusterState state = ClusterState.builder(new ClusterName("a")).build();
        ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
        verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
        ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
        LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
        assertNotNull(licenseMetaData);
        assertNotNull(licenseMetaData.getLicense());
        assertTrue(licenseMetaData.isEligibleForTrial());
        assertEquals("basic", licenseMetaData.getLicense().type());
        assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS,
licenseMetaData.getLicense().expiryDate());
    }

    public void testNonSelfGeneratedBasicLicenseIsReplaced() throws Exception {
        long now = System.currentTimeMillis();
        String uid = UUID.randomUUID().toString();
        final License.Builder builder = License.builder()
            .uid(uid)
            .version(License.VERSION_CURRENT)
            .expiryDate(dateMath("now+2h", now))
            .startDate(now)
            .issueDate(now)
            .type("basic")
            .issuedTo("customer")
            .issuer("elasticsearch")
            .maxNodes(5);
        License license = TestUtils.generateSignedLicense(builder);
```

```
    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    setInitialState(license, licenseState, Settings.EMPTY);
    when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
    licenseService.start();

    MetaData.Builder mdBuilder = MetaData.builder();
    mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
    ClusterState state = ClusterState.builder(new ClusterName("a")).metaData(mdBuilder.build()).build();
    ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
    verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
    ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
    LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
    assertNotNull(licenseMetaData);
    assertNotNull(licenseMetaData.getLicense());
    assertTrue(licenseMetaData.isEligibleForTrial());
    assertEquals("basic", licenseMetaData.getLicense().type());
    assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS,
licenseMetaData.getLicense().expiryDate());
    assertEquals(uid, licenseMetaData.getLicense().uid());
  }

  public void testExpiredSelfGeneratedBasicLicenseIsExtended() throws Exception {
    long now = System.currentTimeMillis();
    String uid = UUID.randomUUID().toString();
    License.Builder builder = License.builder()
         .uid(uid)
         .issuedTo("name")
         .maxNodes(1000)
         .issueDate(dateMath("now-10h", now))
         .type("basic")
         .expiryDate(dateMath("now-2h", now));
    License license = SelfGeneratedLicense.create(builder, License.VERSION_CURRENT);

    XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
    setInitialState(license, licenseState, Settings.EMPTY);
    when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
    licenseService.start();

    MetaData.Builder mdBuilder = MetaData.builder();
    mdBuilder.putCustom(LicensesMetaData.TYPE, new LicensesMetaData(license, null));
    ClusterState state = ClusterState.builder(new ClusterName("a")).metaData(mdBuilder.build()).build();
    ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
    verify(clusterService, Mockito.times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
    ClusterState stateWithLicense = stateUpdater.getValue().execute(state);
    LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
```

```java
        assertNotNull(licenseMetaData);
        assertNotNull(licenseMetaData.getLicense());
        assertTrue(licenseMetaData.isEligibleForTrial());
        assertEquals("basic", licenseMetaData.getLicense().type());
        assertEquals(LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS,
licenseMetaData.getLicense().expiryDate());
        assertEquals(uid, licenseMetaData.getLicense().uid());
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterModule;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.metadata.RepositoriesMetaData;
import org.elasticsearch.cluster.metadata.RepositoryMetaData;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.NamedXContentRegistry;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.ToXContent.Params;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentParser;
import org.elasticsearch.test.ESTestCase;

import java.util.Collections;
import java.util.UUID;
import java.util.stream.Collectors;
import java.util.stream.Stream;

import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.notNullValue;
import static org.hamcrest.Matchers.nullValue;

public class LicensesMetaDataSerializationTests extends ESTestCase {

    public void testXContentSerializationOneSignedLicense() throws Exception {
        License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(2));
        LicensesMetaData licensesMetaData = new LicensesMetaData(license, null);
        XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
```

```java
        licensesMetaData.toXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        LicensesMetaData licensesMetaDataFromXContent =
getLicensesMetaDataFromXContent(createParser(builder));
        assertThat(licensesMetaDataFromXContent.getLicense(), equalTo(license));
        assertNull(licensesMetaDataFromXContent.getMostRecentTrialVersion());
    }

    public void testXContentSerializationOneSignedLicenseWithUsedTrial() throws Exception {
        License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(2));
        LicensesMetaData licensesMetaData = new LicensesMetaData(license, Version.CURRENT);
        XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
        licensesMetaData.toXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        LicensesMetaData licensesMetaDataFromXContent =
getLicensesMetaDataFromXContent(createParser(builder));
        assertThat(licensesMetaDataFromXContent.getLicense(), equalTo(license));
        assertEquals(licensesMetaDataFromXContent.getMostRecentTrialVersion(), Version.CURRENT);
    }

    public void testLicenseMetadataParsingDoesNotSwallowOtherMetaData() throws Exception {
        new Licensing(Settings.EMPTY); // makes sure LicensePlugin is registered in Custom MetaData
        License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(2));
        LicensesMetaData licensesMetaData = new LicensesMetaData(license, Version.CURRENT);
        RepositoryMetaData repositoryMetaData = new RepositoryMetaData("repo", "fs", Settings.EMPTY);
        RepositoriesMetaData repositoriesMetaData = new
RepositoriesMetaData(Collections.singletonList(repositoryMetaData));
        final MetaData.Builder metaDataBuilder = MetaData.builder();
        if (randomBoolean()) { // random order of insertion
            metaDataBuilder.putCustom(licensesMetaData.getWriteableName(), licensesMetaData);
            metaDataBuilder.putCustom(repositoriesMetaData.getWriteableName(), repositoriesMetaData);
        } else {
            metaDataBuilder.putCustom(repositoriesMetaData.getWriteableName(), repositoriesMetaData);
            metaDataBuilder.putCustom(licensesMetaData.getWriteableName(), licensesMetaData);
        }
        // serialize metadata
        XContentBuilder builder = XContentFactory.jsonBuilder();
        Params params = new
ToXContent.MapParams(Collections.singletonMap(MetaData.CONTEXT_MODE_PARAM,
MetaData.CONTEXT_MODE_GATEWAY));
        builder.startObject();
        builder = metaDataBuilder.build().toXContent(builder, params);
        builder.endObject();
        // deserialize metadata again
```

```java
            MetaData metaData = MetaData.Builder.fromXContent(createParser(builder));
            // check that custom metadata still present
            assertThat(metaData.custom(licensesMetaData.getWriteableName()), notNullValue());
            assertThat(metaData.custom(repositoriesMetaData.getWriteableName()), notNullValue());
        }


    public void testXContentSerializationOneTrial() throws Exception {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
            .uid(UUID.randomUUID().toString())
            .issuedTo("customer")
            .maxNodes(5)
            .issueDate(issueDate)
            .type(randomBoolean() ? "trial" : "basic")
            .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
        final License trialLicense = SelfGeneratedLicense.create(specBuilder, License.VERSION_CURRENT);
        LicensesMetaData licensesMetaData = new LicensesMetaData(trialLicense, Version.CURRENT);
        XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
        licensesMetaData.toXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        LicensesMetaData licensesMetaDataFromXContent =
getLicensesMetaDataFromXContent(createParser(builder));
        assertThat(licensesMetaDataFromXContent.getLicense(), equalTo(trialLicense));
        assertEquals(licensesMetaDataFromXContent.getMostRecentTrialVersion(), Version.CURRENT);
    }

    public void testLicenseTombstoneFromXContext() throws Exception {
        final XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
        builder.nullField("license");
        builder.endObject();
        builder.endObject();
        LicensesMetaData metaDataFromXContent = getLicensesMetaDataFromXContent(createParser(builder));
        assertThat(metaDataFromXContent.getLicense(), equalTo(LicensesMetaData.LICENSE_TOMBSTONE));
    }

    public void testLicenseTombstoneWithUsedTrialFromXContext() throws Exception {
        final XContentBuilder builder = XContentFactory.jsonBuilder();
        builder.startObject();
        builder.startObject("licenses");
        builder.nullField("license");
        builder.field("trial_license", Version.CURRENT.toString());
        builder.endObject();
        builder.endObject();
```

```java
            LicensesMetaData metaDataFromXContent = getLicensesMetaDataFromXContent(createParser(builder));
            assertThat(metaDataFromXContent.getLicense(), equalTo(LicensesMetaData.LICENSE_TOMBSTONE));
            assertEquals(metaDataFromXContent.getMostRecentTrialVersion(), Version.CURRENT);
        }

    private static LicensesMetaData getLicensesMetaDataFromXContent(XContentParser parser) throws Exception {
        parser.nextToken(); // consume null
        parser.nextToken(); // consume "licenses"
        LicensesMetaData licensesMetaDataFromXContent = LicensesMetaData.fromXContent(parser);
        parser.nextToken(); // consume endObject
        assertThat(parser.nextToken(), nullValue());
        return licensesMetaDataFromXContent;
    }

    @Override
    protected NamedXContentRegistry xContentRegistry() {
        return new NamedXContentRegistry(Stream.concat(
            new Licensing(Settings.EMPTY).getNamedXContent().stream(),
            ClusterModule.getNamedXWriteables().stream()
        ).collect(Collectors.toList()));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.client.Response;
import org.elasticsearch.client.ResponseException;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.common.io.Streams;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESIntegTestCase;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackClientPlugin;

import java.io.InputStreamReader;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.test.ESIntegTestCase.Scope.SUITE;

@ESIntegTestCase.ClusterScope(scope = SUITE)
```

```java
public class StartTrialLicenseTests extends AbstractLicensesIntegrationTestCase {

    @Override
    protected boolean addMockHttpTransport() {
        return false; // enable http
    }

    @Override
    protected Settings nodeSettings(int nodeOrdinal) {
        return Settings.builder()
                .put(super.nodeSettings(nodeOrdinal))
                .put("node.data", true)
                .put(LicenseService.SELF_GENERATED_LICENSE_TYPE.getKey(), "basic").build();
    }

    @Override
    protected Collection<Class<? extends Plugin>> nodePlugins() {
        return Arrays.asList(LocalStateCompositeXPackPlugin.class, Netty4Plugin.class);
    }

    @Override
    protected Collection<Class<? extends Plugin>> transportClientPlugins() {
        return Arrays.asList(XPackClientPlugin.class, Netty4Plugin.class);
    }

    public void testStartTrial() throws Exception {
        LicensingClient licensingClient = new LicensingClient(client());
        ensureStartingWithBasic();

        RestClient restClient = getRestClient();
        Response response = restClient.performRequest("GET", "/_xpack/license/trial_status");
        String body = Streams.copyToString(new InputStreamReader(response.getEntity().getContent(),
StandardCharsets.UTF_8));
        assertEquals(200, response.getStatusLine().getStatusCode());
        assertEquals("{\"eligible_to_start_trial\":true}", body);

        // Test that starting will fail without acknowledgement
        Response response2 = restClient.performRequest("POST", "/_xpack/license/start_trial");
        String body2 = Streams.copyToString(new InputStreamReader(response2.getEntity().getContent(),
StandardCharsets.UTF_8));
        assertEquals(200, response2.getStatusLine().getStatusCode());
        assertTrue(body2.contains("\"trial_was_started\":false"));
        assertTrue(body2.contains("\"error_message\":\"Operation failed: Needs acknowledgement.\""));
        assertTrue(body2.contains("\"acknowledged\":false"));

        assertBusy(() -> {
            GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();
            assertEquals("basic", getLicenseResponse.license().type());
```

```
    });

    String type = randomFrom(LicenseService.VALID_TRIAL_TYPES);

    Response response3 = restClient.performRequest("POST",
"/_xpack/license/start_trial?acknowledge=true&type=" + type);
    String body3 = Streams.copyToString(new InputStreamReader(response3.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response3.getStatusLine().getStatusCode());
    assertTrue(body3.contains("\"trial_was_started\":true"));
    assertTrue(body3.contains("\"type\":\"" + type + "\""));
    assertTrue(body3.contains("\"acknowledged\":true"));

    assertBusy(() -> {
      GetLicenseResponse postTrialLicenseResponse = licensingClient.prepareGetLicense().get();
      assertEquals(type, postTrialLicenseResponse.license().type());
    });

    Response response4 = restClient.performRequest("GET", "/_xpack/license/trial_status");
    String body4 = Streams.copyToString(new InputStreamReader(response4.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(200, response4.getStatusLine().getStatusCode());
    assertEquals("{\"eligible_to_start_trial\":false}", body4);

    String secondAttemptType = randomFrom(LicenseService.VALID_TRIAL_TYPES);

    ResponseException ex = expectThrows(ResponseException.class,
        () -> restClient.performRequest("POST", "/_xpack/license/start_trial?acknowledge=true&type=" +
secondAttemptType));
    Response response5 = ex.getResponse();
    String body5 = Streams.copyToString(new InputStreamReader(response5.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(403, response5.getStatusLine().getStatusCode());
    assertTrue(body5.contains("\"trial_was_started\":false"));
    assertTrue(body5.contains("\"error_message\":\"Operation failed: Trial was already activated.\""));
  }

  public void testInvalidType() throws Exception {
    ensureStartingWithBasic();

    ResponseException ex = expectThrows(ResponseException.class, () ->
        getRestClient().performRequest("POST", "/_xpack/license/start_trial?type=basic"));
    Response response = ex.getResponse();
    String body = Streams.copyToString(new InputStreamReader(response.getEntity().getContent(),
StandardCharsets.UTF_8));
    assertEquals(400, response.getStatusLine().getStatusCode());
    assertTrue(body.contains("\"type\":\"illegal_argument_exception\""));
    assertTrue(body.contains("\"reason\":\"Cannot start trial of type [basic]. Valid trial types are ["));
```

```
    }

    private void ensureStartingWithBasic() throws Exception {
        LicensingClient licensingClient = new LicensingClient(client());
        GetLicenseResponse getLicenseResponse = licensingClient.prepareGetLicense().get();

        if ("basic".equals(getLicenseResponse.license().type()) == false) {
            licensingClient.preparePostStartBasic().setAcknowledge(true).get();
        }

        assertBusy(() -> {
            GetLicenseResponse postTrialLicenseResponse = licensingClient.prepareGetLicense().get();
            assertEquals("basic", postTrialLicenseResponse.license().type());
        });
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.test.ESTestCase;

import java.util.Arrays;

import static org.hamcrest.Matchers.contains;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.hasSize;

public class LicenseUtilsTests extends ESTestCase {

    public void testNewExpirationException() {
        for (String feature : Arrays.asList("feature", randomAlphaOfLength(5), null, "")) {
            ElasticsearchSecurityException exception = LicenseUtils.newComplianceException(feature);
            assertNotNull(exception);
            assertThat(exception.getMetadataKeys(), contains(LicenseUtils.EXPIRED_FEATURE_METADATA));
            assertThat(exception.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA), hasSize(1));
            assertThat(exception.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA).iterator().next(),
equalTo(feature));
        }
    }

    public void testIsLicenseExpiredException() {
        ElasticsearchSecurityException exception = LicenseUtils.newComplianceException("feature");
        assertTrue(LicenseUtils.isLicenseExpiredException(exception));
```

```java
            exception = new ElasticsearchSecurityException("msg");
            assertFalse(LicenseUtils.isLicenseExpiredException(exception));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.test.ESTestCase;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.util.Base64;
import java.util.Collections;
import java.util.UUID;

import static org.elasticsearch.license.CryptUtils.encryptV3Format;
import static org.hamcrest.Matchers.equalTo;


public class SelfGeneratedLicenseTests extends ESTestCase {

    public void testBasic() throws Exception {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
                .uid(UUID.randomUUID().toString())
                .issuedTo("customer")
                .maxNodes(5)
                .type(randomBoolean() ? "trial" : "basic")
                .issueDate(issueDate)
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
        License trialLicense = SelfGeneratedLicense.create(specBuilder, License.VERSION_CURRENT);
        assertThat(SelfGeneratedLicense.verify(trialLicense), equalTo(true));
    }

    public void testTampered() throws Exception {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
```

```
                .uid(UUID.randomUUID().toString())
                .issuedTo("customer")
                .type(randomBoolean() ? "trial" : "basic")
                .maxNodes(5)
                .issueDate(issueDate)
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
        License trialLicense = SelfGeneratedLicense.create(specBuilder, License.VERSION_CURRENT);
        final String originalSignature = trialLicense.signature();
        License tamperedLicense = License.builder().fromLicenseSpec(trialLicense, originalSignature)
                .expiryDate(System.currentTimeMillis() + TimeValue.timeValueHours(5).getMillis())
                .build();
        assertThat(SelfGeneratedLicense.verify(trialLicense), equalTo(true));
        assertThat(SelfGeneratedLicense.verify(tamperedLicense), equalTo(false));
    }

    public void testFrom1x() throws Exception {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
                .uid(UUID.randomUUID().toString())
                .issuedTo("customer")
                .type("subscription")
                .subscriptionType("trial")
                .issuer("elasticsearch")
                .feature("")
                .version(License.VERSION_START)
                .maxNodes(5)
                .issueDate(issueDate)
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis());
        License pre20TrialLicense = specBuilder.build();
        License license =
SelfGeneratedLicense.create(License.builder().fromPre20LicenseSpec(pre20TrialLicense).type("trial"),
            License.VERSION_CURRENT);
        assertThat(SelfGeneratedLicense.verify(license), equalTo(true));
    }

    public void testTrialLicenseVerifyWithOlderVersion() throws Exception {
        long issueDate = System.currentTimeMillis();
        License.Builder specBuilder = License.builder()
                .issuedTo("customer")
                .maxNodes(5)
                .issueDate(issueDate)
                .expiryDate(issueDate + TimeValue.timeValueHours(2).getMillis())
                .feature("")
                .subscriptionType("trial")
                .version(1);
        License trialLicenseV1 = createTrialLicense(specBuilder);
        assertThat(SelfGeneratedLicense.verify(trialLicenseV1), equalTo(true));
    }
```

```java
    private static License createTrialLicense(License.Builder specBuilder) {
        License spec = specBuilder
                .type(randomBoolean() ? "trial" : "basic")
                .issuer("elasticsearch")
                .uid(UUID.randomUUID().toString())
                .build();
        final String signature;
        try {
            XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
            spec.toXContent(contentBuilder, new
ToXContent.MapParams(Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true")));
            byte[] encrypt = encryptV3Format(BytesReference.toBytes(BytesReference.bytes(contentBuilder)));
            byte[] bytes = new byte[4 + 4 + encrypt.length];
            ByteBuffer byteBuffer = ByteBuffer.wrap(bytes);
            byteBuffer.putInt(-spec.version())
                    .putInt(encrypt.length)
                    .put(encrypt);
            signature = Base64.getEncoder().encodeToString(bytes);
        } catch (IOException e) {
            throw new IllegalStateException(e);
        }
        return License.builder().fromLicenseSpec(spec, signature).build();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.test.ESTestCase;

import java.util.Locale;

import static org.elasticsearch.license.License.OperationMode;
import static org.hamcrest.Matchers.equalTo;

/**
 * Tests {@link License.OperationMode} for correctness.
 * <p>
 * If you change the behavior of these tests, then it means that licensing changes across the products!
 */
public class LicenseOperationModeTests extends ESTestCase {
    public void testResolveTrial() {
        // assert 1.x BWC
        assertResolve(OperationMode.TRIAL, "nONE", "DEv", "deveLopment");
```

```java
    // assert expected (2.x+) variant
    assertResolve(OperationMode.TRIAL, "tRiAl", "trial");
}

public void testResolveBasic() {
    // assert expected (2.x+) variant (note: no 1.x variant of BASIC)
    assertResolve(OperationMode.BASIC, "bAsIc", "basic");
}

public void testResolveStandard() {
    // assert expected (2.x+) variant (note: no 1.x variant of STANDARD)
    assertResolve(OperationMode.STANDARD, "StAnDARd", "standard");
}

public void testResolveGold() {
    // assert expected (2.x+) variant (note: no different 1.x variant of GOLD)
    assertResolve(OperationMode.GOLD, "SiLvEr", "gOlD", "silver", "gold");
}

public void testResolvePlatinum() {
    // assert 1.x BWC
    assertResolve(OperationMode.PLATINUM, "iNtErNaL");
    // assert expected (2.x+) variant
    assertResolve(OperationMode.PLATINUM, "PlAtINum", "platinum");
}

public void testResolveUnknown() {
    // 'enterprise' is a type that exists in cloud but should be rejected under normal operation
    // See https://github.com/elastic/x-plugins/issues/3371
    String[] types = { "unknown", "fake", "enterprise" };

    for (String type : types) {
        try {
            OperationMode.resolve(type);

            fail(String.format(Locale.ROOT, "[%s] should not be recognized as an operation mode", type));
        }
        catch (IllegalArgumentException e) {
            assertThat(e.getMessage(), equalTo("unknown type [" + type + "]"));
        }
    }
}

private static void assertResolve(OperationMode expected, String... types) {
    for (String type : types) {
        assertThat(OperationMode.resolve(type), equalTo(expected));
    }
}
```

```
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.test.ESTestCase;

import static org.elasticsearch.common.unit.TimeValue.timeValueMillis;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.nullValue;

public class ExpirationCallbackTests extends ESTestCase {

    public void testPostExpirationDelay() throws Exception {
        TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
        TimeValue min = TimeValue.timeValueSeconds(1);
        TimeValue max = TimeValue.timeValueSeconds(4);
        TimeValue frequency = TimeValue.timeValueSeconds(1);
        NoopPostExpirationCallback post = new NoopPostExpirationCallback(min, max, frequency);
        long now = System.currentTimeMillis();
        long expiryDate = now + expiryDuration.getMillis();
        assertThat(post.delay(expiryDate, now),
            equalTo(TimeValue.timeValueMillis(expiryDuration.getMillis() + min.getMillis()))); // before license
expiry
        assertThat(post.delay(expiryDate, expiryDate), equalTo(min)); // on license expiry
        int latestValidTriggerDelay = (int) (expiryDuration.getMillis() + max.getMillis());
        int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() + min.getMillis());
        assertExpirationCallbackDelay(post, expiryDuration.millis(), latestValidTriggerDelay,
earliestValidTriggerDelay);
    }

    public void testPreExpirationDelay() throws Exception {
        TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
        TimeValue min = TimeValue.timeValueSeconds(1);
        TimeValue max = TimeValue.timeValueSeconds(4);
        TimeValue frequency = TimeValue.timeValueSeconds(1);
        NoopPreExpirationCallback pre = new NoopPreExpirationCallback(min, max, frequency);
        long now = System.currentTimeMillis();
        long expiryDate = now + expiryDuration.getMillis();
        assertThat(pre.delay(expiryDate, expiryDate), nullValue()); // on license expiry
        int latestValidTriggerDelay = (int) (expiryDuration.getMillis() - min.getMillis());
        int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() - max.getMillis());
        assertExpirationCallbackDelay(pre, expiryDuration.millis(), latestValidTriggerDelay,
earliestValidTriggerDelay);
```

```java
        }

        public void testPostExpirationWithNullMax() throws Exception {
            int postExpirySeconds = randomIntBetween(5, 10);
            TimeValue postExpiryDuration = TimeValue.timeValueSeconds(postExpirySeconds);
            TimeValue min = TimeValue.timeValueSeconds(postExpirySeconds - randomIntBetween(1, 3));

            final ExpirationCallback.Post post = new NoopPostExpirationCallback(min, null, timeValueMillis(10));
            long now = System.currentTimeMillis();
            assertThat(post.delay(now - postExpiryDuration.millis(), now), equalTo(TimeValue.timeValueMillis(0)));
        }

        public void testPreExpirationWithNullMin() throws Exception {
            int expirySeconds = randomIntBetween(5, 10);
            TimeValue expiryDuration = TimeValue.timeValueSeconds(expirySeconds);
            TimeValue max = TimeValue.timeValueSeconds(expirySeconds + randomIntBetween(1, 10));

            final ExpirationCallback.Pre pre = new NoopPreExpirationCallback(null, max, timeValueMillis(10));
            long now = System.currentTimeMillis();
            assertThat(pre.delay(expiryDuration.millis() + now, now), equalTo(TimeValue.timeValueMillis(0)));
        }

        public void testPreExpirationScheduleTime() throws Exception {
            TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
            TimeValue min = TimeValue.timeValueSeconds(1);
            TimeValue max = TimeValue.timeValueSeconds(4);
            TimeValue frequency = TimeValue.timeValueSeconds(1);
            NoopPreExpirationCallback pre = new NoopPreExpirationCallback(min, max, frequency);
            int latestValidTriggerDelay = (int) (expiryDuration.getMillis() - min.getMillis());
            int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() - max.getMillis());
            assertExpirationCallbackScheduleTime(pre, expiryDuration.millis(), latestValidTriggerDelay,
    earliestValidTriggerDelay);
        }

        public void testPostExpirationScheduleTime() throws Exception {
            TimeValue expiryDuration = TimeValue.timeValueSeconds(randomIntBetween(5, 10));
            TimeValue min = TimeValue.timeValueSeconds(1);
            TimeValue max = TimeValue.timeValueSeconds(4);
            TimeValue frequency = TimeValue.timeValueSeconds(1);
            NoopPostExpirationCallback pre = new NoopPostExpirationCallback(min, max, frequency);
            int latestValidTriggerDelay = (int) (expiryDuration.getMillis() + max.getMillis());
            int earliestValidTriggerDelay = (int) (expiryDuration.getMillis() + min.getMillis());
            assertExpirationCallbackScheduleTime(pre, expiryDuration.millis(), latestValidTriggerDelay,
    earliestValidTriggerDelay);
        }

        private void assertExpirationCallbackDelay(ExpirationCallback expirationCallback, long expiryDuration,
                            int latestValidTriggerDelay, int earliestValidTriggerDelay) {
```

```java
        long now = System.currentTimeMillis();
        long expiryDate = now + expiryDuration;
        // bounds
        assertThat(expirationCallback.delay(expiryDate, now + earliestValidTriggerDelay),
    equalTo(TimeValue.timeValueMillis(0)));
        assertThat(expirationCallback.delay(expiryDate, now + latestValidTriggerDelay),
    equalTo(TimeValue.timeValueMillis(0)));
        // in match
        assertThat(expirationCallback.delay(expiryDate,
            now + randomIntBetween(earliestValidTriggerDelay, latestValidTriggerDelay)),
            equalTo(TimeValue.timeValueMillis(0)));
        // out of bounds
        int deltaBeforeEarliestMatch = between(1, earliestValidTriggerDelay);
        assertThat(expirationCallback.delay(expiryDate, now + deltaBeforeEarliestMatch),
            equalTo(TimeValue.timeValueMillis(earliestValidTriggerDelay - deltaBeforeEarliestMatch)));
        int deltaAfterLatestMatch = between(latestValidTriggerDelay + 1, Integer.MAX_VALUE); // after expiry and
    after max
        assertThat(expirationCallback.delay(expiryDate, expiryDate + deltaAfterLatestMatch), nullValue());
    }

    public void assertExpirationCallbackScheduleTime(ExpirationCallback expirationCallback, long expiryDuration,
                                    int latestValidTriggerDelay, int earliestValidTriggerDelay) {
        long now = System.currentTimeMillis();
        long expiryDate = now + expiryDuration;
        int validTriggerInterval = between(earliestValidTriggerDelay, latestValidTriggerDelay);
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate,
            now + validTriggerInterval, now + validTriggerInterval),
            equalTo(now + validTriggerInterval));
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate, now, now + validTriggerInterval),
            equalTo(now + validTriggerInterval + expirationCallback.getFrequency()));

        int deltaBeforeEarliestMatch = between(1, earliestValidTriggerDelay - 1);
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate, now, now +
    deltaBeforeEarliestMatch),
            equalTo(now + deltaBeforeEarliestMatch +
                expirationCallback.delay(expiryDate, now + deltaBeforeEarliestMatch).getMillis()));
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate,
            now + deltaBeforeEarliestMatch, now + deltaBeforeEarliestMatch),
            equalTo(now + deltaBeforeEarliestMatch +
                expirationCallback.delay(expiryDate, now + deltaBeforeEarliestMatch).getMillis()));

        int deltaAfterLatestMatch = between(latestValidTriggerDelay + 1, Integer.MAX_VALUE); // after expiry and
    after max
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate, now, now + deltaAfterLatestMatch),
    equalTo(-1L));
        assertThat(expirationCallback.nextScheduledTimeForExpiry(expiryDate,
            now + deltaAfterLatestMatch, now + deltaAfterLatestMatch),
            equalTo(-1L));
```

```java
    }

    private static class NoopPostExpirationCallback extends ExpirationCallback.Post {

        NoopPostExpirationCallback(TimeValue min, TimeValue max, TimeValue frequency) {
            super(min, max, frequency);
        }

        @Override
        public void on(License license) { }
    }

    private static class NoopPreExpirationCallback extends ExpirationCallback.Pre {

        NoopPreExpirationCallback(TimeValue min, TimeValue max, TimeValue frequency) {
            super(min, max, frequency);
        }

        @Override
        public void on(License license) { }
    }
}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.action.support.PlainActionFuture;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.transport.TransportAddress;
import org.elasticsearch.common.unit.TimeValue;

import java.net.InetAddress;

import static java.util.Collections.emptyMap;
import static java.util.Collections.emptySet;
import static org.hamcrest.Matchers.containsString;
import static org.mockito.Matchers.any;
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;

public class LicenseTLSTests extends AbstractLicenseServiceTestCase {
```

```java
    private InetAddress inetAddress;

  public void testApplyLicenseInDevMode() throws Exception {
      License newLicense = TestUtils.generateSignedLicense(randomFrom("gold", "platinum"),
TimeValue.timeValueHours(24L));
      PutLicenseRequest request = new PutLicenseRequest();
      request.acknowledge(true);
      request.license(newLicense);
      Settings settings = Settings.builder().put("xpack.security.enabled", true).build();
      XPackLicenseState licenseState = new XPackLicenseState(settings);
      inetAddress = InetAddress.getLoopbackAddress();

      setInitialState(null, licenseState, settings);
      licenseService.start();
      PlainActionFuture<PutLicenseResponse> responseFuture = new PlainActionFuture<>();
      licenseService.registerLicense(request, responseFuture);
      verify(clusterService).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

      inetAddress = TransportAddress.META_ADDRESS;
      settings = Settings.builder()
          .put("xpack.security.enabled", true)
          .put("discovery.type", "single-node")
          .build();
      licenseService.stop();
      licenseState = new XPackLicenseState(settings);
      setInitialState(null, licenseState, settings);
      licenseService.start();
      licenseService.registerLicense(request, responseFuture);
      verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
  }

  public void testApplyLicenseInProdMode() throws Exception {
      final String licenseType = randomFrom("GOLD", "PLATINUM");
      License newLicense = TestUtils.generateSignedLicense(licenseType, TimeValue.timeValueHours(24L));
      PutLicenseRequest request = new PutLicenseRequest();
      request.acknowledge(true);
      request.license(newLicense);
      Settings settings = Settings.builder().put("xpack.security.enabled", true).build();
      XPackLicenseState licenseState = new XPackLicenseState(settings);
      inetAddress = TransportAddress.META_ADDRESS;

      setInitialState(null, licenseState, settings);
      licenseService.start();
      PlainActionFuture<PutLicenseResponse> responseFuture = new PlainActionFuture<>();
      IllegalStateException e = expectThrows(IllegalStateException.class, () ->
licenseService.registerLicense(request, responseFuture));
      assertThat(e.getMessage(),
          containsString("Cannot install a [" + licenseType + "] license unless TLS is configured or security is
```

```
disabled"));

        settings = Settings.builder().put("xpack.security.enabled", false).build();
        licenseService.stop();
        licenseState = new XPackLicenseState(settings);
        setInitialState(null, licenseState, settings);
        licenseService.start();
        licenseService.registerLicense(request, responseFuture);
        verify(clusterService).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

        settings = Settings.builder()
            .put("xpack.security.enabled", true)
            .put("xpack.security.transport.ssl.enabled", true)
            .build();
        licenseService.stop();
        licenseState = new XPackLicenseState(settings);
        setInitialState(null, licenseState, settings);
        licenseService.start();
        licenseService.registerLicense(request, responseFuture);
        verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
    }

    @Override
    protected DiscoveryNode getLocalNode() {
        return new DiscoveryNode("localnode", new TransportAddress(inetAddress, randomIntBetween(9300, 9399)),
            emptyMap(), emptySet(), Version.CURRENT);
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionFuture;
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.node.Node;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.test.ESSingleNodeTestCase;
import org.elasticsearch.xpack.core.LocalStateCompositeXPackPlugin;
import org.elasticsearch.xpack.core.XPackSettings;

import java.nio.charset.StandardCharsets;
import java.util.Collection;
```

```
import java.util.Collections;
import java.util.UUID;

import static org.elasticsearch.license.TestUtils.dateMath;
import static org.elasticsearch.license.TestUtils.generateExpiredNonBasicLicense;
import static org.elasticsearch.license.TestUtils.generateSignedLicense;
import static org.hamcrest.CoreMatchers.equalTo;
import static org.hamcrest.CoreMatchers.not;

public class LicensesTransportTests extends ESSingleNodeTestCase {

    @Override
    protected boolean resetNodeAfterTest() {
        return true;
    }

    @Override
    protected Collection<Class<? extends Plugin>> getPlugins() {
        return Collections.singletonList(LocalStateCompositeXPackPlugin.class);
    }

    @Override
    protected Settings nodeSettings() {
        Settings.Builder newSettings = Settings.builder();
        newSettings.put(super.nodeSettings());
        newSettings.put(XPackSettings.SECURITY_ENABLED.getKey(), false);
//        newSettings.put(XPackSettings.MONITORING_ENABLED.getKey(), false);
//        newSettings.put(XPackSettings.WATCHER_ENABLED.getKey(), false);
        newSettings.put(Node.NODE_DATA_SETTING.getKey(), true);
        return newSettings.build();
    }

    public void testEmptyGetLicense() throws Exception {
        // basic license is added async, we should wait for it
        assertBusy(() -> {
            try {
                final ActionFuture<GetLicenseResponse> getLicenseFuture =
                        new GetLicenseRequestBuilder(client().admin().cluster(), GetLicenseAction.INSTANCE).execute();
                final GetLicenseResponse getLicenseResponse;
                getLicenseResponse = getLicenseFuture.get();
                assertNotNull(getLicenseResponse.license());
                assertThat(getLicenseResponse.license().operationMode(), equalTo(License.OperationMode.BASIC));
            } catch (Exception e) {
                throw new RuntimeException("unexpected exception", e);
            }
        });
    }
```

```java
public void testPutLicense() throws Exception {
    License signedLicense = generateSignedLicense(TimeValue.timeValueMinutes(2));

    // put license
    PutLicenseRequestBuilder putLicenseRequestBuilder =
        new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(signedLicense)
            .setAcknowledge(true);
    PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
    assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));

    // get and check license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), equalTo(signedLicense));
}

public void testPutLicenseFromString() throws Exception {
    License signedLicense = generateSignedLicense(TimeValue.timeValueMinutes(2));
    String licenseString = TestUtils.dumpLicense(signedLicense);

    // put license source
    PutLicenseRequestBuilder putLicenseRequestBuilder =
        new PutLicenseRequestBuilder(client().admin().cluster(), PutLicenseAction.INSTANCE)
            .setLicense(new BytesArray(licenseString.getBytes(StandardCharsets.UTF_8)),
XContentType.JSON)
            .setAcknowledge(true);
    PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
    assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));

    // get and check license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), equalTo(signedLicense));
}

public void testPutInvalidLicense() throws Exception {
    License signedLicense = generateSignedLicense(TimeValue.timeValueMinutes(2));

    // modify content of signed license
    License tamperedLicense = License.builder()
        .fromLicenseSpec(signedLicense, signedLicense.signature())
        .expiryDate(signedLicense.expiryDate() + 10 * 24 * 60 * 60 * 1000L)
        .validate()
        .build();
```

```java
    PutLicenseRequestBuilder builder = new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE);
    builder.setLicense(tamperedLicense);

    // try to put license (should be invalid)
    final PutLicenseResponse putLicenseResponse = builder.get();
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.INVALID));

    // try to get invalid license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), not(tamperedLicense));
  }

  public void testPutBasicLicenseIsInvalid() throws Exception {
    License signedLicense = generateSignedLicense("basic", License.VERSION_CURRENT, -1,
TimeValue.timeValueMinutes(2));

    PutLicenseRequestBuilder builder = new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE);
    builder.setLicense(signedLicense);

    // try to put license (should be invalid)
    IllegalArgumentException iae = expectThrows(IllegalArgumentException.class, builder::get);
    assertEquals(iae.getMessage(), "Registering basic licenses is not allowed.");

    // try to get invalid license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), not(signedLicense));
  }

  public void testPutExpiredLicense() throws Exception {
    License expiredLicense = generateExpiredNonBasicLicense();
    PutLicenseRequestBuilder builder = new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE);
    builder.setLicense(expiredLicense);
    PutLicenseResponse putLicenseResponse = builder.get();
    assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.EXPIRED));
    // get license should not return the expired license
    GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
GetLicenseAction.INSTANCE).get();
    assertThat(getLicenseResponse.license(), not(expiredLicense));
  }

  public void testPutLicensesSimple() throws Exception {
    License goldSignedLicense = generateSignedLicense("gold", TimeValue.timeValueMinutes(5));
    PutLicenseRequestBuilder putLicenseRequestBuilder =
```

```java
            new PutLicenseRequestBuilder(client().admin().cluster(),
    PutLicenseAction.INSTANCE).setLicense(goldSignedLicense)
                    .setAcknowledge(true);
        PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
        GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
    GetLicenseAction.INSTANCE).get();
        assertThat(getLicenseResponse.license(), equalTo(goldSignedLicense));

        License platinumSignedLicense = generateSignedLicense("platinum", TimeValue.timeValueMinutes(2));
        putLicenseRequestBuilder.setLicense(platinumSignedLicense);
        putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
        getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
    GetLicenseAction.INSTANCE).get();
        assertThat(getLicenseResponse.license(), equalTo(platinumSignedLicense));
    }

    public void testRemoveLicensesSimple() throws Exception {
        License goldLicense = generateSignedLicense("gold", TimeValue.timeValueMinutes(5));
        PutLicenseRequestBuilder putLicenseRequestBuilder =
            new PutLicenseRequestBuilder(client().admin().cluster(),
    PutLicenseAction.INSTANCE).setLicense(goldLicense)
                    .setAcknowledge(true);
        PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
        GetLicenseResponse getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
    GetLicenseAction.INSTANCE).get();
        assertThat(getLicenseResponse.license(), equalTo(goldLicense));
        // delete all licenses
        DeleteLicenseRequestBuilder deleteLicenseRequestBuilder =
            new DeleteLicenseRequestBuilder(client().admin().cluster(), DeleteLicenseAction.INSTANCE);
        DeleteLicenseResponse deleteLicenseResponse = deleteLicenseRequestBuilder.get();
        assertThat(deleteLicenseResponse.isAcknowledged(), equalTo(true));
        // get licenses (expected no licenses)
        getLicenseResponse = new GetLicenseRequestBuilder(client().admin().cluster(),
    GetLicenseAction.INSTANCE).get();
        assertNull(getLicenseResponse.license());
    }

    public void testLicenseIsRejectWhenStartDateLaterThanNow() throws Exception {
        long now = System.currentTimeMillis();
        final License.Builder builder = License.builder()
            .uid(UUID.randomUUID().toString())
            .version(License.VERSION_CURRENT)
            .expiryDate(dateMath("now+2h", now))
```

```
                .startDate(dateMath("now+1h", now))
                .issueDate(now)
                .type(License.OperationMode.TRIAL.toString())
                .issuedTo("customer")
                .issuer("elasticsearch")
                .maxNodes(5);
        License license = TestUtils.generateSignedLicense(builder);

        PutLicenseRequestBuilder putLicenseRequestBuilder =
                new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(license)
                    .setAcknowledge(true);
        PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.INVALID));
    }

    public void testLicenseIsAcceptedWhenStartDateBeforeThanNow() throws Exception {
        long now = System.currentTimeMillis();
        final License.Builder builder = License.builder()
                .uid(UUID.randomUUID().toString())
                .version(License.VERSION_CURRENT)
                .expiryDate(dateMath("now+2h", now))
                .startDate(now)
                .issueDate(now)
                .type(License.OperationMode.TRIAL.toString())
                .issuedTo("customer")
                .issuer("elasticsearch")
                .maxNodes(5);
        License license = TestUtils.generateSignedLicense(builder);

        PutLicenseRequestBuilder putLicenseRequestBuilder =
                new PutLicenseRequestBuilder(client().admin().cluster(),
PutLicenseAction.INSTANCE).setLicense(license)
                    .setAcknowledge(true);
        PutLicenseResponse putLicenseResponse = putLicenseRequestBuilder.get();
        assertThat(putLicenseResponse.isAcknowledged(), equalTo(true));
        assertThat(putLicenseResponse.status(), equalTo(LicensesStatus.VALID));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.action.ActionListener;
```

```java
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;

import static org.elasticsearch.common.unit.TimeValue.timeValueHours;
import static org.hamcrest.Matchers.equalTo;
import static org.hamcrest.Matchers.not;
import static org.mockito.Matchers.any;
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;

public class LicensesAcknowledgementTests extends AbstractLicenseServiceTestCase {

    public void testAcknowledgment() throws Exception {
        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(TestUtils.generateSignedLicense("gold", timeValueHours(2)), licenseState, Settings.EMPTY);
        licenseService.start();
        // try installing a signed license
        long issueDate = System.currentTimeMillis() - TimeValue.timeValueHours(24 * 2).getMillis();
        License signedLicense = TestUtils.generateSignedLicense("trial", License.VERSION_CURRENT, issueDate,
timeValueHours(10));
        PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
        // ensure acknowledgement message was part of the response
        licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true));
        assertThat(licenseService.getLicense(), not(signedLicense));
        verify(clusterService, times(0)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

        // try installing a signed license with acknowledgement
        putLicenseRequest = new PutLicenseRequest().license(signedLicense).acknowledge(true);
        // ensure license was installed and no acknowledgment message was returned
        licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(true,
LicensesStatus.VALID, false));
        verify(clusterService, times(1)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
    }

    public void testRejectUpgradeToProductionWithoutTLS() throws Exception {
        XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
        setInitialState(TestUtils.generateSignedLicense("trial", timeValueHours(2)), licenseState, Settings.EMPTY);
        licenseService.start();
        // try installing a signed license
        License signedLicense = TestUtils.generateSignedLicense("platinum", timeValueHours(10));
        PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
        // ensure acknowledgement message was part of the response
        IllegalStateException ise = expectThrows(IllegalStateException.class, () ->
                licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true)));
        assertEquals("Cannot install a [PLATINUM] license unless TLS is configured or security is disabled",
```

```
      ise.getMessage());
   }

   public void testUpgradeToProductionWithoutTLSAndSecurityDisabled() throws Exception {
      XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
      setInitialState(TestUtils.generateSignedLicense("trial", timeValueHours(2)), licenseState, Settings.builder()
            .put("xpack.security.enabled", false).build());
      licenseService.start();
      // try installing a signed license
      License signedLicense = TestUtils.generateSignedLicense("platinum", timeValueHours(10));
      PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
      licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true));
      assertThat(licenseService.getLicense(), not(signedLicense));
      verify(clusterService, times(1)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

      // try installing a signed license with acknowledgement
      putLicenseRequest = new PutLicenseRequest().license(signedLicense).acknowledge(true);
      // ensure license was installed and no acknowledgment message was returned
      licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(true,
LicensesStatus.VALID, false));
      verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
   }

   public void testUpgradeToProductionWithTLSAndSecurity() throws Exception {
      XPackLicenseState licenseState = new XPackLicenseState(Settings.EMPTY);
      setInitialState(TestUtils.generateSignedLicense("trial", timeValueHours(2)), licenseState, Settings.builder()
            .put("xpack.security.enabled", true)
            .put("xpack.security.transport.ssl.enabled", true).build());
      licenseService.start();
      // try installing a signed license
      License signedLicense = TestUtils.generateSignedLicense("platinum", timeValueHours(10));
      PutLicenseRequest putLicenseRequest = new PutLicenseRequest().license(signedLicense);
      licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(false,
LicensesStatus.VALID, true));
      assertThat(licenseService.getLicense(), not(signedLicense));
      verify(clusterService, times(1)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));

      // try installing a signed license with acknowledgement
      putLicenseRequest = new PutLicenseRequest().license(signedLicense).acknowledge(true);
      // ensure license was installed and no acknowledgment message was returned
      licenseService.registerLicense(putLicenseRequest, new AssertingLicensesUpdateResponse(true,
LicensesStatus.VALID, false));
      verify(clusterService, times(2)).submitStateUpdateTask(any(String.class), any(ClusterStateUpdateTask.class));
   }

   private static class AssertingLicensesUpdateResponse implements ActionListener<PutLicenseResponse> {
      private final boolean expectedAcknowledgement;
```

```java
    private final LicensesStatus expectedStatus;
    private final boolean expectAckMessages;

    AssertingLicensesUpdateResponse(boolean expectedAcknowledgement, LicensesStatus expectedStatus,
                                    boolean expectAckMessages) {
        this.expectedAcknowledgement = expectedAcknowledgement;
        this.expectedStatus = expectedStatus;
        this.expectAckMessages = expectAckMessages;
    }

    @Override
    public void onResponse(PutLicenseResponse licensesUpdateResponse) {
        assertThat(licensesUpdateResponse.isAcknowledged(), equalTo(expectedAcknowledgement));
        assertThat(licensesUpdateResponse.status(), equalTo(expectedStatus));
        assertEquals(licensesUpdateResponse.acknowledgeMessages().isEmpty(), expectAckMessages == false);
    }

    @Override
    public void onFailure(Exception throwable) {
        throw new RuntimeException(throwable);
    }
  }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterName;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.block.ClusterBlocks;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.cluster.service.ClusterService;
import org.elasticsearch.common.component.Lifecycle;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.env.Environment;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.watcher.ResourceWatcherService;
import org.elasticsearch.xpack.core.XPackPlugin;
import org.elasticsearch.xpack.core.watcher.watch.ClockMock;
import org.junit.After;
import org.junit.Before;
```

```java
import java.nio.file.Path;
import java.util.Arrays;

import static java.util.Collections.emptySet;
import static java.util.Collections.singletonMap;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.when;

public abstract class AbstractLicenseServiceTestCase extends ESTestCase {

    protected LicenseService licenseService;
    protected ClusterService clusterService;
    protected ResourceWatcherService resourceWatcherService;
    protected ClockMock clock;
    protected DiscoveryNodes discoveryNodes;
    protected Environment environment;
    protected String licenseType;

    @Before
    public void init() throws Exception {
        clusterService = mock(ClusterService.class);
        clock = ClockMock.frozen();
        discoveryNodes = mock(DiscoveryNodes.class);
        resourceWatcherService = mock(ResourceWatcherService.class);
        environment = mock(Environment.class);
    }

    protected void setInitialState(License license, XPackLicenseState licenseState, Settings settings) {
        setInitialState(license, licenseState, settings, randomBoolean() ? "trial" : "basic");
    }

    protected void setInitialState(License license, XPackLicenseState licenseState, Settings settings, String
selfGeneratedType) {
        Path tempDir = createTempDir();
        when(environment.configFile()).thenReturn(tempDir);
        licenseType = selfGeneratedType;
        settings =
Settings.builder().put(settings).put(LicenseService.SELF_GENERATED_LICENSE_TYPE.getKey(),
licenseType).build();
        licenseService = new LicenseService(settings, clusterService, clock, environment, resourceWatcherService,
licenseState);
        ClusterState state = mock(ClusterState.class);
        final ClusterBlocks noBlock = ClusterBlocks.builder().build();
        when(state.blocks()).thenReturn(noBlock);
        MetaData metaData = mock(MetaData.class);
        when(metaData.custom(LicensesMetaData.TYPE)).thenReturn(new LicensesMetaData(license, null));
        when(state.metaData()).thenReturn(metaData);
        final DiscoveryNode mockNode = getLocalNode();
```

```java
        when(discoveryNodes.getMasterNode()).thenReturn(mockNode);
        when(discoveryNodes.spliterator()).thenReturn(Arrays.asList(mockNode).spliterator());
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(false);
        when(state.nodes()).thenReturn(discoveryNodes);
        when(state.getNodes()).thenReturn(discoveryNodes); // it is really ridiculous we have nodes() and getNodes()...
        when(clusterService.state()).thenReturn(state);
        when(clusterService.lifecycleState()).thenReturn(Lifecycle.State.STARTED);
        when(clusterService.getClusterName()).thenReturn(new ClusterName("a"));
        when(clusterService.localNode()).thenReturn(mockNode);
    }

    protected DiscoveryNode getLocalNode() {
        return new DiscoveryNode("b", buildNewFakeTransportAddress(),
singletonMap(XPackPlugin.XPACK_INSTALLED_NODE_ATTR, "true"),
            emptySet(), Version.CURRENT);
    }

    @After
    public void tearDown() throws Exception {
        super.tearDown();
        licenseService.stop();
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.Version;
import org.elasticsearch.cluster.ClusterChangedEvent;
import org.elasticsearch.cluster.ClusterName;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.cluster.ClusterStateUpdateTask;
import org.elasticsearch.cluster.metadata.MetaData;
import org.elasticsearch.cluster.node.DiscoveryNode;
import org.elasticsearch.cluster.node.DiscoveryNodes;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.junit.After;
import org.junit.Before;
import org.mockito.ArgumentCaptor;

import static java.util.Collections.emptyMap;
import static java.util.Collections.emptySet;
import static org.hamcrest.Matchers.equalTo;
import static org.mockito.Matchers.any;
```

```java
import static org.mockito.Mockito.times;
import static org.mockito.Mockito.verify;
import static org.mockito.Mockito.when;

public class LicenseClusterChangeTests extends AbstractLicenseServiceTestCase {

    private TestUtils.AssertingLicenseState licenseState;

    @Before
    public void setup() {
        licenseState = new TestUtils.AssertingLicenseState();
        setInitialState(null, licenseState, Settings.EMPTY);
        licenseService.start();
    }

    @After
    public void teardown() {
        licenseService.stop();
    }


    public void testNotificationOnNewLicense() throws Exception {
        ClusterState oldState = ClusterState.builder(new ClusterName("a")).build();
        final License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(24));
        MetaData metaData = MetaData.builder().putCustom(LicensesMetaData.TYPE, new
LicensesMetaData(license, null)).build();
        ClusterState newState = ClusterState.builder(new ClusterName("a")).metaData(metaData).build();
        licenseService.clusterChanged(new ClusterChangedEvent("simulated", newState, oldState));
        assertThat(licenseState.activeUpdates.size(), equalTo(1));
        assertTrue(licenseState.activeUpdates.get(0));
    }

    public void testNoNotificationOnExistingLicense() throws Exception {
        final License license = TestUtils.generateSignedLicense(TimeValue.timeValueHours(24));
        MetaData metaData = MetaData.builder().putCustom(LicensesMetaData.TYPE, new
LicensesMetaData(license, null)).build();
        ClusterState newState = ClusterState.builder(new ClusterName("a")).metaData(metaData).build();
        ClusterState oldState = ClusterState.builder(newState).build();
        licenseService.clusterChanged(new ClusterChangedEvent("simulated", newState, oldState));
        assertThat(licenseState.activeUpdates.size(), equalTo(0));
    }

    public void testSelfGeneratedLicenseGeneration() throws Exception {
        DiscoveryNode master = new DiscoveryNode("b", buildNewFakeTransportAddress(), emptyMap(), emptySet(),
Version.CURRENT);
        ClusterState oldState = ClusterState.builder(new ClusterName("a"))
                .nodes(DiscoveryNodes.builder().masterNodeId(master.getId()).add(master)).build();
        when(discoveryNodes.isLocalNodeElectedMaster()).thenReturn(true);
```

```
    ClusterState newState = ClusterState.builder(oldState).nodes(discoveryNodes).build();


    licenseService.clusterChanged(new ClusterChangedEvent("simulated", newState, oldState));
    ArgumentCaptor<ClusterStateUpdateTask> stateUpdater =
ArgumentCaptor.forClass(ClusterStateUpdateTask.class);
    verify(clusterService, times(1)).submitStateUpdateTask(any(), stateUpdater.capture());
    ClusterState stateWithLicense = stateUpdater.getValue().execute(newState);
    LicensesMetaData licenseMetaData = stateWithLicense.metaData().custom(LicensesMetaData.TYPE);
    assertNotNull(licenseMetaData);
    assertNotNull(licenseMetaData.getLicense());
    assertEquals(licenseType, licenseMetaData.getLicense().type());
    long expiration;
    if (licenseType.equals("basic")) {
        expiration = LicenseService.BASIC_SELF_GENERATED_LICENSE_EXPIRATION_MILLIS;
    } else {
        expiration = LicenseService.NON_BASIC_SELF_GENERATED_LICENSE_DURATION.millis() +
clock.millis();
    }
    assertEquals(expiration, licenseMetaData.getLicense().expiryDate());
  }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.xpack.core.security.authz.permission;

import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.test.ESTestCase;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

public class FieldPermissionsCacheTests extends ESTestCase {

  public void testFieldPermissionsCaching() {
    FieldPermissionsCache fieldPermissionsCache = new FieldPermissionsCache(Settings.EMPTY);
    String[] allowed = new String[]{randomAlphaOfLength(5) + "*", randomAlphaOfLength(5) + "*",
randomAlphaOfLength(5) + "*"};
    String[] denied = new String[]{allowed[0] + randomAlphaOfLength(5), allowed[1] +
randomAlphaOfLength(5),
          allowed[2] + randomAlphaOfLength(5)};
    FieldPermissions fieldPermissions = fieldPermissionsCache.getFieldPermissions(allowed, denied);
    assertNotNull(fieldPermissions);
    final String[] allowed2 = randomBoolean() ? allowed : Arrays.copyOf(allowed, allowed.length);
```

```java
        final String[] denied2 = randomBoolean() ? denied : Arrays.copyOf(denied, denied.length);
        assertSame(fieldPermissions, fieldPermissionsCache.getFieldPermissions(allowed2, denied2));
    }

    public void testMergeFieldPermissions() {
        FieldPermissionsCache fieldPermissionsCache = new FieldPermissionsCache(Settings.EMPTY);
        String allowedPrefix1 = randomAlphaOfLength(5);
        String allowedPrefix2 = randomAlphaOfLength(5);
        String[] allowed1 = new String[]{allowedPrefix1 + "*"};
        String[] allowed2 = new String[]{allowedPrefix2 + "*"};
        String[] denied1 = new String[]{allowedPrefix1 + "a"};
        String[] denied2 = new String[]{allowedPrefix2 + "a"};
        FieldPermissions fieldPermissions1 = randomBoolean() ?
fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
                new FieldPermissions(fieldPermissionDef(allowed1, denied1));
        FieldPermissions fieldPermissions2 = randomBoolean() ?
fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
                new FieldPermissions(fieldPermissionDef(allowed2, denied2));
        FieldPermissions mergedFieldPermissions =
                fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
        assertTrue(mergedFieldPermissions.grantsAccessTo(allowedPrefix1 + "b"));
        assertTrue(mergedFieldPermissions.grantsAccessTo(allowedPrefix2 + "b"));
        assertFalse(mergedFieldPermissions.grantsAccessTo(denied1[0]));
        assertFalse(mergedFieldPermissions.grantsAccessTo(denied2[0]));

        allowed1 = new String[]{randomAlphaOfLength(5) + "*", randomAlphaOfLength(5) + "*"};
        allowed2 = null;
        denied1 = new String[]{allowed1[0] + "a", allowed1[1] + "a"};
        denied2 = null;
        fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
                new FieldPermissions(fieldPermissionDef(allowed1, denied1));
        fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
                new FieldPermissions(fieldPermissionDef(allowed2, denied2));
        mergedFieldPermissions =
                fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
        assertFalse(mergedFieldPermissions.hasFieldLevelSecurity());

        allowed1 = new String[]{};
        allowed2 = new String[]{randomAlphaOfLength(5) + "*", randomAlphaOfLength(5) + "*"};
        denied1 = new String[]{};
        denied2 = new String[]{allowed2[0] + "a", allowed2[1] + "a"};
        fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
                new FieldPermissions(fieldPermissionDef(allowed1, denied1));
        fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
                new FieldPermissions(fieldPermissionDef(allowed2, denied2));
        mergedFieldPermissions =
                fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
        for (String field : allowed2) {
```

```
            assertTrue(mergedFieldPermissions.grantsAccessTo(field));
        }
        for (String field : denied2) {
            assertFalse(mergedFieldPermissions.grantsAccessTo(field));
        }

        allowed1 = randomBoolean() ? null : new String[]{"*"};
        allowed2 = randomBoolean() ? null : new String[]{"*"};
        denied1 = new String[]{"a"};
        denied2 = new String[]{"b"};
        fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
            new FieldPermissions(fieldPermissionDef(allowed1, denied1));
        fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
            new FieldPermissions(fieldPermissionDef(allowed2, denied2));
        mergedFieldPermissions =
            fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
        assertTrue(mergedFieldPermissions.grantsAccessTo("a"));
        assertTrue(mergedFieldPermissions.grantsAccessTo("b"));

        allowed1 = new String[] { "a*" };
        allowed2 = new String[] { "b*" };
        denied1 = new String[] { "aa*" };
        denied2 = null;
        fieldPermissions1 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed1, denied1) :
            new FieldPermissions(fieldPermissionDef(allowed1, denied1));
        fieldPermissions2 = randomBoolean() ? fieldPermissionsCache.getFieldPermissions(allowed2, denied2) :
            new FieldPermissions(fieldPermissionDef(allowed2, denied2));
        mergedFieldPermissions =
            fieldPermissionsCache.getFieldPermissions(Arrays.asList(fieldPermissions1, fieldPermissions2));
        assertTrue(mergedFieldPermissions.grantsAccessTo("a"));
        assertTrue(mergedFieldPermissions.grantsAccessTo("b"));
        assertFalse(mergedFieldPermissions.grantsAccessTo("aa"));
        assertFalse(mergedFieldPermissions.grantsAccessTo("aa1"));
        assertTrue(mergedFieldPermissions.grantsAccessTo("a1"));
    }

    public void testNonFlsAndFlsMerging() {
        List<FieldPermissions> permissionsList = new ArrayList<>();
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {"field1"}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {"field2", "query*"}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {"field1", "field2"}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(new String[] {}, null)));
        permissionsList.add(new FieldPermissions(fieldPermissionDef(null, null)));

        FieldPermissionsCache cache = new FieldPermissionsCache(Settings.EMPTY);
        for (int i = 0; i < scaledRandomIntBetween(1, 12); i++) {
            Collections.shuffle(permissionsList, random());
            FieldPermissions result = cache.getFieldPermissions(permissionsList);
```

```
        assertFalse(result.hasFieldLevelSecurity());
      }
    }

    private static FieldPermissionsDefinition fieldPermissionDef(String[] granted, String[] denied) {
      return new FieldPermissionsDefinition(granted, denied);
    }
}
```

UnboundID LDAP SDK Free Use License

THIS IS AN AGREEMENT BETWEEN YOU ("YOU") AND UNBOUNDID CORP. ("UNBOUNDID")
REGARDING YOUR USE OF UNBOUNDID LDAP SDK FOR JAVA AND ANY ASSOCIATED
DOCUMENTATION, OBJECT CODE, COMPILED LIBRARIES, SOURCE CODE AND SOURCE FILES OR
OTHER MATERIALS MADE AVAILABLE BY UNBOUNDID (COLLECTIVELY REFERRED TO IN THIS
AGREEMENT AS THE ("SDK").

BY INSTALLING, ACCESSING OR OTHERWISE USING THE SDK, YOU ACCEPT THE TERMS OF
THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT
INSTALL, ACCESS OR USE THE SDK.

USE OF THE SDK. Subject to your compliance with this Agreement, UnboundID
grants to You a non-exclusive, royalty-free license, under UnboundID's
intellectual property rights in the SDK, to use, reproduce, modify and
distribute this release of the SDK; provided that no license is granted herein
under any patents that may be infringed by your modifications, derivative works
or by other works in which the SDK may be incorporated (collectively, your
"Applications"). You may reproduce and redistribute the SDK with your
Applications provided that you (i) include this license file and an
unmodified copy of the unboundid-ldapsdk-se.jar file; and (ii) such
redistribution is subject to a license whose terms do not conflict with or
contradict the terms of this Agreement. You may also reproduce and redistribute
the SDK without your Applications provided that you redistribute the SDK
complete and unmodified (i.e., with all "read me" files, copyright notices, and
other legal notices and terms that UnboundID has included in the SDK).

SCOPE OF LICENSES. This Agreement does not grant You the right to use any
UnboundID intellectual property which is not included as part of the SDK. The
SDK is licensed, not sold. This Agreement only gives You some rights to use
the SDK. UnboundID reserves all other rights. Unless applicable law gives You
more rights despite this limitation, You may use the SDK only as expressly
permitted in this Agreement.

SUPPORT. UnboundID is not obligated to provide any technical or other support
("Support Services") for the SDK to You under this Agreement. However, if
UnboundID chooses to provide any Support Services to You, Your use of such
Support Services will be governed by then-current UnboundID support policies.

TERMINATION. UnboundID reserves the right to discontinue offering the SDK and

to modify the SDK at any time in its sole discretion. Notwithstanding anything contained in this Agreement to the contrary, UnboundID may also, in its sole discretion, terminate or suspend access to the SDK to You or any end user at any time. In addition, if you fail to comply with the terms of this Agreement, then any rights granted herein will be automatically terminated if such failure is not corrected within 30 days of the initial notification of such failure. You acknowledge that termination and/or monetary damages may not be a sufficient remedy if You breach this Agreement and that UnboundID will be entitled, without waiving any other rights or remedies, to injunctive or equitable relief as may be deemed proper by a court of competent jurisdiction in the event of a breach. UnboundID may also terminate this Agreement if the SDK becomes, or in UnboundID?s reasonable opinion is likely to become, the subject of a claim of intellectual property infringement or trade secret misappropriation. All rights and licenses granted herein will simultaneously and automatically terminate upon termination of this Agreement for any reason.

DISCLAIMER OF WARRANTY. THE SDK IS PROVIDED "AS IS" AND UNBOUNDID DOES NOT WARRANT THAT THE SDK WILL BE ERROR-FREE, VIRUS-FREE, WILL PERFORM IN AN UNINTERRUPTED, SECURE OR TIMELY MANNER, OR WILL INTEROPERATE WITH OTHER HARDWARE, SOFTWARE, SYSTEMS OR DATA. TO THE MAXIMUM EXTENT ALLOWED BY LAW, ALL
CONDITIONS, REPRESENTATIONS AND WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (EVEN IF UNBOUNDID HAD BEEN INFORMED OF SUCH PURPOSE), OR NON-INFRINGEMENT OF THIRD PARTY RIGHTS ARE HEREBY DISCLAIMED.

LIMITATION OF LIABILITY. IN NO EVENT WILL UNBOUNDID OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, REVENUE, DATA OR DATA USE, BUSINESS INTERRUPTION, COST OF COVER, DIRECT, INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY KIND) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SDK OR IN ANY WAY RELATED TO THIS AGREEMENT, EVEN IF UNBOUNDID HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ADDITIONAL RIGHTS. Certain states do not allow the exclusion of implied warranties or limitation of liability for certain kinds of damages, so the exclusion of limited warranties and limitation of liability set forth above may not apply to You.

EXPORT RESTRICTIONS. The SDK is subject to United States export control laws. You acknowledge and agree that You are responsible for compliance with all domestic and international export laws and regulations that apply to the SDK.

MISCELLANEOUS. This Agreement constitutes the entire agreement with respect to the SDK. If any provision of this Agreement shall be held to be invalid, illegal or unenforceable, the validity, legality and enforceability of the remaining provisions shall in no way be affected or impaired thereby. This

Agreement and performance hereunder shall be governed by and construed in
accordance with the laws of the State of Texas without regard to its conflict
of laws rules. Any disputes related to this Agreement shall be exclusively
litigated in the state or federal courts located in Travis County, Texas.

http://www.opensource.org/licenses/bsd-license.php

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the following
conditions are met:

Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with
the distribution.

Neither the name of JLine nor the names of its contributors
may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
You may use under either the Apache License Version 2.0 or the BSD
3-Clause License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"

means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and

attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are

met:

1. Redistributions of source code must retain the above copyright

   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

   notice, this list of conditions and the following disclaimer in the

   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a)the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b)ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.
Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:
(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and
(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).
(c) The licenses granted in Sections2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.
(d) Notwithstanding Section2.1(b) above, no patent license is granted: (1)for code that You delete from the Original Software, or (2)for infringements caused by: (i)the modification of the Original Software, or (ii)the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1)Modifications made by that Contributor (or portions thereof); and (2)the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section2.2(b) above, no patent license is granted: (1)for any code that Contributor has deleted from the Contributor Version; (2)for infringements caused by: (i)third party modifications of Contributor Version, or (ii)the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3)under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone,

and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.
You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.
Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.
4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a)rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b)otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE

PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a commercial item, as that term is defined in 48C.F.R.2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48C.F.R.12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes,

any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
The GlassFish code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.1

1. Definitions.

   1.1. "Contributor" means each individual or entity that creates or
   contributes to the creation of Modifications.

   1.2. "Contributor Version" means the combination of the Original
   Software, prior Modifications used by a Contributor (if any), and
   the Modifications made by that particular Contributor.

   1.3. "Covered Software" means (a) the Original Software, or (b)
   Modifications, or (c) the combination of files containing Original
   Software with files containing Modifications, in each case including
   portions thereof.

   1.4. "Executable" means the Covered Software in any form other than

Source Code.

1.5. "Initial Developer" means the individual or entity that first
makes Original Software available under this License.

1.6. "Larger Work" means a work which combines Covered Software or
portions thereof with code not governed by the terms of this License.

1.7. "License" means this document.

1.8. "Licensable" means having the right to grant, to the maximum
extent possible, whether at the time of the initial grant or
subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means the Source Code and Executable form of
any of the following:

A. Any file that results from an addition to, deletion from or
modification of the contents of a file containing Original Software
or previous Modifications;

B. Any new file that contains any part of the Original Software or
previous Modification; or

C. Any new file that is contributed or otherwise made available
under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form
of computer software code that is originally released under this
License.

1.11. "Patent Claims" means any patent claim(s), now owned or
hereafter acquired, including without limitation, method, process,
and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software
code in which modifications are made and (b) associated
documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity
exercising rights under, and complying with all of the terms of,
this License. For legal entities, "You" includes any entity which
controls, is controlled by, or is under common control with You. For
purposes of this definition, "control" means (a) the power, direct
or indirect, to cause the direction or management of such entity,
whether by contract or otherwise, or (b) ownership of more than
fifty percent (50%) of the outstanding shares or beneficial
ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or

otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in
Source Code form that alters or restricts the applicable version of
this License or the recipients' rights hereunder. You may choose to
offer, and to charge a fee for, warranty, support, indemnity or
liability obligations to one or more recipients of Covered Software.
However, you may do so only on Your own behalf, and not on behalf of
the Initial Developer or any Contributor. You must make it
absolutely clear that any such warranty, support, indemnity or
liability obligation is offered by You alone, and You hereby agree
to indemnify the Initial Developer and every Contributor for any
liability incurred by the Initial Developer or such Contributor as a
result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under
the terms of this License or under the terms of a license of Your
choice, which may contain terms different from this License,
provided that You are in compliance with the terms of this License
and that the license for the Executable form does not attempt to
limit or alter the recipient's rights in the Source Code form from
the rights set forth in this License. If You distribute the Covered
Software in Executable form under a different license, You must make
it absolutely clear that any terms which differ from this License
are offered by You alone, not by the Initial Developer or
Contributor. You hereby agree to indemnify the Initial Developer and
every Contributor for any liability incurred by the Initial
Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with
other code not governed by the terms of this License and distribute
the Larger Work as a single product. In such a case, You must make
sure the requirements of this License are fulfilled for the Covered
Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or
new versions of this License from time to time. Each version will be
given a distinguishing version number. Except as provided in Section
4.3, no one other than the license steward has the right to modify
this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the
Covered Software available under the terms of the version of the
License under which You originally received the Covered Software. If
the Initial Developer includes a notice in the Original Software
prohibiting it from being distributed or otherwise made available
under any subsequent version of the License, You must distribute and
make the Covered Software available under the terms of the version
of the License under which You originally received the Covered
Software. Otherwise, You may also choose to use, distribute or
otherwise make the Covered Software available under the terms of any
subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new
license for Your Original Software, You may create and use a
modified version of this License if You: (a) rename the license and
remove any references to the name of the license steward (except to
note that the license differs from this License); and (b) otherwise
make it clear that the license contains terms which differ from this
License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS,
WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,
INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE
IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR
NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF
THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE
DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY
OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING,
REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN
ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS
AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate
automatically if You fail to comply with terms herein and fail to
cure such breach within 30 days of becoming aware of the breach.
Provisions which, by their nature, must remain in effect beyond the
termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding

declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

   The Covered Software is a "commercial item," as that term is defined
   in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer
   software" (as that term is defined at 48 C.F.R.
   252.227-7014(a)(1)) and "commercial computer software documentation"
   as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent
   with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4
   (June 1995), all U.S. Government End Users acquire Covered Software
   with only those rights set forth herein. This U.S. Government Rights
   clause is in lieu of, and supersedes, any other FAR, DFAR, or other
   clause or provision that addresses Government rights in computer
   software under this License.

9. MISCELLANEOUS.

   This License represents the complete agreement concerning subject
   matter hereof. If any provision of this License is held to be
   unenforceable, such provision shall be reformed only to the extent
   necessary to make it enforceable. This License shall be governed by
   the law of the jurisdiction specified in a notice contained within
   the Original Software (except to the extent applicable law, if any,
   provides otherwise), excluding such jurisdiction's conflict-of-law
   provisions. Any litigation relating to this License shall be subject
   to the jurisdiction of the courts located in the jurisdiction and
   venue specified in a notice contained within the Original Software,
   with the losing party responsible for costs, including, without
   limitation, court costs and reasonable attorneys' fees and expenses.
   The application of the United Nations Convention on Contracts for
   the International Sale of Goods is expressly excluded. Any law or
   regulation which provides that the language of a contract shall be
   construed against the drafter shall not apply to this License. You
   agree that You alone are responsible for compliance with the United
   States export administration regulations (and the export control
   laws and regulation of any other countries) when You use, distribute
   or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

   As between Initial Developer and the Contributors, each party is
   responsible for claims and damages arising, directly or indirectly,
   out of its utilization of rights under this License and You agree to
   work with Initial Developer and Contributors to distribute such
   responsibility on an equitable basis. Nothing herein is intended or
   shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

 The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor
Boston, MA 02110-1335
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code.

And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it,

under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are

prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those

countries, so that distribution is permitted only in or among countries
not thus excluded. In such case, this License incorporates the
limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new
versions of the General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Program does not specify a version
number of this License, you may choose any version ever published by the
Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the
author to ask for permission. For software which is copyrighted by the
Free Software Foundation, write to the Free Software Foundation; we
sometimes make exceptions for this. Our decision will be guided by the
two goals of preserving the free status of all derivatives of our free
software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND,
EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE
ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH
YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL
NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR
DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM
(INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED
INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF
THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR
OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    One line to give the program's name and a brief idea of what it does.
    Copyright (C) <year> <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful, but
    WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
    General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1335 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type
    `show w'. This is free software, and you are welcome to redistribute
    it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the
appropriate parts of the General Public License. Of course, the commands
you use may be called something other than `show w' and `show c'; they
could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

#

Certain source files distributed by Oracle America, Inc. and/or its affiliates are subject to the following clarification and special exception to the GPLv2, based on the GNU Project exception for its Classpath libraries, known as the GNU Classpath Exception, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the LICENSE file that accompanied this code."

You should also note that Oracle includes multiple, independent programs in this software package. Some of those programs are provided under licenses deemed incompatible with the GPLv2 by the Free Software Foundation and others. For example, the package includes programs licensed under the Apache License, Version 2.0. Such programs are licensed to you under their original licenses.

Oracle facilitates your further distribution of this package by adding the Classpath Exception to the necessary parts of its GPLv2 code, which permits you to use that code in combination with other independent modules not licensed under the GPLv2. However, note that this would not permit you to commingle code under an incompatible license with Oracle's GPLv2 licensed code by, for example, cutting and pasting such code into a file also containing Oracle's GPLv2 licensed code and then distributing the result. Additionally, if you were to remove the Classpath Exception from any of the files to which it applies and distribute the result, you would likely be required to license some or all of the other code in that distribution under the GPLv2 as well, and since the GPLv2 is incompatible with the license terms of some items included in the distribution by Oracle, removing the Classpath Exception could therefore effectively compromise your ability to further distribute the package.

Proceed with caution and we recommend that you obtain the advice of a

lawyer skilled in open source matters before removing the Classpath
Exception or making modifications to this package which may
subsequently be redistributed and/or involve the use of third party
software.

CLASSPATH EXCEPTION
Linking this library statically or dynamically with other modules is
making a combined work based on this library. Thus, the terms and
conditions of the GNU General Public License version 2 cover the whole
combination.

As a special exception, the copyright holders of this library give you
permission to link this library with independent modules to produce an
executable, regardless of the license terms of these independent
modules, and to copy and distribute the resulting executable under
terms of your choice, provided that you also meet, for each linked
independent module, the terms and conditions of the license of that
module. An independent module is a module which is not derived from or
based on this library. If you modify this library, you may extend this
exception to your version of the library, but you are not obligated to
do so. If you do not wish to do so, delete this exception statement
from your version.

```
{
 "xpack.license.get_basic_status": {
   "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
   "methods": ["GET"],
   "url": {
    "path": "/_xpack/license/basic_status",
    "paths": ["/_xpack/license/basic_status"],
    "parts" : {
    },
    "params": {
    }
   },
   "body": null
 }
}
{
 "xpack.license.get_trial_status": {
   "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
   "methods": ["GET"],
   "url": {
    "path": "/_xpack/license/trial_status",
    "paths": ["/_xpack/license/trial_status"],
    "parts" : {
    },
    "params": {
    }
```

```
      },
      "body": null
    }
  }
  {
   "xpack.license.delete": {
    "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
    "methods": ["DELETE"],
    "url": {
     "path": "/_xpack/license",
     "paths": ["/_xpack/license"],
     "parts" : {}
    },
    "body": null
   }
  }
  {
   "xpack.license.post_start_trial": {
    "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
    "methods": ["POST"],
    "url": {
     "path": "/_xpack/license/start_trial",
     "paths": ["/_xpack/license/start_trial"],
     "parts" : {
     },
     "params": {
      "type": {
       "type" : "string",
       "description" : "The type of trial license to generate (default: \"trial\")"
      },
      "acknowledge": {
       "type" : "boolean",
       "description" : "whether the user has acknowledged acknowledge messages (default: false)"
      }
     }
    },
    "body": null
   }
  }
  {
   "xpack.license.post_start_basic": {
    "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
    "methods": ["POST"],
    "url": {
     "path": "/_xpack/license/start_basic",
     "paths": ["/_xpack/license/start_basic"],
     "parts" : {
     },
```

```
    "params": {
     "acknowledge": {
      "type" : "boolean",
      "description" : "whether the user has acknowledged acknowledge messages (default: false)"
     }
    }
   },
   "body": null
 }
}
{
 "xpack.license.get": {
  "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
  "methods": ["GET"],
  "url": {
   "path": "/_xpack/license",
   "paths": ["/_xpack/license"],
   "parts" : {
   },
   "params": {
    "local": {
     "type" : "boolean",
     "description" : "Return local information, do not retrieve the state from master node (default: false)"
    }
   }
  },
  "body": null
 }
}
{
 "xpack.license.post": {
  "documentation": "https://www.elastic.co/guide/en/x-pack/current/license-management.html",
  "methods": ["PUT", "POST"],
  "url": {
   "path": "/_xpack/license",
   "paths": ["/_xpack/license"],
   "parts" : {
   },
   "params": {
    "acknowledge": {
     "type" : "boolean",
     "description" : "whether the user has acknowledged acknowledge messages (default: false)"
    }
   }
  },
  "body": {
   "description" : "licenses to be installed"
  }
```

```
    }
  }
---

teardown:
 - do:
    xpack.license.post:
     acknowledge: true
     body: |
```
```
       {"licenses":[{"uid":"3aa62ffe-36e1-4fad-bfdc-
9dff8301eb22","type":"trial","issue_date_in_millis":1523456691721,"expiry_date_in_millis":1838816691721,"max
_nodes":5,"issued_to":"customer","issuer":"elasticsearch","signature":"AAAABAAAAA2kWNcuc+DT0lrlmYZKA
AAAIAo5/x6hrsGh1GqqrJmy4qgmEC7gK0U4zQ6q5ZEMhm4jAAABAEn6fG9y2VxKBu2T3D5hffh56kzOQOD
COdhr0y2d17ZSIJMZRqO7ZywPCWNS1aR33GhfIHkTER0ysML0xMH/gXavhyRvMBndJj0UBKzuwpTawSlnx
YtcqN8mSBIvJC7Ki+uJ1SpAILC2ZP9fnkRlqwXqBlTwfYn7xnZgu9DKrOWru/ipTPObo7jcePl8VTK6nWFen7/h
CFDQTUFZ0jQvd+nq7A1PAcHGNxGfdbMVmAXCXgGWkRfT3clo9/vadgo+isNyh1sPq9mN7gwsvBAKtA1Frp
H2EXYYbfOsSpBvUmhYMgErLg1k3/CbS0pCWLKOaX1xTMayosdZOjagU3auZXY=","start_date_in_millis":-
1}]}
```
```
---

"Installing and getting license works":

 ## current license version
 - do:
    xpack.license.post:
     acknowledge: true
     body: |
```
```
       {"licenses":[{"uid":"894371dc-9t49-4997-93cb-
8o2e3r7fa6a8","type":"trial","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1916956799999,"max
_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"AAAAAgAAAA0FWh0T9njItjQ2qammAAABmC
9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhllTHNoN1N2MXM
vRFk4d3JTZEx3R3RRRZ0pzU3lobWJKZnQvSEFva0pplTHBkWkprprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095M
XRGN1lIZlpYYcVVTTnFrcTE2dzhjJZmZrdFQrN3JQeGwxb0U0MXZ0dDJHSERRiZTVLOHNzSDByWnpooZEphZH
BEZjUrTVBxRENNSXNwWWJjZ1laODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQa
VB2azhmUk9QVzhFeTFFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JSQkFLFLV2VXUmVHNDZ2R3o2VE1q
bVNQS2lxOHN5bUErZlNIWkdJWEtaU09wTTJENDVvT1NCYklacUYyK2xwRW9Wxa0t6dldMbmMzSGtQ
c3FWOTgzZZ3ZUcXMvQkt2RUZwMFJnZzlvLmd2bDRWUzh6UG5dENGWFReXNKNkE9PQAAAQBZhvozA0
trrxhUZ1QbaTsKTna9C5KVQ6pv8yg1pnsBpZXCl8kX1SrgoFn1bXq61IvJwfw5qnmYNiH3hRhTO9EyaCBqaLk8
NXZQ6TrRkQSpEnnBwAYUkZeKXsIuBoOk4B4mzwC/r8aMAkzrTiEBtBbog+57cSaU9y37Gkdd+1jXCQrxP+jO
EUf7gnXWZvE6oeRroLvCt1fYn09k0CF8kKTbrPTSjC6igZR3uvTHyee74XQ9PRavvHax73T4UOEdQZX/P1ibS
QIWKbBRD5YQ1POYVjTayoltTnWLMxfEcAkkATJZLhpBEHST7kZWjrTS6J1dCReJc7a8Vsj/78HXvOIy"}]}
```
```

 - match: { license_status: "valid" }

 - do:
    xpack.license.get: {}

 ## a license object has 11 attributes
 - length: { license: 11 }
```

## bwc for licenses format
- do:
    xpack.license.post:
      acknowledge: true
      body: |
        {"licenses":[{"uid":"893361dc-9749-4997-93cb-802e3d7fa4a8","type":"gold","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1914278399999,"max_nodes":1,"issued_to":"issued_to","issuer":"issuer","signature":"AAAAAwAAAA2T3vqdBBetKQaBgxipAAAB mC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxRxVk1PSmkxakxZdW5IMlhlTHNoN1N2M XMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZ2xnQvSEFva0ppTHBkWkprrZWZSQi9iNmRQNkw1SlpLN0lDalZCS09 5MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdRFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEph ZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOE hQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53Mwk2K055c28zSmRnVkF1b2JSQkFSLV2VXUmVHNDZ2R3o2V E1qbVNQS2lxOHN5bUErZlNIWkZSVmZZIWEtaSU9wTTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldkMbmMzS GtQc3FWOTgzZ3ZZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdDEnGWFRreXNKNkE9PQAAAQB7pG FYgawfLm9zzT80LvcLHjy1t/v2uSzCQWKdXXhrwSy4WrAH2uK/+PEiQ7aEpW5erLsyJ5KLA6OEZJDaP7r+mj OPuLt0++l5j4DMn7ybMzOPHXWBc6LETE3+pp0GZPyOmwsDkZSRUegTtciR2R6z+mdnGrhOYM80y08KVW whdU/DHw41MK7ePo6tq73Nz49y9lDgt9fxA0t4ggEBPbnTDDBVQ25AjauY8sa0M5eg9rDDRayw1KamYWrara8 PIGX+2YjhtUeQhmlCPdlxc9wECJ7/knPss5bI3ZoXQR3fyXhjcXNnHEIsblqLrMCal3pLxs7lI+KPYMa2ZYL/am4 P"}]}

- match: { license_status: "valid" }

- do:
    xpack.license.get: {}

- length: { license: 11 }

## license version: 1.x
- do:
    xpack.license.post:
      acknowledge: true
      body: |
        {"licenses":[{"uid":"893361dc-9749-4997-93cb-802e3d7fa4a8","type":"subscription","subscription_type":"gold","issue_date_in_millis":1411948800000,"feature":" shield","expiry_date_in_millis":1914278399999,"max_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature ":"AAAAAQAAAA0LVAywwpSH94cyXr4zAAABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVo MVZwUzRxRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKZ2xnQvSEFva0pp THBkWkprrZWZSQi9iNmRQNkw1SlpLN0lDalZCS095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdRFQrN3JQeG wxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWkt WktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53Mwk2K 055c28zSmRuVkF1b2JSQkFSLV2VXUmVHNDZ2R3o2VE1qbVNQS2lxOHN5bUErZlNIWkZSVmZZIWEtaSU9w TTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldkMbmMzSGtQc3FWOTgzZ3ZZUcXMvQkt2RUZwMFJnZzlvL2d2 bDRWUzh6UG5pdDEnGWFRreXNKNkE9PQAAAQA4qscc/URRZVdFoLwgy9dqybYEQLW8YLkiAyPV5XHHH dtk+dtZIepiNEDkUXhSX2waVJlsNRF8/4kqplDfwNoD2TUM8fTgiIfiSiZYGDTGST+yW/5eAveEU5J5v1liBN27b wkqL+V4YAa0Tcm7NKKwjScWKAHiTU3vF8chPkGfCHE0kQgVwPC9RE82pTw0s6/uR4PfLGNFfqPM0uiE5n ucfVrtj89JQiO/KA/7ZyFbo7VTNXxZQt7T7rZWBCP9KIjptXzcWuk08Q5S+rSoJNYbFo3HGKtrCVsRz/55rceNtd wKKXu1IwnSeir4I1/KLduQTtFLy0+1th87VS8T88UT"}]}

```yaml
  - match: { license_status: "valid" }

  - do:
      xpack.license.get: {}

  - length: { license: 11 }

## multiple licenses version: 1.x
  - do:
      xpack.license.post:
        acknowledge: true
        body: |
```
          {"licenses":[{"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"internal","subscription_type":"none","issue_date_in_millis":1411948800000,"feature":"shiel
d","expiry_date_in_millis":1440892799999,"max_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"A
AAAAQAAAA04Q4ky3rFyyWLFkytEAAABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZw
UzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXMvRFk4d3JTZEx3R3RRRZ0pzU3lobWJKZnQvSEFva0ppTHBkW
kprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFFrrN3JQeGwxb0U0
MXZ0dDJHSERiZTVVLOHNzSDByWnpoZZEphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNW
GM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhqaVB2azhmmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28z
SmRnVkF1b2JSQkFFV2VXUmVHHNDZ2R3o2VE1qbVNQS2lxOHN5bUErZZlNIWkZSVmZIWEtaSU9wTTTJEND
VvT1NCYklacUYYK2FwRW9xa0t6dldMbmMzSGtQc3FWOTgzZ3UcXMvQkt2RUZwMFJnZzlvL2d2bDRWUz
h6UG5pdENHWFRreXNKNkE9PQAAAQBxMvUMn4h2E4R4TQMijahTxQj4LPQO4f1M79UxX/XkDlGcH+J5p
RHx08OtTRPsFL1lED+h+PIXx307Vo+PNDsOxrWvoYZeYBkOLAO3ny9vhQga+52jYhMxIuFrT9xbcSCSNpMh
GojgOIPU2WgiopVdVcimo1+Gk8VtklPB1wPwFzfOjOnPgp/Icx3WYpfkeAUUOyWUYiFIBAe4bnz84iF+xwLKb
gYk6aHF25ECBtdb/Uruhcm9+jEFpoIEUtCouvvk9C+NJZ4OickV4xpRgaRG2x9PONH8ZN0QGhGYhJGbisoCxu
DmlLsyVxqxfMu3n/r7/jdsEJScjAlSrsLDOu6H"},{"uid":"893361dc-9749-4997-93cb-
802e3dofh7aa","type":"internal","subscription_type":"none","issue_date_in_millis":1443484800000,"feature":"watc
her","expiry_date_in_millis":1914278399999,"max_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"
AAAAAQAAAA0Sc90guRIaQEmgLvMnAAABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMV
ZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXMvRFk4d3JTZEx3R3RRRZ0pzU3lobWJKZnQvSEFva0ppTHB
kWkprZWZSQi9iNmRQNkw1SlpLN0lDalZCS095MXRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdFFrrN3JQeGwxb0
U0MXZ0dDJHSERiZTVVLOHNzSDByWnpoZZEphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktN
WGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhqaVB2azhmmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c2
8zSmRnVkF1b2JSQkFFV2VXUmVHHNDZ2R3o2VE1qbVNQS2lxOHN5bUErZZlNIWkZSVmZIWEtaSU9wTTTJEN
DVvT1NCYklacUYYK2FwRW9xa0t6dldMbmMzSGtQc3FWOTgzZ3UcXMvQkt2RUZwMFJnZzlvL2d2bDRWUR
Uzh6UG5pdENHWFRreXNKNkE9PQAAAQCQ94dju0pnDZR3Uuypi0ic3aQJ+nvVqe+U8u79Dga5n1qIjcHDh7H
vIBJEkF+tnVPlo/PXV/x7BZSwVY1PVErit+6rYix1yuHEgqwxmx/VdRICjCaZM6tk0Ob4dZCPv6Ebn2Mmk89KH
C/PwiLPqF6QfwV/Pkpa8k2A3ORJmvYSDvXhe6tCs8dqc4ebrsFxqrZjwWh5CZSpzqqZBFXlngDv2N0hHhpGlueR
szD0JJ5dfEL5ZA1DDOrgO9OJVejSHyRqe1L5QRUNdXPVfS+EAG0Dd1cNdJ/sMpYCPnVjbw6iq2/YgM3cuztsX
VBY7ij4WnoP3ce7Zjs9TwHn+IqzftC6"}]}
```yaml
  - match: { license_status: "valid" }

  - do:
      xpack.license.get: {}
```

```
- length: { license: 11 }
- match: { license.uid: "893361dc-9749-4997-93cb-802e3dofh7aa" }
---
"Should throw 404 after license deletion":
 - do:
    xpack.license.delete: {}

 - match: { acknowledged: true }

 - do:
    xpack.license.get: {}
    catch: missing

---
"Should install a feature type license":

  # VERSION_NO_FEATURE_TYPE license version
 - do:
    xpack.license.post:
      acknowledge: true
      body: |
       {"license": {"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"gold","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1914278399999,"ma
x_nodes":1,"issued_to":"issued_to","issuer":"issuer","signature":"AAAAAgAAAA3U8+YmnvwC+CWsV/mRAA
ABmC9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N
2MXMvRFk4d3JTZEx3R3RRZ0pzU3lobWJKKZnQvSEFva0ppTHBkWkprrZWZSQi9iNmRQNkw1SlpLN0lDaIZC
S095MXRGN1lIZlpYYcVVTTnFrcTE2dzhJZmZrdFFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZ
EphZHBEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pS
OEhQaVB2azhmUk9QVzhFeTFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JJSQkFLV2VXUmVHNDZ2R3
o2VE1qbVNQUS2lxOHN5bUErZlNIWkZSVmZJWEtaSU9wTTJJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMbm
MzSGtQc3FWOTgzZ3ZUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdENGWFJreXNNTkkE9PQAAAQB
e8GfzDm6T537Iuuvjetb3xK5dvg0K5NQapv+rczWcQFxgCuzbF8plkgetP1aAGZP4uRESDQPMIOCsx4d0UqqAm
9f7GbBQ3l93P+PogInPFeEH9NvOmaAQovmxVM9SE6DsDqlX4cXSO+bgWpXPTd2LmpoQc1fXd6BZ8GeuyYp
VHVKp9hVU0tAYjw6HzYOE7+zuO1oJYOxElqy66AnIfkvHrvni+flym3tE7tDTgsDRaz7W3iBhaqiSntEqabEkvH
dPHQdSR99XGaEvnHO1paK01/35iZF6OXHsF7CCj+558GRXiVxzueOe7TsGSSt8g7YjZwV9bRCyU7oB4B/nidg
I"}}

 - match: { license_status: "valid" }

 - do:
    xpack.license.get: {}

 - length: { license: 11 }
---
"Cannot start basic":

 - do:
    catch: bad_request
```

```
xpack.license.post:
  acknowledge: true
  body: |
    {"license":{"uid":"893361dc-9749-4997-93cb-
802e3d7fa4a8","type":"basic","issue_date_in_millis":1411948800000,"expiry_date_in_millis":1914278399999,"ma
x_nodes":1,"issued_to":"issuedTo","issuer":"issuer","signature":"AAAAgAAAA0lKPZ0a7aZquUltho/AAABmC
9ZN0hjZDBGYnVyRXpCOW5Bb3FjZDAxOWpSbTVoMVZwUzRxVk1PSmkxakxZdW5IMlhlTHNoN1N2MXM
vRFk4d3JTZEx3R3RRRZ0pzU3lobWJKZnQvSEFva0ppTHBkWkprZWZZSQi9iNmRQNkw1SlpLN0lDalZCS095M
XRGN1lIZlpYcVVTTnFrcTE2dzhJZmZrdfFQrN3JQeGwxb0U0MXZ0dDJHSERiZTVLOHNzSDByWnpoZEphZH
BEZjUrTVBxRENNSXNsWWJjZllaODdzVmEzUjNiWktNWGM5TUhQV2plaUo4Q1JOUml4MXNuL0pSOEhQa
VB2azhmUk9QVzhFeTFFoM1Q0RnJXSG53MWk2K055c28zSmRnVkF1b2JSQkFLV2VXUmVHNDZ2R3o2VE1q
bVNQS2lxOHN5bUErZlNIWkZSVmZIWEtaSU9wTTJENDVvT1NCYklacUYyK2FwRW9xa0t6dldMMbmMzSGtQ
c3FWOTgzZ3ZUUcXMvQkt2RUZwMFJnZzlvL2d2bDRWUzh6UG5pdENNGWFRreXNKNkE9PQAAAQAALuQ44
S3IG6SzolcXVJ6Z4CIXORDrYQ+wdLCeey0XdujTslAOj+k+vNgo6wauc7Uswi01esHu4lb5IgpvKy7RRCbh5bj/z2
ubu2qMJqopp9BQyD7VQjVfqmG6seUMJwJ1a5Avvm9r41YPSPcrii3bKK2e1l6jK6N8ibCvnTyY/XkYGCJrBWT
SJePDbg6ErbyodrZ37x1StLbPWcNAkmweyHjDJnvYnbeZZO7A3NmubXZjW7Ttf8/YwQyE00PqMcl7fVPY3hk
KpAeHf8aaJbqkKYbqZuER3EWJX7ZvLVb1dNdNg8aXRn7YrkQcYwWgptYQpfV+D7yEJ4j5muAEoler"}}

 - match: { error.root_cause.0.reason: 'Installing basic licenses is no longer allowed. Use the POST
/_xpack/license/start_basic API to install a basic license that does not expire.' }
---
"Should fail gracefully when body content is not provided":

 - do:
    catch: bad_request
    xpack.license.post:
      acknowledge: true

 - match: { error.root_cause.0.reason: 'The license must be provided in the request body' }
---
"Current license is trial means not eligle to start trial":

 - do:
    xpack.license.get_trial_status: {}

 - match: { eligible_to_start_trial: false }

 - do:
    xpack.license.post_start_basic:
      acknowledge: true

 - match: { basic_was_started: true }

 - do:
    xpack.license.get_trial_status: {}

 - match: { eligible_to_start_trial: false }
```

```
  - do:
      catch: forbidden
      xpack.license.post_start_trial:
        acknowledge: true


  - match: { trial_was_started: false }
  - match: { error_message: "Operation failed: Trial was already activated." }
---
"Trial license cannot be basic":
  - do:
      catch: bad_request
      xpack.license.post_start_trial:
        type: "basic"
        acknowledge: true

---
"Can start basic license if do not already have basic":
  - do:
      xpack.license.get_basic_status: {}


  - match: { eligible_to_start_basic: true }


  - do:
      xpack.license.post_start_basic:
        acknowledge: true


  - match: { basic_was_started: true }
  - match: { acknowledged: true }


  - do:
        xpack.license.get_basic_status: {}


  - match: { eligible_to_start_basic: false }


  - do:
      catch: forbidden
      xpack.license.post_start_basic: {}


  - match: { basic_was_started: false }
  - match: { acknowledged: true }
  - match: { error_message: "Operation failed: Current license is basic." }
---
"Must acknowledge to start basic":
  - do:
      xpack.license.post_start_basic: {}


  - match: { basic_was_started: false }
  - match: { acknowledged: false }
  - match: { error_message: "Operation failed: Needs acknowledgement." }
```

```java
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.action.support.PlainActionFuture;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.cluster.ClusterState;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.License.OperationMode;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.test.junit.annotations.TestLogging;
import org.elasticsearch.transport.Transport;
import org.elasticsearch.xpack.core.TestXPackTransportClient;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.ml.action.CloseJobAction;
import org.elasticsearch.xpack.core.ml.action.DeleteDatafeedAction;
import org.elasticsearch.xpack.core.ml.action.DeleteJobAction;
import org.elasticsearch.xpack.core.ml.action.GetDatafeedsStatsAction;
import org.elasticsearch.xpack.core.ml.action.GetJobsStatsAction;
import org.elasticsearch.xpack.core.ml.action.OpenJobAction;
import org.elasticsearch.xpack.core.ml.action.PutDatafeedAction;
import org.elasticsearch.xpack.core.ml.action.PutJobAction;
import org.elasticsearch.xpack.core.ml.action.StartDatafeedAction;
import org.elasticsearch.xpack.core.ml.action.StopDatafeedAction;
import org.elasticsearch.xpack.core.ml.client.MachineLearningClient;
import org.elasticsearch.xpack.core.ml.datafeed.DatafeedState;
import org.elasticsearch.xpack.core.ml.job.config.JobState;
import org.elasticsearch.persistent.PersistentTasksCustomMetaData;
import org.elasticsearch.xpack.ml.LocalStateMachineLearning;
import org.elasticsearch.xpack.ml.support.BaseMlIntegTestCase;
import org.junit.Before;

import java.util.Collections;

import static org.hamcrest.Matchers.containsString;
import static org.hamcrest.Matchers.hasItem;
import static org.hamcrest.Matchers.is;

@TestLogging("org.elasticsearch.xpack.ml.action:DEBUG")
public class MachineLearningLicensingTests extends BaseMlIntegTestCase {

    @Before
```

```java
    public void resetLicensing() {
        enableLicensing();

        ensureStableCluster(1);
        ensureYellow();
    }

    public void testMachineLearningPutJobActionRestricted() throws Exception {
        String jobId = "testmachinelearningputjobactionrestricted";
        // Pick a license that does not allow machine learning
        License.OperationMode mode = randomInvalidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(false);
        // test that license restricted apis do not work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> listener = PlainActionFuture. newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), listener);
            listener.actionGet();
            fail("put job action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);
        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), listener);
            PutJobAction.Response response = listener.actionGet();
            assertNotNull(response);
        }
    }

    public void testMachineLearningOpenJobActionRestricted() throws Exception {
        String jobId = "testmachinelearningopenjobactionrestricted";
        assertMLAllowed(true);
        // test that license restricted apis do now work
```

```java
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response response = putJobListener.actionGet();
            assertNotNull(response);
        }

        // Pick a license that does not allow machine learning
        License.OperationMode mode = randomInvalidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(false);
        // test that license restricted apis do not work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<OpenJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), listener);
            listener.actionGet();
            fail("open job action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);

        // now that the license is invalid, the job should get closed:
        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.CLOSED, jobState);
        });

        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<OpenJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), listener);
            OpenJobAction.Response response = listener.actionGet();
            assertNotNull(response);
```

```java
            }
        }

    public void testMachineLearningPutDatafeedActionRestricted() throws Exception {
        String jobId = "testmachinelearningputdatafeedactionrestricted";
        String datafeedId = jobId + "-datafeed";
        assertMLAllowed(true);
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
        }

        // Pick a license that does not allow machine learning
        License.OperationMode mode = randomInvalidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(false);
        // test that license restricted apis do not work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId, Collections.singletonList(jobId))),
listener);
            listener.actionGet();
            fail("put datafeed action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);
        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutDatafeedAction.Response> listener = PlainActionFuture.newFuture();
```

```
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId, Collections.singletonList(jobId))),
    listener);
            PutDatafeedAction.Response response = listener.actionGet();
            assertNotNull(response);
        }
    }

    public void testAutoCloseJobWithDatafeed() throws Exception {
        String jobId = "testautoclosejobwithdatafeed";
        String datafeedId = jobId + "-datafeed";
        assertMLAllowed(true);
        String datafeedIndex = jobId + "-data";
        prepareCreate(datafeedIndex).addMapping("type", "{\"type\":{\"properties\":{\"time\":{\"type\":\"date\"}}}}",
            XContentType.JSON).get();
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            // put job
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
            // put datafeed
            PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                    new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
                        Collections.singletonList(datafeedIndex))), putDatafeedListener);
            PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
            assertNotNull(putDatafeedResponse);
            // open job
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);
            // start datafeed
            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
    listener);
            listener.actionGet();
        }

        if (randomBoolean()) {
            enableLicensing(randomInvalidLicenseType());
        } else {
            disableLicensing();
        }
```

```java
        assertMLAllowed(false);

        // now that the license is invalid, the job should be closed and datafeed stopped:
        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.CLOSED, jobState);

            DatafeedState datafeedState = getDatafeedStats(datafeedId).getDatafeedState();
            assertEquals(DatafeedState.STOPPED, datafeedState);

            ClusterState state = client().admin().cluster().prepareState().get().getState();
            PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
            assertEquals(0, tasks.taskMap().size());
        });

        enableLicensing(randomValidLicenseType());
        assertMLAllowed(true);

        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
            client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            // open job
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);
            // start datafeed
            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
listener);
            listener.actionGet();
        }

        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.OPENED, jobState);

            DatafeedState datafeedState = getDatafeedStats(datafeedId).getDatafeedState();
            assertEquals(DatafeedState.STARTED, datafeedState);

            ClusterState state = client().admin().cluster().prepareState().get().getState();
            PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
            assertEquals(2, tasks.taskMap().size());
        });

        if (randomBoolean()) {
            enableLicensing(randomInvalidLicenseType());
        } else {
```

```java
        disableLicensing();
    }
    assertMLAllowed(false);

    // now that the license is invalid, the job should be closed and datafeed stopped:
    assertBusy(() -> {
        JobState jobState = getJobStats(jobId).getState();
        assertEquals(JobState.CLOSED, jobState);

        DatafeedState datafeedState = getDatafeedStats(datafeedId).getDatafeedState();
        assertEquals(DatafeedState.STOPPED, datafeedState);

        ClusterState state = client().admin().cluster().prepareState().get().getState();
        PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
        assertEquals(0, tasks.taskMap().size());
    });
}

public void testMachineLearningStartDatafeedActionRestricted() throws Exception {
    String jobId = "testmachinelearningstartdatafeedactionrestricted";
    String datafeedId = jobId + "-datafeed";
    assertMLAllowed(true);
    String datafeedIndex = jobId + "-data";
    prepareCreate(datafeedIndex).addMapping("type", "{\"type\":{\"properties\":{\"time\":{\"type\":\"date\"}}}}",
        XContentType.JSON).get();
    // test that license restricted apis do now work
    Settings settings = internalCluster().transportClient().settings();
    try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
        PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
        new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
        PutJobAction.Response putJobResponse = putJobListener.actionGet();
        assertNotNull(putJobResponse);
        PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
        new MachineLearningClient(client).putDatafeed(
            new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
                Collections.singletonList(datafeedIndex))), putDatafeedListener);
        PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
        assertNotNull(putDatafeedResponse);
        PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
        new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
        OpenJobAction.Response openJobResponse = openJobListener.actionGet();
        assertNotNull(openJobResponse);
    }

    // Pick a license that does not allow machine learning
    License.OperationMode mode = randomInvalidLicenseType();
```

```java
        enableLicensing(mode);
        assertMLAllowed(false);

        // now that the license is invalid, the job should get closed:
        assertBusy(() -> {
            JobState jobState = getJobStats(jobId).getState();
            assertEquals(JobState.CLOSED, jobState);
            ClusterState state = client().admin().cluster().prepareState().get().getState();
            PersistentTasksCustomMetaData tasks = state.metaData().custom(PersistentTasksCustomMetaData.TYPE);
            assertEquals(0, tasks.taskMap().size());
        });

        // test that license restricted apis do not work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
listener);
            listener.actionGet();
            fail("start datafeed action should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
            assertThat(e.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.MACHINE_LEARNING));
        }

        // Pick a license that does allow machine learning
        mode = randomValidLicenseType();
        enableLicensing(mode);
        assertMLAllowed(true);
        // test that license restricted apis do now work
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            // re-open job now that the license is valid again
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);

            PlainActionFuture<StartDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(new StartDatafeedAction.Request(datafeedId, 0L),
listener);
            StartDatafeedAction.Response response = listener.actionGet();
            assertNotNull(response);
        }
```

```java
        }

    public void testMachineLearningStopDatafeedActionNotRestricted() throws Exception {
        String jobId = "testmachinelearningstopdatafeedactionnotrestricted";
        String datafeedId = jobId + "-datafeed";
        assertMLAllowed(true);
        String datafeedIndex = jobId + "-data";
        prepareCreate(datafeedIndex).addMapping("type", "{\"type\":{\"properties\":{\"time\":{\"type\":\"date\"}}}}",
            XContentType.JSON).get();
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
            PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putDatafeed(
                new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
                    Collections.singletonList(datafeedIndex))), putDatafeedListener);
            PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
            assertNotNull(putDatafeedResponse);
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);
            PlainActionFuture<StartDatafeedAction.Response> startDatafeedListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).startDatafeed(
                new StartDatafeedAction.Request(datafeedId, 0L), startDatafeedListener);
            StartDatafeedAction.Response startDatafeedResponse = startDatafeedListener.actionGet();
            assertNotNull(startDatafeedResponse);
        }

        boolean invalidLicense = randomBoolean();
        if (invalidLicense) {
            enableLicensing(randomInvalidLicenseType());
        } else {
            enableLicensing(randomValidLicenseType());
        }

        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<StopDatafeedAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).stopDatafeed(new StopDatafeedAction.Request(datafeedId), listener);
            if (invalidLicense) {
```

```java
            // the stop datafeed due to invalid license happens async, so check if the datafeed turns into stopped state:
            assertBusy(() -> {
                GetDatafeedsStatsAction.Response response =
                    new MachineLearningClient(client)
                        .getDatafeedsStats(new GetDatafeedsStatsAction.Request(datafeedId)).actionGet();
                assertEquals(DatafeedState.STOPPED, response.getResponse().results().get(0).getDatafeedState());
            });
        } else {
            listener.actionGet();
        }

        if (invalidLicense) {
            // the close due to invalid license happens async, so check if the job turns into closed state:
            assertBusy(() -> {
                GetJobsStatsAction.Response response =
                    new MachineLearningClient(client).getJobsStats(new
GetJobsStatsAction.Request(jobId)).actionGet();
                assertEquals(JobState.CLOSED, response.getResponse().results().get(0).getState());
            });
        }
    }
}

    public void testMachineLearningCloseJobActionNotRestricted() throws Exception {
        String jobId = "testmachinelearningclosejobactionnotrestricted";
        assertMLAllowed(true);
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
            PlainActionFuture<OpenJobAction.Response> openJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).openJob(new OpenJobAction.Request(jobId), openJobListener);
            OpenJobAction.Response openJobResponse = openJobListener.actionGet();
            assertNotNull(openJobResponse);
        }

        boolean invalidLicense = randomBoolean();
        if (invalidLicense) {
            enableLicensing(randomInvalidLicenseType());
        } else {
            enableLicensing(randomValidLicenseType());
        }
```

```java
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<CloseJobAction.Response> listener = PlainActionFuture.newFuture();
            CloseJobAction.Request request = new CloseJobAction.Request(jobId);
            request.setCloseTimeout(TimeValue.timeValueSeconds(20));
            if (invalidLicense) {
                // the close due to invalid license happens async, so check if the job turns into closed state:
                assertBusy(() -> {
                    GetJobsStatsAction.Response response =
                            new MachineLearningClient(client).getJobsStats(new
GetJobsStatsAction.Request(jobId)).actionGet();
                    assertEquals(JobState.CLOSED, response.getResponse().results().get(0).getState());
                });
            } else {
                new MachineLearningClient(client).closeJob(request, listener);
                listener.actionGet();
            }
        }
    }

    public void testMachineLearningDeleteJobActionNotRestricted() throws Exception {
        String jobId = "testmachinelearningclosejobactionnotrestricted";
        assertMLAllowed(true);
        // test that license restricted apis do now work
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
            PutJobAction.Response putJobResponse = putJobListener.actionGet();
            assertNotNull(putJobResponse);
        }

        // Pick a random license
        License.OperationMode mode = randomLicenseType();
        enableLicensing(mode);

        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            PlainActionFuture<DeleteJobAction.Response> listener = PlainActionFuture.newFuture();
            new MachineLearningClient(client).deleteJob(new DeleteJobAction.Request(jobId), listener);
            listener.actionGet();
        }
    }
```

```
public void testMachineLearningDeleteDatafeedActionNotRestricted() throws Exception {
    String jobId = "testmachinelearningdeletedatafeedactionnotrestricted";
    String datafeedId = jobId + "-datafeed";
    assertMLAllowed(true);
    // test that license restricted apis do now work
    Settings settings = internalCluster().transportClient().settings();
    try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
        client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
        PlainActionFuture<PutJobAction.Response> putJobListener = PlainActionFuture.newFuture();
        new MachineLearningClient(client).putJob(new PutJobAction.Request(createJob(jobId)), putJobListener);
        PutJobAction.Response putJobResponse = putJobListener.actionGet();
        assertNotNull(putJobResponse);
        PlainActionFuture<PutDatafeedAction.Response> putDatafeedListener = PlainActionFuture.newFuture();
        new MachineLearningClient(client).putDatafeed(
            new PutDatafeedAction.Request(createDatafeed(datafeedId, jobId,
                Collections.singletonList(jobId))), putDatafeedListener);
        PutDatafeedAction.Response putDatafeedResponse = putDatafeedListener.actionGet();
        assertNotNull(putDatafeedResponse);
    }

    // Pick a random license
    License.OperationMode mode = randomLicenseType();
    enableLicensing(mode);

    try (TransportClient client = new TestXPackTransportClient(settings, LocalStateMachineLearning.class)) {
        client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
        PlainActionFuture<DeleteDatafeedAction.Response> listener = PlainActionFuture.newFuture();
        new MachineLearningClient(client).deleteDatafeed(new DeleteDatafeedAction.Request(datafeedId),
listener);
        listener.actionGet();
    }
}

private static OperationMode randomInvalidLicenseType() {
    return randomFrom(License.OperationMode.GOLD, License.OperationMode.STANDARD,
License.OperationMode.BASIC);
}

private static OperationMode randomValidLicenseType() {
    return randomFrom(License.OperationMode.TRIAL, License.OperationMode.PLATINUM);
}

private static OperationMode randomLicenseType() {
    return randomFrom(License.OperationMode.values());
}
```

```
  private static void assertMLAllowed(boolean expected) {
    for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
      assertEquals(licenseState.isMachineLearningAllowed(), expected);
    }
  }

  public static void disableLicensing() {
    disableLicensing(randomValidLicenseType());
  }

  public static void disableLicensing(License.OperationMode operationMode) {
    for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
      licenseState.update(operationMode, false);
    }
  }

  public static void enableLicensing() {
    enableLicensing(randomValidLicenseType());
  }

  public static void enableLicensing(License.OperationMode operationMode) {
    for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
      licenseState.update(operationMode, true);
    }
  }
}
/*
*                           Apache License
*                     Version 2.0, January 2004
*                  http://www.apache.org/licenses/
*
*   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
*
*   1. Definitions.
*
*      "License" shall mean the terms and conditions for use, reproduction,
*      and distribution as defined by Sections 1 through 9 of this document.
*
*      "Licensor" shall mean the copyright owner or entity authorized by
*      the copyright owner that is granting the License.
*
*      "Legal Entity" shall mean the union of the acting entity and all
*      other entities that control, are controlled by, or are under common
*      control with that entity. For the purposes of this definition,
*      "control" means (i) the power, direct or indirect, to cause the
*      direction or management of such entity, whether by contract or
*      otherwise, or (ii) ownership of fifty percent (50%) or more of the
*      outstanding shares, or (iii) beneficial ownership of such entity.
```

```
 *
 *      "You" (or "Your") shall mean an individual or Legal Entity
 *      exercising permissions granted by this License.
 *
 *      "Source" form shall mean the preferred form for making modifications,
 *      including but not limited to software source code, documentation
 *      source, and configuration files.
 *
 *      "Object" form shall mean any form resulting from mechanical
 *      transformation or translation of a Source form, including but
 *      not limited to compiled object code, generated documentation,
 *      and conversions to other media types.
 *
 *      "Work" shall mean the work of authorship, whether in Source or
 *      Object form, made available under the License, as indicated by a
 *      copyright notice that is included in or attached to the work
 *      (an example is provided in the Appendix below).
 *
 *      "Derivative Works" shall mean any work, whether in Source or Object
 *      form, that is based on (or derived from) the Work and for which the
 *      editorial revisions, annotations, elaborations, or other modifications
 *      represent, as a whole, an original work of authorship. For the purposes
 *      of this License, Derivative Works shall not include works that remain
 *      separable from, or merely link (or bind by name) to the interfaces of,
 *      the Work and Derivative Works thereof.
 *
 *      "Contribution" shall mean any work of authorship, including
 *      the original version of the Work and any modifications or additions
 *      to that Work or Derivative Works thereof, that is intentionally
 *      submitted to Licensor for inclusion in the Work by the copyright owner
 *      or by an individual or Legal Entity authorized to submit on behalf of
 *      the copyright owner. For the purposes of this definition, "submitted"
 *      means any form of electronic, verbal, or written communication sent
 *      to the Licensor or its representatives, including but not limited to
 *      communication on electronic mailing lists, source code control systems,
 *      and issue tracking systems that are managed by, or on behalf of, the
 *      Licensor for the purpose of discussing and improving the Work, but
 *      excluding communication that is conspicuously marked or otherwise
 *      designated in writing by the copyright owner as "Not a Contribution."
 *
 *      "Contributor" shall mean Licensor and any individual or Legal Entity
 *      on behalf of whom a Contribution has been received by Licensor and
 *      subsequently incorporated within the Work.
 *
 *   2. Grant of Copyright License. Subject to the terms and conditions of
 *      this License, each Contributor hereby grants to You a perpetual,
 *      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
 *      copyright license to reproduce, prepare Derivative Works of,
```

```
 *      publicly display, publicly perform, sublicense, and distribute the
 *      Work and such Derivative Works in Source or Object form.
 *
 *   3. Grant of Patent License. Subject to the terms and conditions of
 *      this License, each Contributor hereby grants to You a perpetual,
 *      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
 *      (except as stated in this section) patent license to make, have made,
 *      use, offer to sell, sell, import, and otherwise transfer the Work,
 *      where such license applies only to those patent claims licensable
 *      by such Contributor that are necessarily infringed by their
 *      Contribution(s) alone or by combination of their Contribution(s)
 *      with the Work to which such Contribution(s) was submitted. If You
 *      institute patent litigation against any entity (including a
 *      cross-claim or counterclaim in a lawsuit) alleging that the Work
 *      or a Contribution incorporated within the Work constitutes direct
 *      or contributory patent infringement, then any patent licenses
 *      granted to You under this License for that Work shall terminate
 *      as of the date such litigation is filed.
 *
 *   4. Redistribution. You may reproduce and distribute copies of the
 *      Work or Derivative Works thereof in any medium, with or without
 *      modifications, and in Source or Object form, provided that You
 *      meet the following conditions:
 *
 *      (a) You must give any other recipients of the Work or
 *          Derivative Works a copy of this License; and
 *
 *      (b) You must cause any modified files to carry prominent notices
 *          stating that You changed the files; and
 *
 *      (c) You must retain, in the Source form of any Derivative Works
 *          that You distribute, all copyright, patent, trademark, and
 *          attribution notices from the Source form of the Work,
 *          excluding those notices that do not pertain to any part of
 *          the Derivative Works; and
 *
 *      (d) If the Work includes a "NOTICE" text file as part of its
 *          distribution, then any Derivative Works that You distribute must
 *          include a readable copy of the attribution notices contained
 *          within such NOTICE file, excluding those notices that do not
 *          pertain to any part of the Derivative Works, in at least one
 *          of the following places: within a NOTICE text file distributed
 *          as part of the Derivative Works; within the Source form or
 *          documentation, if provided along with the Derivative Works; or,
 *          within a display generated by the Derivative Works, if and
 *          wherever such third-party notices normally appear. The contents
 *          of the NOTICE file are for informational purposes only and
 *          do not modify the License. You may add Your own attribution
```

```
 *       notices within Derivative Works that You distribute, alongside
 *       or as an addendum to the NOTICE text from the Work, provided
 *       that such additional attribution notices cannot be construed
 *       as modifying the License.
 *
 *    You may add Your own copyright statement to Your modifications and
 *    may provide additional or different license terms and conditions
 *    for use, reproduction, or distribution of Your modifications, or
 *    for any such Derivative Works as a whole, provided Your use,
 *    reproduction, and distribution of the Work otherwise complies with
 *    the conditions stated in this License.
 *
 *  5. Submission of Contributions. Unless You explicitly state otherwise,
 *    any Contribution intentionally submitted for inclusion in the Work
 *    by You to the Licensor shall be under the terms and conditions of
 *    this License, without any additional terms or conditions.
 *    Notwithstanding the above, nothing herein shall supersede or modify
 *    the terms of any separate license agreement you may have executed
 *    with Licensor regarding such Contributions.
 *
 *  6. Trademarks. This License does not grant permission to use the trade
 *    names, trademarks, service marks, or product names of the Licensor,
 *    except as required for reasonable and customary use in describing the
 *    origin of the Work and reproducing the content of the NOTICE file.
 *
 *  7. Disclaimer of Warranty. Unless required by applicable law or
 *    agreed to in writing, Licensor provides the Work (and each
 *    Contributor provides its Contributions) on an "AS IS" BASIS,
 *    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 *    implied, including, without limitation, any warranties or conditions
 *    of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
 *    PARTICULAR PURPOSE. You are solely responsible for determining the
 *    appropriateness of using or redistributing the Work and assume any
 *    risks associated with Your exercise of permissions under this License.
 *
 *  8. Limitation of Liability. In no event and under no legal theory,
 *    whether in tort (including negligence), contract, or otherwise,
 *    unless required by applicable law (such as deliberate and grossly
 *    negligent acts) or agreed to in writing, shall any Contributor be
 *    liable to You for damages, including any direct, indirect, special,
 *    incidental, or consequential damages of any character arising as a
 *    result of this License or out of the use or inability to use the
 *    Work (including but not limited to damages for loss of goodwill,
 *    work stoppage, computer failure or malfunction, or any and all
 *    other commercial damages or losses), even if such Contributor
 *    has been advised of the possibility of such damages.
 *
 *  9. Accepting Warranty or Additional Liability. While redistributing
```

```
{
 "metadata": {
   "name": "X-Pack Monitoring: License Expiration",
   "xpack": {
    "link": "license",
    "expires_days": [ 60, 30, 14, 7 ],
    "severity": 0,
    "alert_index": ".monitoring-alerts-6",
    "cluster_uuid": "${monitoring.watch.cluster_uuid}",
    "type": "monitoring",
    "version_created": 7000001,
```

```
              "watch": "${monitoring.watch.id}"
            }
          },
          "trigger": {
            "schedule": {
              "interval": "1m"
            }
          },
          "input": {
            "chain": {
              "inputs": [
                {
                  "check": {
                    "search": {
                      "request": {
                        "indices": [
                          ".monitoring-es-*"
                        ],
                        "body": {
                          "size": 1,
                          "sort": [
                            {
                              "timestamp": {
                                "order": "desc"
                              }
                            }
                          ],
                          "_source": [
                            "license.*"
                          ],
                          "query": {
                            "bool": {
                              "filter": [
                                {
                                  "term": {
                                    "cluster_uuid": "{{ctx.metadata.xpack.cluster_uuid}}"
                                  }
                                },
                                {
                                  "term": {
                                    "type": "cluster_stats"
                                  }
                                },
                                {
                                  "range": {
                                    "timestamp": {
                                      "gte": "now-2m"
                                    }
```

```
                }
              }
            ]
          }
        }
      }
    }
  },
  {
    "alert": {
      "search": {
        "request": {
          "indices": [
            ".monitoring-alerts-6"
          ],
          "body": {
            "size": 1,
            "terminate_after": 1,
            "query": {
              "bool": {
                "filter": {
                  "term": {
                    "_id": "{{ctx.watch_id}}"
                  }
                }
              }
            },
            "sort": [
              { "timestamp": { "order": "desc" } }
            ]
          }
        }
      }
    }
  },
  {
    "kibana_settings": {
      "search": {
        "request": {
          "indices": [
            ".monitoring-kibana-6-*"
          ],
          "body": {
            "size": 1,
            "query": {
              "bool": {
```

```
            "filter": {
              "term": {
                "type": "kibana_settings"
              }
            }
          }
        },
        "sort": [
          {
            "timestamp": {
              "order": "desc"
            }
          }
        ]
      }
    }
  }
]
      }
    },
  "condition": {
    "script": {
      "source": "if (ctx.payload.check.hits.total == 0) {return false;}def license =
ctx.payload.check.hits.hits[0]._source.license;if (license == null) {return false;}ctx.vars.fails_check = false;Instant
expiry = Instant.ofEpochMilli(license.expiry_date_in_millis);ctx.vars.expiry = expiry;if (license.status != 'active')
{ctx.vars.expired = true;ctx.vars.fails_check = true;ctx.metadata.xpack.severity = 2001;} else {Instant now =
Instant.ofEpochMilli(new Date().getTime());ctx.vars.now = now;for (int i = ctx.metadata.xpack.expires_days.length
- 1;i > -1;--i) {if (license.type == 'trial' && i < 2) {break;}Instant fromNow =
now.plusSeconds(ctx.metadata.xpack.expires_days[i] * 24 * 60 * 60);if (fromNow.isAfter(expiry))
{ctx.vars.fails_check = true;ctx.metadata.xpack.severity = i * 1000;break;}}}ctx.vars.not_resolved =
(ctx.payload.alert.hits.total == 1 && ctx.payload.alert.hits.hits[0]._source.resolved_timestamp ==
null);ctx.vars.update = ctx.vars.fails_check || ctx.vars.not_resolved;"
    }
  },
  "transform": {
    "script": {
      "source": "ctx.vars.email_recipient = (ctx.payload.kibana_settings.hits.total > 0) ?
ctx.payload.kibana_settings.hits.hits[0]._source.kibana_settings.xpack.default_admin_email : null;ctx.vars.is_new =
ctx.vars.fails_check && !ctx.vars.not_resolved;ctx.vars.is_resolved = !ctx.vars.fails_check &&
ctx.vars.not_resolved;def alertMessage = null;if (ctx.vars.fails_check) { alertMessage = 'Update your license.';} if
(ctx.vars.not_resolved) { ctx.payload = ctx.payload.alert.hits.hits[0]._source;ctx.payload.metadata =
ctx.metadata.xpack;if (ctx.vars.fails_check == false) { ctx.payload.resolved_timestamp = ctx.execution_time;} }
else { ctx.payload = [ 'timestamp': ctx.execution_time, 'prefix': 'This cluster\\'s license is going to expire in
{{#relativeTime}}metadata.time{{/relativeTime}} at {{#absoluteTime}}metadata.time{{/absoluteTime}}.',
'message': alertMessage, 'metadata': ctx.metadata.xpack ];} if (ctx.vars.fails_check) { ctx.payload.metadata.time =
ctx.vars.expiry.toString();} ctx.payload.update_timestamp = ctx.execution_time;return ctx.payload;"
```

```
      }
    },
    "actions": {
      "add_to_alerts_index": {
        "index": {
          "index": ".monitoring-alerts-6",
          "doc_type": "doc",
          "doc_id": "${monitoring.watch.unique_id}"
        }
      },
      "send_email_to_admin": {
        "condition": {
          "script": "return ctx.vars.email_recipient != null && (ctx.vars.is_new || ctx.vars.is_resolved)"
        },
        "email": {
          "to": "X-Pack Admin <{{ctx.vars.email_recipient}}>",
          "from": "X-Pack Admin <{{ctx.vars.email_recipient}}>",
          "subject":
"[{{#ctx.vars.is_new}}NEW{{/ctx.vars.is_new}}{{#ctx.vars.is_resolved}}RESOLVED{{/ctx.vars.is_resolved}}]
{{ctx.metadata.name}}",
          "body": {
            "text": "{{#ctx.vars.is_resolved}}This cluster alert has been resolved: {{/ctx.vars.is_resolved}} This cluster's
license {{#ctx.vars.is_new}}is going to expire on
{{ctx.payload.metadata.time}}{{/ctx.vars.is_new}}{{#ctx.vars.is_resolved}}was going to
expire{{ctx.payload.metadata.time}}{{/ctx.vars.is_resolved}}.
{{#ctx.vars.is_new}}{{ctx.payload.message}}{{/ctx.vars.is_new}}"
          }
        }
      }
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license;

import org.apache.http.message.BasicHeader;
import org.elasticsearch.ElasticsearchSecurityException;
import org.elasticsearch.action.DocWriteResponse;
import org.elasticsearch.action.admin.cluster.health.ClusterHealthResponse;
import org.elasticsearch.action.admin.cluster.node.stats.NodesStatsResponse;
import org.elasticsearch.action.admin.cluster.stats.ClusterStatsIndices;
import org.elasticsearch.action.admin.cluster.stats.ClusterStatsResponse;
import org.elasticsearch.action.admin.indices.stats.IndicesStatsResponse;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.client.Client;
```

```java
import org.elasticsearch.client.Response;
import org.elasticsearch.client.ResponseException;
import org.elasticsearch.client.transport.NoNodeAvailableException;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.cluster.routing.ShardRoutingState;
import org.elasticsearch.common.settings.SecureString;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.common.util.concurrent.ThreadContext;
import org.elasticsearch.discovery.DiscoveryModule;
import org.elasticsearch.node.MockNode;
import org.elasticsearch.node.Node;
import org.elasticsearch.plugins.Plugin;
import org.elasticsearch.rest.RestStatus;
import org.elasticsearch.test.MockHttpTransport;
import org.elasticsearch.test.SecurityIntegTestCase;
import org.elasticsearch.test.SecuritySettingsSource;
import org.elasticsearch.test.SecuritySettingsSourceField;
import org.elasticsearch.test.discovery.TestZenDiscovery;
import org.elasticsearch.test.junit.annotations.TestLogging;
import org.elasticsearch.transport.Netty4Plugin;
import org.elasticsearch.transport.Transport;
import org.elasticsearch.xpack.core.TestXPackTransportClient;
import org.elasticsearch.xpack.core.XPackField;
import org.elasticsearch.xpack.core.security.SecurityField;
import org.elasticsearch.xpack.core.security.action.user.GetUsersResponse;
import org.elasticsearch.xpack.core.security.authc.support.UsernamePasswordToken;
import org.elasticsearch.xpack.core.security.client.SecurityClient;
import org.elasticsearch.xpack.security.LocalStateSecurity;
import org.junit.After;
import org.junit.Before;

import java.nio.file.Files;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;

import static org.elasticsearch.common.xcontent.XContentFactory.jsonBuilder;
import static org.elasticsearch.test.hamcrest.ElasticsearchAssertions.assertNoFailures;
import static org.hamcrest.Matchers.containsString;
import static org.hamcrest.Matchers.greaterThanOrEqualTo;
import static org.hamcrest.Matchers.hasItem;
import static org.hamcrest.Matchers.is;
import static org.hamcrest.Matchers.notNullValue;

@TestLogging("org.elasticsearch.cluster.service:TRACE,org.elasticsearch.discovery.zen:TRACE,org.elasticsearch.
action.search:TRACE," +
    "org.elasticsearch.search:TRACE")
```

```java
public class LicensingTests extends SecurityIntegTestCase {
  public static final String ROLES =
      SecuritySettingsSource.TEST_ROLE + ":\n" +
          "  cluster: [ all ]\n" +
          "  indices:\n" +
          "    - names: '*'\n" +
          "      privileges: [manage]\n" +
          "    - names: '/.*/'\n" +
          "      privileges: [write]\n" +
          "    - names: 'test'\n" +
          "      privileges: [read]\n" +
          "    - names: 'test1'\n" +
          "      privileges: [read]\n" +
          "\n" +
          "role_a:\n" +
          "  indices:\n" +
          "    - names: 'a'\n" +
          "      privileges: [all]\n" +
          "\n" +
          "role_b:\n" +
          "  indices:\n" +
          "    - names: 'b'\n" +
          "      privileges: [all]\n";

  public static final String USERS =
      SecuritySettingsSource.CONFIG_STANDARD_USER +
          "user_a:{plain}passwd\n" +
          "user_b:{plain}passwd\n";

  public static final String USERS_ROLES =
      SecuritySettingsSource.CONFIG_STANDARD_USER_ROLES +
          "role_a:user_a,user_b\n" +
          "role_b:user_b\n";

  @Override
  protected String configRoles() {
    return ROLES;
  }

  @Override
  protected String configUsers() {
    return USERS;
  }

  @Override
  protected String configUsersRoles() {
    return USERS_ROLES;
  }
```

```java
@Override
protected boolean addMockHttpTransport() {
    return false; // enable http
}

@Override
protected Collection<Class<? extends Plugin>> nodePlugins() {
    ArrayList<Class<? extends Plugin>> plugins = new ArrayList<>(super.nodePlugins());
    plugins.add(Netty4Plugin.class); // for http
    return plugins;
}

@Override
protected int maxNumberOfNodes() {
    return super.maxNumberOfNodes() + 1;
}

@Override
public Settings nodeSettings(int nodeOrdinal) {
    return Settings.builder().put(super.nodeSettings(nodeOrdinal))
        .put(TestZenDiscovery.USE_MOCK_PINGS.getKey(), false)
        .build();
}

@Before
public void resetLicensing() {
    enableLicensing();
}

@After
public void cleanupSecurityIndex() {
    deleteSecurityIndex();
}

public void testEnableDisableBehaviour() throws Exception {
    IndexResponse indexResponse = index("test", "type", jsonBuilder()
        .startObject()
        .field("name", "value")
        .endObject());
    assertEquals(DocWriteResponse.Result.CREATED, indexResponse.getResult());


    indexResponse = index("test1", "type", jsonBuilder()
        .startObject()
        .field("name", "value1")
        .endObject());
    assertEquals(DocWriteResponse.Result.CREATED, indexResponse.getResult());
```

```java
    refresh();
    // wait for all replicas to be started (to make sure that there are no more cluster state updates when we disable
licensing)
    assertBusy(() -> assertTrue(client().admin().cluster().prepareState().get().getState().routingTable()
        .shardsWithState(ShardRoutingState.INITIALIZING).isEmpty()));

    Client client = internalCluster().transportClient();

    disableLicensing();

    assertElasticsearchSecurityException(() -> client.admin().indices().prepareStats().get());
    assertElasticsearchSecurityException(() -> client.admin().cluster().prepareClusterStats().get());
    assertElasticsearchSecurityException(() -> client.admin().cluster().prepareHealth().get());
    assertElasticsearchSecurityException(() -> client.admin().cluster().prepareNodesStats().get());

    enableLicensing(randomFrom(License.OperationMode.values()));

    IndicesStatsResponse indicesStatsResponse = client.admin().indices().prepareStats().get();
    assertNoFailures(indicesStatsResponse);

    ClusterStatsResponse clusterStatsNodeResponse = client.admin().cluster().prepareClusterStats().get();
    assertThat(clusterStatsNodeResponse, notNullValue());
    ClusterStatsIndices indices = clusterStatsNodeResponse.getIndicesStats();
    assertThat(indices, notNullValue());
    assertThat(indices.getIndexCount(), greaterThanOrEqualTo(2));

    ClusterHealthResponse clusterIndexHealth = client.admin().cluster().prepareHealth().get();
    assertThat(clusterIndexHealth, notNullValue());

    NodesStatsResponse nodeStats = client.admin().cluster().prepareNodesStats().get();
    assertThat(nodeStats, notNullValue());
}

public void testRestAuthenticationByLicenseType() throws Exception {
    Response response = getRestClient().performRequest("GET", "/");
    // the default of the licensing tests is basic
    assertThat(response.getStatusLine().getStatusCode(), is(200));
    ResponseException e = expectThrows(ResponseException.class,
        () -> getRestClient().performRequest("GET", "/_xpack/security/_authenticate"));
    assertThat(e.getResponse().getStatusLine().getStatusCode(), is(403));

    // generate a new license with a mode that enables auth
    License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.TRIAL,
        License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
    enableLicensing(mode);
    e = expectThrows(ResponseException.class, () -> getRestClient().performRequest("GET", "/"));
```

```java
        assertThat(e.getResponse().getStatusLine().getStatusCode(), is(401));
        e = expectThrows(ResponseException.class,
            () -> getRestClient().performRequest("GET", "/_xpack/security/_authenticate"));
        assertThat(e.getResponse().getStatusLine().getStatusCode(), is(401));

        final String basicAuthValue =
UsernamePasswordToken.basicAuthHeaderValue(SecuritySettingsSource.TEST_USER_NAME,
            new SecureString(SecuritySettingsSourceField.TEST_PASSWORD.toCharArray()));
        response = getRestClient().performRequest("GET", "/", new BasicHeader("Authorization", basicAuthValue));
        assertThat(response.getStatusLine().getStatusCode(), is(200));
        response = getRestClient().performRequest("GET", "/_xpack/security/_authenticate",
            new BasicHeader("Authorization", basicAuthValue));
        assertThat(response.getStatusLine().getStatusCode(), is(200));

    }

    public void testSecurityActionsByLicenseType() throws Exception {
        // security actions should not work!
        Settings settings = internalCluster().transportClient().settings();
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            new SecurityClient(client).prepareGetUsers().get();
            fail("security actions should not be enabled!");
        } catch (ElasticsearchSecurityException e) {
            assertThat(e.status(), is(RestStatus.FORBIDDEN));
            assertThat(e.getMessage(), containsString("non-compliant"));
        }

        // enable a license that enables security
        License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.TRIAL,
            License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
        enableLicensing(mode);
        // security actions should not work!
        try (TransportClient client = new TestXPackTransportClient(settings, LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            GetUsersResponse response = new SecurityClient(client).prepareGetUsers().get();
            assertNotNull(response);
        }
    }

    public void testTransportClientAuthenticationByLicenseType() throws Exception {
        Settings.Builder builder = Settings.builder()
            .put(internalCluster().transportClient().settings());
        // remove user info
        builder.remove(SecurityField.USER_SETTING.getKey());
```

```java
        builder.remove(ThreadContext.PREFIX + "." + UsernamePasswordToken.BASIC_AUTH_HEADER);

        // basic has no auth
        try (TransportClient client = new TestXPackTransportClient(builder.build(), LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            assertGreenClusterState(client);
        }

        // enable a license that enables security
        License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.TRIAL,
            License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
        enableLicensing(mode);

        try (TransportClient client = new TestXPackTransportClient(builder.build(), LocalStateSecurity.class)) {
client.addTransportAddress(internalCluster().getDataNodeInstance(Transport.class).boundAddress().publishAddress
());
            client.admin().cluster().prepareHealth().get();
            fail("should not have been able to connect to a node!");
        } catch (NoNodeAvailableException e) {
            // expected
        }
    }

    public void testNodeJoinWithoutSecurityExplicitlyEnabled() throws Exception {
        License.OperationMode mode = randomFrom(License.OperationMode.GOLD,
License.OperationMode.PLATINUM, License.OperationMode.STANDARD);
        enableLicensing(mode);
        ensureGreen();

        Path home = createTempDir();
        Path conf = home.resolve("config");
        Files.createDirectories(conf);
        Settings nodeSettings = Settings.builder()
            .put(nodeSettings(maxNumberOfNodes() - 1).filter(s -> "xpack.security.enabled".equals(s) == false))
            .put("node.name", "my-test-node")
            .put("network.host", "localhost")
            .put("cluster.name", internalCluster().getClusterName())
            .put("discovery.zen.minimum_master_nodes",
                internalCluster().getInstance(Settings.class).get("discovery.zen.minimum_master_nodes"))
            .put("path.home", home)
            .put(TestZenDiscovery.USE_MOCK_PINGS.getKey(), false)
            .put(DiscoveryModule.DISCOVERY_TYPE_SETTING.getKey(), "test-zen")
            .put(DiscoveryModule.DISCOVERY_HOSTS_PROVIDER_SETTING.getKey(), "test-zen")
            .build();
        Collection<Class<? extends Plugin>> mockPlugins = Arrays.asList(LocalStateSecurity.class,
TestZenDiscovery.TestPlugin.class,
```

```
          MockHttpTransport.TestPlugin.class);
      try (Node node = new MockNode(nodeSettings, mockPlugins)) {
        node.start();
        ensureStableCluster(cluster().size() + 1);
      }
    }

    private static void assertElasticsearchSecurityException(ThrowingRunnable runnable) {
      ElasticsearchSecurityException ee = expectThrows(ElasticsearchSecurityException.class, runnable);
      assertThat(ee.getMetadata(LicenseUtils.EXPIRED_FEATURE_METADATA),
hasItem(XPackField.SECURITY));
      assertThat(ee.status(), is(RestStatus.FORBIDDEN));
    }

    public static void disableLicensing() {
      disableLicensing(License.OperationMode.BASIC);
    }

    public static void disableLicensing(License.OperationMode operationMode) {
      for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
        licenseState.update(operationMode, false);
      }
    }

    public static void enableLicensing() {
      enableLicensing(License.OperationMode.BASIC);
    }

    public static void enableLicensing(License.OperationMode operationMode) {
      for (XPackLicenseState licenseState : internalCluster().getInstances(XPackLicenseState.class)) {
        licenseState.update(operationMode, true);
      }
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.xpack.security.authz.permission;

import org.elasticsearch.action.get.GetAction;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.security.authz.permission.Role;
import org.elasticsearch.xpack.core.security.authz.privilege.Privilege;
import org.junit.Before;

import java.util.function.Predicate;
```

```java
import static org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege.MONITOR;
import static org.elasticsearch.xpack.core.security.authz.privilege.IndexPrivilege.READ;
import static org.hamcrest.Matchers.is;
import static org.hamcrest.Matchers.notNullValue;

public class PermissionTests extends ESTestCase {
    private Role permission;

    @Before
    public void init() {
        Role.Builder builder = Role.builder("test");
        builder.add(MONITOR, "test_*", "/foo.*/");
        builder.add(READ, "baz_*foo", "/fool.*bar/");
        builder.add(MONITOR, "/bar.*/");
        permission = builder.build();
    }

    public void testAllowedIndicesMatcherAction() throws Exception {
        testAllowedIndicesMatcher(permission.indices().allowedIndicesMatcher(GetAction.NAME));
    }

    public void testAllowedIndicesMatcherActionCaching() throws Exception {
        Predicate<String> matcher1 = permission.indices().allowedIndicesMatcher(GetAction.NAME);
        Predicate<String> matcher2 = permission.indices().allowedIndicesMatcher(GetAction.NAME);
        assertThat(matcher1, is(matcher2));
    }

    public void testBuildEmptyRole() {
        Role.Builder permission = Role.builder(new String[] { "some_role" });
        Role role = permission.build();
        assertThat(role, notNullValue());
        assertThat(role.cluster(), notNullValue());
        assertThat(role.indices(), notNullValue());
        assertThat(role.runAs(), notNullValue());
    }

    public void testRunAs() {
        Role permission = Role.builder("some_role")
                .runAs(new Privilege("name", "user1", "run*"))
                .build();
        assertThat(permission.runAs().check("user1"), is(true));
        assertThat(permission.runAs().check("user"), is(false));
        assertThat(permission.runAs().check("run" + randomAlphaOfLengthBetween(1, 10)), is(true));
    }

    // "baz_*foo", "/fool.*bar/"
    private void testAllowedIndicesMatcher(Predicate<String> indicesMatcher) {
```

```java
        assertThat(indicesMatcher.test("foobar"), is(false));
        assertThat(indicesMatcher.test("fool"), is(false));
        assertThat(indicesMatcher.test("fool2bar"), is(true));
        assertThat(indicesMatcher.test("baz_foo"), is(true));
        assertThat(indicesMatcher.test("barbapapa"), is(false));
    }
}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.xpack.security.authz.permission;

import org.elasticsearch.ElasticsearchParseException;
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.test.ESTestCase;
import org.elasticsearch.xpack.core.security.authz.RoleDescriptor;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissions;
import org.elasticsearch.xpack.core.security.authz.permission.FieldPermissionsDefinition;

import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.atomic.AtomicReferenceArray;

import static org.hamcrest.Matchers.containsString;

public class FieldPermissionsTests extends ESTestCase {

    public void testParseFieldPermissions() throws Exception {
        String q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"field_security\": {" +
            "\"grant\": [\"f1\", \"f2\", \"f3\", \"f4\"]," +
            "\"except\": [\"f3\",\"f4\"]" +
            "}}]}";
        RoleDescriptor rd =
            RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
            new String[] { "f1", "f2", "f3", "f4" });
        assertArrayEquals(rd.getIndicesPrivileges()[0].getDeniedFields(),
            new String[] { "f3", "f4" });

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"field_security\": {" +
            "\"except\": [\"f3\",\"f4\"]," +
            "\"grant\": [\"f1\", \"f2\", \"f3\", \"f4\"]" +
```

```
    "}}]}";
rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
    new String[] { "f1", "f2", "f3", "f4" });
assertArrayEquals(rd.getIndicesPrivileges()[0].getDeniedFields(),
    new String[] { "f3", "f4" });


q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
    "\"field_security\": {" +
    "\"grant\": [\"f1\", \"f2\"]" +
    "}}]}";
rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
    new String[] { "f1", "f2" });
assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());


q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
    "\"field_security\": {" +
    "\"grant\": []" +
    "}}]}";
rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[] {});
assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());


q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
    "\"field_security\": {" +
    "\"except\": []," +
    "\"grant\": []" +
    "}}]}";
rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[] {});
assertArrayEquals(rd.getIndicesPrivileges()[0].getDeniedFields(), new String[] {});


final String exceptWithoutGrant = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\":" +
    " [\"p3\"], \"field_security\": {" +
    "\"except\": [\"f1\"]" +
    "}}]}";
ElasticsearchParseException e = expectThrows(ElasticsearchParseException.class,
    () -> RoleDescriptor.parse("test", new BytesArray(exceptWithoutGrant), false,
        XContentType.JSON));
assertThat(e.getDetailedMessage(),
    containsString("failed to parse indices privileges for role [test]. field_security"
        + " requires grant if except is given"));


final String grantNull = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"]," +
    " \"field_security\": {" +
    "\"grant\": null" +
    "}}]}";
```

```
e = expectThrows(ElasticsearchParseException.class,
    () -> RoleDescriptor.parse("test", new BytesArray(grantNull), false,
        XContentType.JSON));
assertThat(e.getDetailedMessage(), containsString("failed to parse indices privileges for" +
    " role [test]. grant must not be null."));


final String exceptNull = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": " +
    "[\"p3\"], \"field_security\": {" +
    "\"grant\": [\"*\"]," +
    "\"except\": null" +
    "}}]}";
e = expectThrows(ElasticsearchParseException.class,
    () -> RoleDescriptor.parse("test", new BytesArray(exceptNull), false,
        XContentType.JSON));
assertThat(e.getDetailedMessage(),
    containsString("failed to parse indices privileges for role [test]. except must" +
        " not be null."));


final String exceptGrantNull = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": " +
    "[\"p3\"], \"field_security\": {" +
    "\"grant\": null," +
    "\"except\": null" +
    "}}]}";
e = expectThrows(ElasticsearchParseException.class,
    () -> RoleDescriptor.parse("test", new BytesArray(exceptGrantNull), false,
        XContentType.JSON));
assertThat(e.getDetailedMessage(), containsString("failed to parse indices privileges " +
    "for role [test]. grant must not be null."));


final String bothFieldsMissing = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": " +
    "[\"p3\"], \"field_security\": {" +
    "}}]}";
e = expectThrows(ElasticsearchParseException.class,
    () -> RoleDescriptor.parse("test", new BytesArray(bothFieldsMissing), false,
        XContentType.JSON));
assertThat(e.getDetailedMessage(), containsString("failed to parse indices privileges " +
    "for role [test]. \"field_security\" must not be empty."));


// try with two indices and mix order a little
q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
    "\"field_security\": {" +
    "\"grant\": []" +
    "}}," +
    "{\"names\": \"idx3\",\n" +
    " \"field_security\": {\n" +
    " \"grant\": [\"*\"], \n" +
    " \"except\": [\"f2\"]}," +
    "\"privileges\": [\"p3\"]}]}";
```

```
        rd = RoleDescriptor.parse("test", new BytesArray(q), false, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[] { });
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());
        assertArrayEquals(rd.getIndicesPrivileges()[1].getGrantedFields(), new String[] {"*"});
        assertArrayEquals(rd.getIndicesPrivileges()[1].getDeniedFields(), new String[] {"f2"});
    }


    // test old syntax for field permissions
    public void testBWCFieldPermissions() throws Exception {
        String q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"fields\": [\"f1\", \"f2\"]" +
            "}]}";
        RoleDescriptor rd = RoleDescriptor.parse("test", new BytesArray(q), true,
            XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(),
            new String[]{"f1", "f2"});
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());

        final String failingQuery = q;
        ElasticsearchParseException e = expectThrows(ElasticsearchParseException.class,
            () -> RoleDescriptor.parse("test", new BytesArray(failingQuery), false,
                XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("[\"fields\": [...]] format has " +
            "changed for field permissions in role [test]" +
            ", use [\"field_security\": {\"grant\":[...],\"except\":[...]}] instead"));

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"fields\": []" +
            "}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), true, XContentType.JSON);
        assertArrayEquals(rd.getIndicesPrivileges()[0].getGrantedFields(), new String[]{ });
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());
        final String failingQuery2 = q;
        e = expectThrows(ElasticsearchParseException.class,
            () -> RoleDescriptor.parse("test", new BytesArray(failingQuery2), false,
                XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("[\"fields\": [...]] format has " +
            "changed for field permissions in role [test]" +
            ", use [\"field_security\": {\"grant\":[...],\"except\":[...]}] instead"));

        q = "{\"indices\": [ {\"names\": \"idx2\", \"privileges\": [\"p3\"], " +
            "\"fields\": null" +
            "}]}";
        rd = RoleDescriptor.parse("test", new BytesArray(q), true, XContentType.JSON);
        assertNull(rd.getIndicesPrivileges()[0].getGrantedFields());
        assertNull(rd.getIndicesPrivileges()[0].getDeniedFields());
        final String failingQuery3 = q;
        e = expectThrows(ElasticsearchParseException.class,
```

```java
            () -> RoleDescriptor.parse("test", new BytesArray(failingQuery3), false,
                XContentType.JSON));
        assertThat(e.getDetailedMessage(), containsString("[\"fields\": [...]] format has " +
            "changed for field permissions in role [test]" +
            ", use [\"field_security\": {\"grant\":[...],\"except\":[...]}] instead"));
    }


    public void testFieldPermissionsHashCodeThreadSafe() throws Exception {
        final int numThreads = scaledRandomIntBetween(4, 16);
        final FieldPermissions fieldPermissions =new FieldPermissions(
            new FieldPermissionsDefinition(new String[] { "*" }, new String[] { "foo" }));
        final CountDownLatch latch = new CountDownLatch(numThreads + 1);
        final AtomicReferenceArray<Integer> hashCodes = new AtomicReferenceArray<>(numThreads);
        List<Thread> threads = new ArrayList<>(numThreads);
        for (int i = 0; i < numThreads; i++) {
            final int threadNum = i;
            threads.add(new Thread(() -> {
                latch.countDown();
                try {
                    latch.await();
                } catch (InterruptedException e) {
                    Thread.currentThread().interrupt();
                }
                final int hashCode = fieldPermissions.hashCode();
                hashCodes.set(threadNum, hashCode);
            }));
        }

        for (Thread thread : threads) {
            thread.start();
        }
        latch.countDown();
        for (Thread thread : threads) {
            thread.join();
        }

        final int hashCode = fieldPermissions.hashCode();
        for (int i = 0; i < numThreads; i++) {
            assertEquals((Integer) hashCode, hashCodes.get(i));
        }
    }
}
```

Apache Log4j SLF4J Binding

Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

Dropwizard

Copyright 2010-2013 Coda Hale and Yammer, Inc., 2014-2016 Dropwizard Team

This product includes software developed by Coda Hale and Yammer, Inc.

```html
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="utf-8">
 <link rel="dns-prefetch" href="https://assets-cdn.github.com">
 <link rel="dns-prefetch" href="https://avatars0.githubusercontent.com">
 <link rel="dns-prefetch" href="https://avatars1.githubusercontent.com">
 <link rel="dns-prefetch" href="https://avatars2.githubusercontent.com">
 <link rel="dns-prefetch" href="https://avatars3.githubusercontent.com">
 <link rel="dns-prefetch" href="https://github-cloud.s3.amazonaws.com">
 <link rel="dns-prefetch" href="https://user-images.githubusercontent.com/">
```

```html
 <link crossorigin="anonymous" href="https://assets-cdn.github.com/assets/frameworks-
2d2d4c150f7000385741c6b992b302689ecd172246c6428904e0813be9bceca6.css" media="all" rel="stylesheet" />
 <link crossorigin="anonymous" href="https://assets-cdn.github.com/assets/github-
0522ae8d3b3bdc841d2f91f90efd5f1fd9040d910905674cd134ced43a6dfea6.css" media="all" rel="stylesheet" />
```

```html
 <link crossorigin="anonymous" href="https://assets-cdn.github.com/assets/site-
cfab053e93f0e27f4c63d4ff6b7957bd25f711667fe678e747f8a4d88c47b38d.css" media="all" rel="stylesheet" />
```

```html
<meta name="viewport" content="width=device-width">
```

```html
<title>cryptacular/NOTICE at master  vt-middleware/cryptacular  GitHub</title>
<link rel="search" type="application/opensearchdescription+xml" href="/opensearch.xml" title="GitHub">
<link rel="fluid-icon" href="https://github.com/fluidicon.png" title="GitHub">
<meta property="fb:app_id" content="1401488693436528">
```

```html
  <meta content="https://avatars7.githubusercontent.com/u/6122907?v=4&s=400" property="og:image" /><meta
content="GitHub" property="og:site_name" /><meta content="object" property="og:type" /><meta content="vt-
middleware/cryptacular" property="og:title" /><meta content="https://github.com/vt-middleware/cryptacular"
property="og:url" /><meta content="cryptacular - The friendly complement to the BouncyCastle crypto API for
Java." property="og:description" />
```

```
<link rel="assets" href="https://assets-cdn.github.com/">

<meta name="pjax-timeout" content="1000">

<meta name="request-id" content="E0E4:26F16:12A5AE6:1D11801:596C7978" data-pjax-transient>


<meta name="selected-link" value="repo_source" data-pjax-transient>

<meta name="google-site-verification" content="KT5gs8h0wvaagLKAVWq8bbeNwnZZK1r1XQysX3xurLU">
<meta name="google-site-verification" content="ZzhVyEFwb7w3e0-uOTltm8Jsck2F5StVihD0exw2fsA">
  <meta name="google-analytics" content="UA-3769691-2">

<meta content="collector.githubapp.com" name="octolytics-host" /><meta content="github" name="octolytics-app-
id" /><meta content="https://collector.githubapp.com/github-external/browser_event" name="octolytics-event-url"
/><meta content="E0E4:26F16:12A5AE6:1D11801:596C7978" name="octolytics-dimension-request_id" /><meta
content="sea" name="octolytics-dimension-region_edge" /><meta content="iad" name="octolytics-dimension-
region_render" />
<meta content="/<user-name>/<repo-name>/blob/show" data-pjax-transient="true" name="analytics-location" />




<meta class="js-ga-set" name="dimension1" content="Logged Out">




    <meta name="hostname" content="github.com">
<meta name="user-login" content="">


    <meta name="expected-hostname" content="github.com">
  <meta name="js-proxy-site-detection-payload"
content="N2ZhMjk0NTA4MjI1NmZhYTVlNzM5NzVjZmFkOWY2NGFkNmMxYzcyMGViNzAzZGQxMGMzZ
mJhZDQ3YWZiZTI0OHx7InJlbW90ZV9hZGRyZXNzIjoiMTEwLjIwLjIyMC4xMzUiLCJyZXF1ZXN0X2lkIjoiR
TBFNDoyNkYxNjoxMkE1QUU2OjFEMTE4MDE6NTk2Qzc5NzgiLCJ0aW1lc3RhbXAiOjE1MDAyODEyMDgsI
mhvc3QiOiJnaXRodWIuY29tIn0=">


<meta name="html-safe-nonce" content="5aa226b80a18dc40894e1d405e4eb31cfca7d616">

<meta http-equiv="x-pjax-version" content="f682644ce1bb9629b9d9d9bedf64801b">


    <link href="https://github.com/vt-middleware/cryptacular/commits/master.atom" rel="alternate" title="Recent
Commits to cryptacular:master" type="application/atom+xml">

<meta name="description" content="cryptacular - The friendly complement to the BouncyCastle crypto API for
```

Java.">
 <meta name="go-import" content="github.com/vt-middleware/cryptacular git https://github.com/vt-middleware/cryptacular.git">

 <meta content="6122907" name="octolytics-dimension-user_id" /><meta content="vt-middleware" name="octolytics-dimension-user_login" /><meta content="15714989" name="octolytics-dimension-repository_id" /><meta content="vt-middleware/cryptacular" name="octolytics-dimension-repository_nwo" /><meta content="true" name="octolytics-dimension-repository_public" /><meta content="false" name="octolytics-dimension-repository_is_fork" /><meta content="15714989" name="octolytics-dimension-repository_network_root_id" /><meta content="vt-middleware/cryptacular" name="octolytics-dimension-repository_network_root_nwo" /><meta content="false" name="octolytics-dimension-repository_explore_github_marketplace_ci_cta_shown" />

  <link rel="canonical" href="https://github.com/vt-middleware/cryptacular/blob/master/NOTICE" data-pjax-transient>

 <meta name="browser-stats-url" content="https://api.github.com/_private/browser/stats">

 <meta name="browser-errors-url" content="https://api.github.com/_private/browser/errors">

 <link rel="mask-icon" href="https://assets-cdn.github.com/pinned-octocat.svg" color="#000000">
 <link rel="icon" type="image/x-icon" href="https://assets-cdn.github.com/favicon.ico">

<meta name="theme-color" content="#1e2327">

</head>

<body class="logged-out env-production page-blob">

<div class="position-relative js-header-wrapper ">
  Skip to content
  <div id="js-pjax-loader-bar" class="pjax-loader-bar"><div class="progress"></div></div>

    <header class="site-header js-details-container Details" role="banner">
 <div class="site-nav-container">

```html
<a class="header-logo-invertocat" href="https://github.com/" aria-label="Homepage" data-ga-click="(Logged out)
Header, go to homepage, icon:logo-wordmark">
    <svg aria-hidden="true" class="octicon octicon-mark-github" height="32" version="1.1" viewBox="0 0 16 16"
width="32"><path fill-rule="evenodd" d="M8 0C3.58 0 0 3.58 0 8c0 3.54 2.29 6.53 5.47 7.59.4.07.55-.17.55-.38 0-
.19-.01-.82-.01-1.49-2.01.37-2.53-.49-2.69-.94-.09-.23-.48-.94-.82-1.13-.28-.15-.68-.52-.01-.53.63-.01 1.08.58
1.23.82.72 1.21 1.87.87 2.33.66.07-.52.28-.87.51-1.07-1.78-.2-3.64-.89-3.64-3.95 0-.87.31-1.59.82-2.15-.08-.2-.36-
1.02.08-2.12 0 0 .67-.21 2.2.82.64-.18 1.32-.27 2-.27.68 0 1.36.09 2 .27 1.53-1.04 2.2-.82 2.2-.82.44 1.1.16 1.92.08
2.12.51.56.82 1.27.82 2.15 0 3.07-1.87 3.75-3.65 3.95.29.25.54.73.54 1.48 0 1.07-.01 1.93-.01 2.2 0
.21.15.46.55.38A8.013 8.013 0 0 0 16 8c0-4.42-3.58-8-8-8z"/></svg>
  </a>

  <button class="btn-link float-right site-header-toggle js-details-target" type="button" aria-label="Toggle
navigation" aria-expanded="false">
    <svg aria-hidden="true" class="octicon octicon-three-bars" height="24" version="1.1" viewBox="0 0 12 16"
width="18"><path fill-rule="evenodd" d="M11.41 9H.59C0 9 0 8.59 0 8c0-.59 0-1 .59-1H11.4c.59 0 .59.41.59 1 0
.59 0 1-.59 1h.01zm0-4H.59C0 5 0 4.59 0 4c0-.59 0-1 .59-1H11.4c.59 0 .59.41.59 1 0 .59 0 1-.59 1h.01zM.59
11H11.4c.59 0 .59.41.59 1 0 .59 0 1-.59 1H.59C0 13 0 12.59 0 12c0-.59 0-1 .59-1z"/></svg>
  </button>

  <div class="site-header-menu">
    <nav class="site-header-nav">
      <a href="/features" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu -
item:features" data-selected-links="/features /features/code-review /features/project-management
/features/integrations /features">
        Features
</a>      <a href="/business" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu
- item:business" data-selected-links="/business /business/security /business/customers /business">
        Business
</a>      <a href="/explore" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu -
item:explore" data-selected-links="/explore /trending /trending/developers /stars /integrations
/integrations/feature/code /integrations/feature/collaborate /integrations/feature/ship showcases showcases_search
showcases_landing /explore">
        Explore
</a>        <a href="/marketplace" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav
menu - item:marketplace" data-selected-links=" /marketplace">
          Marketplace
</a>      <a href="/pricing" class="js-selected-navigation-item nav-item" data-ga-click="Header, click, Nav menu -
item:pricing" data-selected-links="/pricing /pricing/developer /pricing/team /pricing/business-hosted
/pricing/business-enterprise /pricing">
        Pricing
</a>      </nav>

    <div class="site-header-actions">
      <div class="header-search scoped-search site-scoped-search js-site-search" role="search">
 <!-- "" --><!-- </textarea></xmp> --></option></form><form accept-charset="UTF-8" action="/vt-
middleware/cryptacular/search" class="js-site-search-form" data-scoped-search-url="/vt-
middleware/cryptacular/search" data-unscoped-search-url="/search" method="get"><div
style="margin:0;padding:0;display:inline"><input name="utf8" type="hidden" value="&#x2713;" /></div>
```

```
<label class="form-control header-search-wrapper js-chromeless-input-container">
    <a href="/vt-middleware/cryptacular/blob/master/NOTICE" class="header-search-scope no-underline">This
repository</a>
  <input type="text"
  class="form-control header-search-input js-site-search-focus js-site-search-field is-clearable"
  data-hotkey="s"
  name="q"
  value=""
  placeholder="Search"
  aria-label="Search this repository"
  data-unscoped-placeholder="Search GitHub"
  data-scoped-placeholder="Search"
  autocapitalize="off">
  <input type="hidden" class="js-site-search-type-field" name="type" >
 </label>
</form></div>


    <a class="text-bold site-header-link" href="/login?return_to=%2Fvt-
middleware%2Fcryptacular%2Fblob%2Fmaster%2FNOTICE" data-ga-click="(Logged out) Header, clicked Sign
in, text:sign-in">Sign in</a>
      <span class="text-gray">or</span>
      <a class="text-bold site-header-link" href="/join?source=header-repo" data-ga-click="(Logged out) Header,
clicked Sign up, text:sign-up">Sign up</a>
   </div>
  </div>
 </div>
</header>


 </div>

 <div id="start-of-content" class="show-on-focus"></div>

  <div id="js-flash-container">
</div>



 <div role="main">
    <div itemscope itemtype="http://schema.org/SoftwareSourceCode">
  <div id="js-repo-pjax-container" data-pjax-container>
```

```
<div class="pagehead repohead instapaper_ignore readability-menu experiment-repo-nav">
  <div class="container repohead-details-container">

    <ul class="pagehead-actions">
<li>
    <a href="/login?return_to=%2Fvt-middleware%2Fcryptacular"
  class="btn btn-sm btn-with-count tooltipped tooltipped-n"
  aria-label="You must be signed in to watch a repository" rel="nofollow">
    <svg aria-hidden="true" class="octicon octicon-eye" height="16" version="1.1" viewBox="0 0 16 16"
width="16"><path fill-rule="evenodd" d="M8.06 2C3 2 0 8 0 8s3 6 8.06 6C13 14 16 8 16 8s-3-6-7.94-6zM8 12c-
2.2 0-4-1.78-4-4 0-2.2 1.8-4 4-4 2.22 0 4 1.8 4 4 0 2.22-1.78 4-4 4zm2-4c0 1.11-.89 2-2 2-1.11 0-2-.89-2-2 0-
1.11.89-2 2-2 1.11 0 2 .89 2 2z"/></svg>
  Watch
  </a>
<a class="social-count" href="/vt-middleware/cryptacular/watchers"
    aria-label="9 users are watching this repository">
  9
  </a>

  </li>

  <li>
    <a href="/login?return_to=%2Fvt-middleware%2Fcryptacular"
  class="btn btn-sm btn-with-count tooltipped tooltipped-n"
  aria-label="You must be signed in to star a repository" rel="nofollow">
    <svg aria-hidden="true" class="octicon octicon-star" height="16" version="1.1" viewBox="0 0 14 16"
width="14"><path fill-rule="evenodd" d="M14 6l-4.9-.64L7 1 4.9 5.36 0 6l3.6 3.26L2.67 14 7 11.67 11.33 14l-.93-
4.74z"/></svg>
  Star
  </a>

  <a class="social-count js-social-count" href="/vt-middleware/cryptacular/stargazers"
    aria-label="15 users starred this repository">
  15
  </a>

  </li>

  <li>
    <a href="/login?return_to=%2Fvt-middleware%2Fcryptacular"
    class="btn btn-sm btn-with-count tooltipped tooltipped-n"
    aria-label="You must be signed in to fork a repository" rel="nofollow">
    <svg aria-hidden="true" class="octicon octicon-repo-forked" height="16" version="1.1" viewBox="0 0 10 16"
width="10"><path fill-rule="evenodd" d="M8 1a1.993 1.993 0 0 0-1 3.72V6L5 8 3 6V4.72A1.993 1.993 0 0 0 2
1a1.993 1.993 0 0 0-1 3.72V6.5l3 3v1.78A1.993 1.993 0 0 0 5 15a1.993 1.993 0 0 0 1-3.72V9.5l3-3V4.72A1.993
1.993 0 0 0 8 1zM2 4.2C1.34 4.2.8 3.65.8 3c0-.66.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2zm3 10c-
.66 0-1.2-.55-1.2-1.2 0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2zm3-10c-.66 0-1.2-.55-1.2-1.2 0-
```

```
.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2z"/></svg>
     Fork
    </a>

  <a href="/vt-middleware/cryptacular/network" class="social-count"
    aria-label="1 user forked this repository">
    1
  </a>
 </li>
</ul>

    <h1 class="public ">
 <svg aria-hidden="true" class="octicon octicon-repo" height="16" version="1.1" viewBox="0 0 12 16"
width="12"><path fill-rule="evenodd" d="M4 9H3V8h1v1zm0-3H3v1h1V6zm0-2H3v1h1V4zm0-
2H3v1h1V2zm8-1v12c0 .55-.45 1-1 1H6v2l-1.5-1.5L3 16v-2H1c-.55 0-1-.45-1-1V1c0-.55.45-1 1-1h10c.55 0 1 .45
1 1zm-1 10H1v2h2v-1h3v1h5v-2zm0-10H2v9h9V1z"/></svg>
 <span class="author" itemprop="author"><a href="/vt-middleware" class="url fn" rel="author">vt-
middleware</a></span><!--
--><span class="path-divider">/</span><!--
--><strong itemprop="name"><a href="/vt-middleware/cryptacular" data-pjax="#js-repo-pjax-
container">cryptacular</a></strong>

</h1>

    </div>
    <div class="container">

<nav class="reponav js-repo-nav js-sidenav-container-pjax"
    itemscope
    itemtype="http://schema.org/BreadcrumbList"
    role="navigation"
    data-pjax="#js-repo-pjax-container">

 <span itemscope itemtype="http://schema.org/ListItem" itemprop="itemListElement">
  <a href="/vt-middleware/cryptacular" class="js-selected-navigation-item selected reponav-item" data-hotkey="g
c" data-selected-links="repo_source repo_downloads repo_commits repo_releases repo_tags repo_branches /vt-
middleware/cryptacular" itemprop="url">
    <svg aria-hidden="true" class="octicon octicon-code" height="16" version="1.1" viewBox="0 0 14 16"
width="14"><path fill-rule="evenodd" d="M9.5 3L8 4.5 11.5 8 8 11.5 9.5 13 14 8 9.5 3zm-5 0L0 8l4.5 5L6 11.5
2.5 8 6 4.5 4.5 3z"/></svg>
    <span itemprop="name">Code</span>
    <meta itemprop="position" content="1">
</a>  </span>

  <span itemscope itemtype="http://schema.org/ListItem" itemprop="itemListElement">
    <a href="/vt-middleware/cryptacular/issues" class="js-selected-navigation-item reponav-item" data-hotkey="g i"
data-selected-links="repo_issues repo_labels repo_milestones /vt-middleware/cryptacular/issues" itemprop="url">
      <svg aria-hidden="true" class="octicon octicon-issue-opened" height="16" version="1.1" viewBox="0 0 14 16"
```

```
width="14"><path fill-rule="evenodd" d="M7 2.3c3.14 0 5.7 2.56 5.7 5.7s-2.56 5.7-5.7 5.7A5.71 5.71 0 0 1 1.3
8c0-3.14 2.56-5.7 5.7-5.7zM7 1C3.14 1 0 4.14 0 8s3.14 7 7 7 7-3.14 7-7-3.14-7-7-7zm1 3H6v5h2V4zm0
6H6v2h2v-2z"/></svg>
   <span itemprop="name">Issues</span>
   <span class="Counter">4</span>
   <meta itemprop="position" content="2">
</a>   </span>


<span itemscope itemtype="http://schema.org/ListItem" itemprop="itemListElement">
  <a href="/vt-middleware/cryptacular/pulls" class="js-selected-navigation-item reponav-item" data-hotkey="g p"
data-selected-links="repo_pulls /vt-middleware/cryptacular/pulls" itemprop="url">
   <svg aria-hidden="true" class="octicon octicon-git-pull-request" height="16" version="1.1" viewBox="0 0 12
16" width="12"><path fill-rule="evenodd" d="M11 11.28V5c-.03-.78-.34-1.47-.94-2.06C9.46 2.35 8.78 2.03 8
2H7V0L4 3l3 3V4h1c.27.02.48.11.69.31.21.2.3.42.31.69v6.28A1.993 1.993 0 0 0 10 15a1.993 1.993 0 0 0 1-
3.72zm-1 2.92c-.66 0-1.2-.55-1.2-1.2 0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2zM4 3c0-1.11-
.89-2-2-2a1.993 1.993 0 0 0-1 3.72v6.56A1.993 1.993 0 0 0 2 15a1.993 1.993 0 0 0 1-3.72V4.72c.59-.34 1-.98 1-
1.72zm-.8 10c0 .66-.55 1.2-1.2 1.2-.65 0-1.2-.55-1.2-1.2 0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2zM2 4.2C1.34 4.2.8
3.65.8 3c0-.65.55-1.2 1.2-1.2.65 0 1.2.55 1.2 1.2 0 .65-.55 1.2-1.2 1.2z"/></svg>
   <span itemprop="name">Pull requests</span>
   <span class="Counter">1</span>
   <meta itemprop="position" content="3">
</a>   </span>


  <a href="/vt-middleware/cryptacular/projects" class="js-selected-navigation-item reponav-item" data-selected-
links="repo_projects new_repo_project repo_project /vt-middleware/cryptacular/projects">
   <svg aria-hidden="true" class="octicon octicon-project" height="16" version="1.1" viewBox="0 0 15 16"
width="15"><path fill-rule="evenodd" d="M10 12h3V2h-3v10zm-4-2h3V2H6v8zm-4 4h3V2H2v12zm-1-
1h13V1H1v14zM14 0H1a1 1 0 0 0-1 1v14a1 1 0 0 0 1 1h13a1 1 0 0 0 1-1V1a1 1 0 0 0-1-1z"/></svg>
   Projects
   <span class="Counter" >0</span>
</a>



  <div class="reponav-dropdown js-menu-container">
   <button type="button" class="btn-link reponav-item reponav-dropdown js-menu-target " data-no-toggle aria-
expanded="false" aria-haspopup="true">
   Insights
   <svg aria-hidden="true" class="octicon octicon-triangle-down v-align-middle text-gray" height="11"
version="1.1" viewBox="0 0 12 16" width="8"><path fill-rule="evenodd" d="M0 5l6 6 6-6z"/></svg>
   </button>
   <div class="dropdown-menu-content js-menu-content">
   <div class="dropdown-menu dropdown-menu-sw">
    <a class="dropdown-item" href="/vt-middleware/cryptacular/pulse" data-skip-pjax>
    <svg aria-hidden="true" class="octicon octicon-pulse" height="16" version="1.1" viewBox="0 0 14 16"
width="14"><path fill-rule="evenodd" d="M11.5 8L8.8 5.4 6.6 8.5 5.5 1.6 2.38 8H0v2h3.6l.9-1.8.9 5.4L9 8.5l1.6
1.5H14V8z"/></svg>
     Pulse
     </a>
```

```html
          <a class="dropdown-item" href="/vt-middleware/cryptacular/graphs" data-skip-pjax>
            <svg aria-hidden="true" class="octicon octicon-graph" height="16" version="1.1" viewBox="0 0 16 16"
width="16"><path fill-rule="evenodd" d="M16 14v1H0V0h1v14h15zM5 13H3V8h2v5zm4 0H7V3h2v10zm4 0h-
2V6h2v7z"/></svg>
            Graphs
          </a>
        </div>
      </div>
    </div>
</nav>


        </div>
      </div>

<div class="container new-discussion-timeline experiment-repo-nav">
 <div class="repository-content">


 <a href="/vt-middleware/cryptacular/blob/c26911e3cd28497ce9daa3ce682e09cb2d1d8688/NOTICE" class="d-
none js-permalink-shortcut" data-hotkey="y">Permalink</a>

 <!-- blob contrib key: blob_contributors:v21:f5054b9e46039e0ad937c69e6151b7d4 -->

 <div class="file-navigation js-zeroclipboard-container">

<div class="select-menu branch-select-menu js-menu-container js-select-menu float-left">
 <button class=" btn btn-sm select-menu-button js-menu-target css-truncate" data-hotkey="w"

  type="button" aria-label="Switch branches or tags" aria-expanded="false" aria-haspopup="true">
    <i>Branch:</i>
    <span class="js-select-button css-truncate-target">master</span>
 </button>

 <div class="select-menu-modal-holder js-menu-content js-navigation-container" data-pjax>

  <div class="select-menu-modal">
   <div class="select-menu-header">
    <svg aria-label="Close" class="octicon octicon-x js-menu-close" height="16" role="img" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M7.48 8l3.75 3.75-1.48 1.48L6 9.48l-3.75 3.75-
1.48-1.48L4.52 8 .77 4.25l1.48-1.48L6 6.52l3.75-3.75 1.48 1.48z"/></svg>
     <span class="select-menu-title">Switch branches/tags</span>
   </div>

   <div class="select-menu-filters">
    <div class="select-menu-text-filter">
     <input type="text" aria-label="Filter branches/tags" id="context-commitish-filter-field" class="form-control
js-filterable-field js-navigation-enable" placeholder="Filter branches/tags">
    </div>
```

```
            <div class="select-menu-tabs">
              <ul>
                <li class="select-menu-tab">
                  <a href="#" data-tab-filter="branches" data-filter-placeholder="Filter branches/tags" class="js-select-menu-
tab" role="tab">Branches</a>
                </li>
                <li class="select-menu-tab">
                  <a href="#" data-tab-filter="tags" data-filter-placeholder="Find a tag" class="js-select-menu-tab"
role="tab">Tags</a>
                </li>
              </ul>
            </div>
          </div>

          <div class="select-menu-list select-menu-tab-bucket js-select-menu-tab-bucket" data-tab-filter="branches"
role="menu">

            <div data-filterable-for="context-commitish-filter-field" data-filterable-type="substring">


              <a class="select-menu-item js-navigation-item js-navigation-open "
                href="/vt-middleware/cryptacular/blob/ISSUE-31+32/NOTICE"
                data-name="ISSUE-31+32"
                data-skip-pjax="true"
                rel="nofollow">
                <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
                <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
                  ISSUE-31+32
                </span>
              </a>
              <a class="select-menu-item js-navigation-item js-navigation-open "
                href="/vt-middleware/cryptacular/blob/gh-pages/NOTICE"
                data-name="gh-pages"
                data-skip-pjax="true"
                rel="nofollow">
                <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
                <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
                  gh-pages
                </span>
              </a>
              <a class="select-menu-item js-navigation-item js-navigation-open selected"
                href="/vt-middleware/cryptacular/blob/master/NOTICE"
                data-name="master"
                data-skip-pjax="true"
                rel="nofollow">
                <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
```

```
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
      <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
        master
      </span>
    </a>
    <a class="select-menu-item js-navigation-item js-navigation-open "
      href="/vt-middleware/cryptacular/blob/v1.1/NOTICE"
      data-name="v1.1"
      data-skip-pjax="true"
      rel="nofollow">
      <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
      <span class="select-menu-item-text css-truncate-target js-select-menu-filter-text">
        v1.1
      </span>
    </a>
  </div>

    <div class="select-menu-no-results">Nothing to show</div>
  </div>

  <div class="select-menu-list select-menu-tab-bucket js-select-menu-tab-bucket" data-tab-filter="tags">
    <div data-filterable-for="context-commitish-filter-field" data-filterable-type="substring">


    <a class="select-menu-item js-navigation-item js-navigation-open "
      href="/vt-middleware/cryptacular/tree/v1.2.1/NOTICE"
      data-name="v1.2.1"
      data-skip-pjax="true"
      rel="nofollow">
      <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
      <span class="select-menu-item-text css-truncate-target" title="v1.2.1">
        v1.2.1
      </span>
    </a>
    <a class="select-menu-item js-navigation-item js-navigation-open "
      href="/vt-middleware/cryptacular/tree/v1.2.0/NOTICE"
      data-name="v1.2.0"
      data-skip-pjax="true"
      rel="nofollow">
      <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
      <span class="select-menu-item-text css-truncate-target" title="v1.2.0">
        v1.2.0
      </span>
    </a>
    <a class="select-menu-item js-navigation-item js-navigation-open "
```

```
href="/vt-middleware/cryptacular/tree/v1.1.2/NOTICE"
data-name="v1.1.2"
data-skip-pjax="true"
rel="nofollow">
<svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
<span class="select-menu-item-text css-truncate-target" title="v1.1.2">
 v1.1.2
</span>
</a>
<a class="select-menu-item js-navigation-item js-navigation-open "
href="/vt-middleware/cryptacular/tree/v1.1.1/NOTICE"
data-name="v1.1.1"
data-skip-pjax="true"
rel="nofollow">
<svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
<span class="select-menu-item-text css-truncate-target" title="v1.1.1">
 v1.1.1
</span>
</a>
<a class="select-menu-item js-navigation-item js-navigation-open "
href="/vt-middleware/cryptacular/tree/v1.1.0/NOTICE"
data-name="v1.1.0"
data-skip-pjax="true"
rel="nofollow">
<svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
<span class="select-menu-item-text css-truncate-target" title="v1.1.0">
 v1.1.0
</span>
</a>
<a class="select-menu-item js-navigation-item js-navigation-open "
href="/vt-middleware/cryptacular/tree/v1.0/NOTICE"
data-name="v1.0"
data-skip-pjax="true"
rel="nofollow">
<svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
<span class="select-menu-item-text css-truncate-target" title="v1.0">
 v1.0
</span>
</a>
<a class="select-menu-item js-navigation-item js-navigation-open "
href="/vt-middleware/cryptacular/tree/v1.0-RC6/NOTICE"
data-name="v1.0-RC6"
data-skip-pjax="true"
rel="nofollow">
```

```
<svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
    <span class="select-menu-item-text css-truncate-target" title="v1.0-RC6">
      v1.0-RC6
    </span>
  </a>
  <a class="select-menu-item js-navigation-item js-navigation-open "
    href="/vt-middleware/cryptacular/tree/v1.0-RC4/NOTICE"
    data-name="v1.0-RC4"
    data-skip-pjax="true"
    rel="nofollow">
    <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
    <span class="select-menu-item-text css-truncate-target" title="v1.0-RC4">
      v1.0-RC4
    </span>
  </a>
  <a class="select-menu-item js-navigation-item js-navigation-open "
    href="/vt-middleware/cryptacular/tree/v1.0-RC3/NOTICE"
    data-name="v1.0-RC3"
    data-skip-pjax="true"
    rel="nofollow">
    <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
    <span class="select-menu-item-text css-truncate-target" title="v1.0-RC3">
      v1.0-RC3
    </span>
  </a>
  <a class="select-menu-item js-navigation-item js-navigation-open "
    href="/vt-middleware/cryptacular/tree/v1.0-RC2/NOTICE"
    data-name="v1.0-RC2"
    data-skip-pjax="true"
    rel="nofollow">
    <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
    <span class="select-menu-item-text css-truncate-target" title="v1.0-RC2">
      v1.0-RC2
    </span>
  </a>
  <a class="select-menu-item js-navigation-item js-navigation-open "
    href="/vt-middleware/cryptacular/tree/v1.0-RC1/NOTICE"
    data-name="v1.0-RC1"
    data-skip-pjax="true"
    rel="nofollow">
    <svg aria-hidden="true" class="octicon octicon-check select-menu-item-icon" height="16" version="1.1"
viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M12 5l-8 8-4-4 1.5-1.5L4 10l6.5-6.5z"/></svg>
    <span class="select-menu-item-text css-truncate-target" title="v1.0-RC1">
      v1.0-RC1
```

```
        </span>
      </a>
    </div>

    <div class="select-menu-no-results">Nothing to show</div>
  </div>

 </div>
 </div>
</div>

  <div class="BtnGroup float-right">
    <a href="/vt-middleware/cryptacular/find/master"
       class="js-pjax-capture-input btn btn-sm BtnGroup-item"
       data-pjax
       data-hotkey="t">
     Find file
    </a>
    <button aria-label="Copy file path to clipboard" class="js-zeroclipboard btn btn-sm BtnGroup-item tooltipped
tooltipped-s" data-copied-hint="Copied!" type="button">Copy path</button>
  </div>
  <div class="breadcrumb js-zeroclipboard-target">
    <span class="repo-root js-repo-root"><span class="js-path-segment"><a href="/vt-
middleware/cryptacular"><span>cryptacular</span></a></span></span><span class="separator">/</span><strong
class="final-path">NOTICE</strong>
  </div>
 </div>




 <div class="commit-tease">
   <span class="float-right">
    <a class="commit-tease-sha" href="/vt-
middleware/cryptacular/commit/6dd6f199ac3ecc3b4c5aef9e04be3bbe265a30a1" data-pjax>
      6dd6f19
    </a>
    <relative-time datetime="2017-07-06T22:28:36Z">Jul 7, 2017</relative-time>
   </span>
   <div>
    <img alt="@dfish3r" class="avatar" height="20"
src="https://avatars6.githubusercontent.com/u/1051499?v=4&s=40" width="20" />
     <a href="/dfish3r" class="user-mention" rel="contributor">dfish3r</a>
      <a href="/vt-middleware/cryptacular/commit/6dd6f199ac3ecc3b4c5aef9e04be3bbe265a30a1"
class="message" data-pjax="true" title="Update year in notice.">Update year in notice.</a>
   </div>

  <div class="commit-tease-contributors">
    <button type="button" class="btn-link muted-link contributors-toggle" data-facebox="#blob_contributors_box">
```

```html
      <strong>1</strong>
      contributor
    </button>

  </div>

  <div id="blob_contributors_box" style="display:none">
    <h2 class="facebox-header" data-facebox-id="facebox-header">Users who have contributed to this file</h2>
    <ul class="facebox-user-list" data-facebox-id="facebox-description">
      <li class="facebox-user-list-item">
        <img alt="@dfish3r" height="24" src="https://avatars4.githubusercontent.com/u/1051499?v=4&s=48" width="24" />
        <a href="/dfish3r">dfish3r</a>
      </li>
    </ul>
  </div>
</div>

<div class="file">
  <div class="file-header">
<div class="file-actions">

  <div class="BtnGroup">
    <a href="/vt-middleware/cryptacular/raw/master/NOTICE" class="btn btn-sm BtnGroup-item" id="raw-url">Raw</a>
      <a href="/vt-middleware/cryptacular/blame/master/NOTICE" class="btn btn-sm js-update-url-with-hash BtnGroup-item" data-hotkey="b">Blame</a>
    <a href="/vt-middleware/cryptacular/commits/master/NOTICE" class="btn btn-sm BtnGroup-item" rel="nofollow">History</a>
  </div>


    <button type="button" class="btn-octicon disabled tooltipped tooltipped-nw"
      aria-label="You must be signed in to make or propose changes">
      <svg aria-hidden="true" class="octicon octicon-pencil" height="16" version="1.1" viewBox="0 0 14 16" width="14"><path fill-rule="evenodd" d="M0 12v3h3l8-8-3-3-8 8zm3 2H1v-2h1v1h1v1zm10.3-9.3L12 6 9 3l1.3-1.3a.996.996 0 0 1 1.41 0l1.59 1.59c.39.39.39 1.02 0 1.41z"/></svg>
    </button>
    <button type="button" class="btn-octicon btn-octicon-danger disabled tooltipped tooltipped-nw"
      aria-label="You must be signed in to make or propose changes">
      <svg aria-hidden="true" class="octicon octicon-trashcan" height="16" version="1.1" viewBox="0 0 12 16" width="12"><path fill-rule="evenodd" d="M11 2H9c0-.55-.45-1-1-1H5c-.55 0-1 .45-1 1H2c-.55 0-1 .45-1 1v1c0 .55.45 1 1 1v9c0 .55.45 1 1 1h7c.55 0 1-.45 1-1V5c.55 0 1-.45 1-1V3c0-.55-.45-1-1-1zm-1 12H3V5h1v8h1V5h1v8h1V5h1v8h1V5h1v9zm1-10H2V3h9v1z"/></svg>
    </button>
  </div>

  <div class="file-info">
```

7 lines (5 sloc)

165 Bytes
</div>
</div>

<div itemprop="text" class="blob-wrapper data type-text">
<table class="highlight tab-size js-file-line-container" data-tab-size="8">
<tr>
<td id="L1" class="blob-num js-line-number" data-line-number="1"></td>
<td id="LC1" class="blob-code blob-code-inner js-file-line">Cryptacular Java Library</td>
</tr>
<tr>
<td id="L2" class="blob-num js-line-number" data-line-number="2"></td>
<td id="LC2" class="blob-code blob-code-inner js-file-line">Copyright (C) 2003-2017 Virginia Tech.</td>
</tr>
<tr>
<td id="L3" class="blob-num js-line-number" data-line-number="3"></td>
<td id="LC3" class="blob-code blob-code-inner js-file-line">All rights reserved.</td>
</tr>
<tr>
<td id="L4" class="blob-num js-line-number" data-line-number="4"></td>
<td id="LC4" class="blob-code blob-code-inner js-file-line">
</td>
</tr>
<tr>
<td id="L5" class="blob-num js-line-number" data-line-number="5"></td>
<td id="LC5" class="blob-code blob-code-inner js-file-line">This product includes software developed at</td>
</tr>
<tr>
<td id="L6" class="blob-num js-line-number" data-line-number="6"></td>
<td id="LC6" class="blob-code blob-code-inner js-file-line">Virginia Tech (http://www.vt.edu).</td>
</tr>
</table>

</div>

</div>

<button type="button" data-facebox="#jump-to-line" data-facebox-class="linejump" data-hotkey="l" class="d-none">Jump to Line</button>
<div id="jump-to-line" style="display:none">
<!-- '"` --><!-- </textarea></xmp> --></option></form><form accept-charset="UTF-8" action="" class="js-jump-to-line-form" method="get"><div style="margin:0;padding:0;display:inline"><input name="utf8" type="hidden" value="&#x2713;" /></div>

```
    <input class="form-control linejump-input js-jump-to-line-field" type="text" placeholder="Jump to line&hellip;"
aria-label="Jump to line" autofocus>
      <button type="submit" class="btn">Go</button>
</form>  </div>

 </div>
 <div class="modal-backdrop js-touch-events"></div>
</div>

   </div>
 </div>

 </div>


<div class="container site-footer-container">
 <div class="site-footer " role="contentinfo">
   <ul class="site-footer-links float-right">
      <li><a href="https://github.com/contact" data-ga-click="Footer, go to contact, text:contact">Contact
GitHub</a></li>
    <li><a href="https://developer.github.com" data-ga-click="Footer, go to api, text:api">API</a></li>
    <li><a href="https://training.github.com" data-ga-click="Footer, go to training, text:training">Training</a></li>
    <li><a href="https://shop.github.com" data-ga-click="Footer, go to shop, text:shop">Shop</a></li>
     <li><a href="https://github.com/blog" data-ga-click="Footer, go to blog, text:blog">Blog</a></li>
      <li><a href="https://github.com/about" data-ga-click="Footer, go to about, text:about">About</a></li>

   </ul>

   <a href="https://github.com" aria-label="Homepage" class="site-footer-mark" title="GitHub">
    <svg aria-hidden="true" class="octicon octicon-mark-github" height="24" version="1.1" viewBox="0 0 16 16"
width="24"><path fill-rule="evenodd" d="M8 0C3.58 0 0 3.58 0 8c0 3.54 2.29 6.53 5.47 7.59.4.07.55-.17.55-.38 0-
.19-.01-.82-.01-1.49-2.01.37-2.53-.49-2.69-.94-.09-.23-.48-.94-.82-1.13-.28-.15-.68-.52-.01-.53.63-.01 1.08.58
1.23.82.72 1.21 1.87.87 2.33.66.07-.52.28-.87.51-1.07-1.78-.2-3.64-.89-3.64-3.95 0-.87.31-1.59.82-2.15-.08-.2-.36-
1.02.08-2.12 0 0 .67-.21 2.2.82.64-.18 1.32-.27 2-.27.68 0 1.36.09 2 .27 1.53-1.04 2.2-.82 2.2-.82.44 1.1.16 1.92.08
2.12.51.56.82 1.27.82 2.15 0 3.07-1.87 3.75-3.65 3.95.29.25.54.73.54 1.48 0 1.07-.01 1.93-.01 2.2 0
.21.15.46.55.38A8.013 8.013 0 0 0 16 8c0-4.42-3.58-8-8-8z"/></svg>
</a>
    <ul class="site-footer-links">
    <li>&copy; 2017 <span title="0.10938s from unicorn-2925809464-f2ltq">GitHub</span>, Inc.</li>
      <li><a href="https://github.com/site/terms" data-ga-click="Footer, go to terms, text:terms">Terms</a></li>
      <li><a href="https://github.com/site/privacy" data-ga-click="Footer, go to privacy,
text:privacy">Privacy</a></li>
      <li><a href="https://github.com/security" data-ga-click="Footer, go to security,
text:security">Security</a></li>
      <li><a href="https://status.github.com/" data-ga-click="Footer, go to status, text:status">Status</a></li>
      <li><a href="https://help.github.com" data-ga-click="Footer, go to help, text:help">Help</a></li>
   </ul>
 </div>
```

```
</div>


  <div id="ajax-error-message" class="ajax-error-message flash flash-error">
    <svg aria-hidden="true" class="octicon octicon-alert" height="16" version="1.1" viewBox="0 0 16 16"
width="16"><path fill-rule="evenodd" d="M8.865 1.52c-.18-.31-.51-.5-.87-.5s-.69.19-.87.5L.275 13.5c-.18.31-
.18.69 0 1 .19.31.52.5.87.5h13.7c.36 0 .69-.19.86-.5.17-.31.18-.69.01-1L8.865 1.52zM8.995 13h-2v-2h2v2zm0-3h-
2V6h2v4z"/></svg>
    <button type="button" class="flash-close js-flash-close js-ajax-error-dismiss" aria-label="Dismiss error">
      <svg aria-hidden="true" class="octicon octicon-x" height="16" version="1.1" viewBox="0 0 12 16"
width="12"><path fill-rule="evenodd" d="M7.48 8l3.75 3.75-1.48 1.48L6 9.48l-3.75 3.75-1.48-1.48L4.52 8 .77
4.25l1.48-1.48L6 6.52l3.75-3.75 1.48 1.48z"/></svg>
    </button>
    You can't perform that action at this time.
  </div>


  <script crossorigin="anonymous" src="https://assets-cdn.github.com/assets/compat-
91f98c37fc84eac24836eec2567e9912742094369a04c4eba6e3cd1fa18902d9.js"></script>
  <script crossorigin="anonymous" src="https://assets-cdn.github.com/assets/frameworks-
f84bb87b149685d1e6c6f057ee324f2cd496e677f5a359a8b5db853313bb83e6.js"></script>

  <script async="async" crossorigin="anonymous" src="https://assets-cdn.github.com/assets/github-
13fa3aa50ac8f9fa9a7d198f0cd13b0905775d39446ad076d17d8f74a998438a.js"></script>




  <div class="js-stale-session-flash stale-session-flash flash flash-warn flash-banner d-none">
    <svg aria-hidden="true" class="octicon octicon-alert" height="16" version="1.1" viewBox="0 0 16 16"
width="16"><path fill-rule="evenodd" d="M8.865 1.52c-.18-.31-.51-.5-.87-.5s-.69.19-.87.5L.275 13.5c-.18.31-
.18.69 0 1 .19.31.52.5.87.5h13.7c.36 0 .69-.19.86-.5.17-.31.18-.69.01-1L8.865 1.52zM8.995 13h-2v-2h2v2zm0-3h-
2V6h2v4z"/></svg>
    <span class="signed-in-tab-flash">You signed in with another tab or window. <a href="">Reload</a> to refresh
your session.</span>
    <span class="signed-out-tab-flash">You signed out in another tab or window. <a href="">Reload</a> to refresh
your session.</span>
  </div>
  <div class="facebox" id="facebox" style="display:none;">
  <div class="facebox-popup">
    <div class="facebox-content" role="dialog" aria-labelledby="facebox-header" aria-describedby="facebox-
description">
    </div>
    <button type="button" class="facebox-close js-facebox-close" aria-label="Close modal">
      <svg aria-hidden="true" class="octicon octicon-x" height="16" version="1.1" viewBox="0 0 12 16"
width="12"><path fill-rule="evenodd" d="M7.48 8l3.75 3.75-1.48 1.48L6 9.48l-3.75 3.75-1.48-1.48L4.52 8 .77
4.25l1.48-1.48L6 6.52l3.75-3.75 1.48 1.48z"/></svg>
```

```
    </button>
  </div>
</div>


  </body>
</html>
```
 Copyright (c) 2004-2017 QOS.ch

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache XML Security for Java
Copyright 2000-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

```
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license.licensor;

import org.apache.lucene.util.BytesRef;
import org.apache.lucene.util.BytesRefIterator;
import org.elasticsearch.common.bytes.BytesReference;
import org.elasticsearch.common.hash.MessageDigests;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
```

```java
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.CryptUtils;
import org.elasticsearch.license.License;

import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.file.Files;
import java.nio.file.Path;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Signature;
import java.security.SignatureException;
import java.util.Base64;
import java.util.Collections;
import java.util.Map;

/**
 * Responsible for generating a license signature according to the signature spec and sign it with
 * the provided encrypted private key
 */
public class LicenseSigner {

    private static final int MAGIC_LENGTH = 13;
    private final Path publicKeyPath;
    private final Path privateKeyPath;

    public LicenseSigner(final Path privateKeyPath, final Path publicKeyPath) {
        this.publicKeyPath = publicKeyPath;
        this.privateKeyPath = privateKeyPath;
    }

    /**
     * Generates a signature for the {@code licenseSpec}. Signature structure:
     * <code>
     * | VERSION | MAGIC | PUB_KEY_DIGEST | SIGNED_LICENSE_CONTENT |
     * </code>
     *
     * @return a signed License
     */
    public License sign(License licenseSpec) throws IOException {
        XContentBuilder contentBuilder = XContentFactory.contentBuilder(XContentType.JSON);
        final Map<String, String> licenseSpecViewMode =
            Collections.singletonMap(License.LICENSE_SPEC_VIEW_MODE, "true");
        licenseSpec.toXContent(contentBuilder, new ToXContent.MapParams(licenseSpecViewMode));
```

```java
        final byte[] signedContent;
        final boolean preV4 = licenseSpec.version() < License.VERSION_CRYPTO_ALGORITHMS;
        try {
            final Signature rsa = Signature.getInstance("SHA512withRSA");
            PrivateKey decryptedPrivateKey =
CryptUtils.readEncryptedPrivateKey(Files.readAllBytes(privateKeyPath));
            rsa.initSign(decryptedPrivateKey);
            final BytesRefIterator iterator = BytesReference.bytes(contentBuilder).iterator();
            BytesRef ref;
            while((ref = iterator.next()) != null) {
                rsa.update(ref.bytes, ref.offset, ref.length);
            }
            signedContent = rsa.sign();
        } catch (InvalidKeyException
                | IOException
                | NoSuchAlgorithmException
                | SignatureException e) {
            throw new IllegalStateException(e);
        }
        final byte[] magic = new byte[MAGIC_LENGTH];
        SecureRandom random = new SecureRandom();
        random.nextBytes(magic);
        final byte[] publicKeyBytes = Files.readAllBytes(publicKeyPath);
        PublicKey publicKey = CryptUtils.readPublicKey(publicKeyBytes);
        final byte[] pubKeyFingerprint = preV4 ?
Base64.getEncoder().encode(CryptUtils.writeEncryptedPublicKey(publicKey)) :
                getPublicKeyFingerprint(publicKeyBytes);
        byte[] bytes = new byte[4 + 4 + MAGIC_LENGTH + 4 + pubKeyFingerprint.length + 4 +
signedContent.length];
        ByteBuffer byteBuffer = ByteBuffer.wrap(bytes);
        byteBuffer.putInt(licenseSpec.version())
                .putInt(magic.length)
                .put(magic)
                .putInt(pubKeyFingerprint.length)
                .put(pubKeyFingerprint)
                .putInt(signedContent.length)
                .put(signedContent);

        return License.builder()
                .fromLicenseSpec(licenseSpec, Base64.getEncoder().encodeToString(bytes))
                .build();
    }

    private byte[] getPublicKeyFingerprint(byte[] keyBytes) {
        MessageDigest sha256 = MessageDigests.sha256();
        sha256.update(keyBytes);
        return sha256.digest();
    }
```

}
0"0
 *H
0
+5Bj:h$Tfsr&avDgpi]2oR<Cizzz E(yhZ}TCjh~!P|499cq5b!Y{VX=,Vd ;g9hD`Xv8n|i) 34aYfsWBkuOo@
u@175/YmI4~
d&^w t<q^XQ^KG#|X?tV}f;5ENHgXyk9Mh/WM*D;D^]H`"
}W"o'"YY@9x"|`hNb>a3>8ZOSd}R4HxQU99dQ\nTxL?>*}QMB7fhIE952^hn=-m2-
pF8T_I^yT1=)dc=t*Ge^^f71cb~_^|5yZl&9LSdxR8XNUC1%^R;CiVIN~O}c%\'h|&UX;j{_#9/^/1H@_Wr$l,})v[#?
;m;`O'ZiB <Pu
OkH4jbr
WvF\w'#`)~Vx/|?{zDa
\| hGEi?#r~N
zm`nL|~1~'?/'>k?~gyk,b' _.41K??3d2kgb$7~QjlqN]j,Lr-9Vl34uNe+'4dpP.V}uvq4D}08LrTeg5Z+n@&+(P=-M"
[rw8baSjJaam+ +?b!u&2tW!IYi.fj@/ G#uZv_sgj2UXk.ſdbq7kT%h5'Nv5[D4!l@1
5}*`"DK[%N^(Af9l] 0X`M:c01uAm\jB/.;f
es.logger.level=INFO
log4j.rootLogger=${es.logger.level}, out

log4j.logger.org.apache.http=INFO, out
log4j.additivity.org.apache.http=false

log4j.logger.org.elasticsearch.license=TRACE

log4j.appender.out=org.apache.log4j.ConsoleAppender
log4j.appender.out.layout=org.apache.log4j.PatternLayout
log4j.appender.out.layout.conversionPattern=[%d{ISO8601}][%-5p][%-25c] %m%n
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor;

import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.license.DateUtils;
import org.elasticsearch.license.License;
import org.elasticsearch.license.LicenseVerifier;
import org.elasticsearch.test.ESTestCase;
import org.junit.After;
import org.junit.Before;

import java.nio.file.Files;
import java.nio.file.Path;

public class LicenseVerificationTests extends ESTestCase {

  protected Path pubKeyPath = null;

```java
    protected Path priKeyPath = null;

    @Before
    public void setup() throws Exception {
        pubKeyPath = getDataPath("/public.key");
        priKeyPath = getDataPath("/private.key");
    }

    @After
    public void cleanUp() {
        pubKeyPath = null;
        priKeyPath = null;
    }

    public void testGeneratedLicenses() throws Exception {
        final TimeValue fortyEightHours = TimeValue.timeValueHours(2 * 24);
        final License license =
            TestUtils.generateSignedLicense(fortyEightHours, pubKeyPath, priKeyPath);
        assertTrue(LicenseVerifier.verifyLicense(license, Files.readAllBytes(pubKeyPath)));
    }

    public void testLicenseTampering() throws Exception {
        final TimeValue twoHours = TimeValue.timeValueHours(2);
        License license = TestUtils.generateSignedLicense(twoHours, pubKeyPath, priKeyPath);

        final License tamperedLicense = License.builder()
            .fromLicenseSpec(license, license.signature())
            .expiryDate(license.expiryDate() + 10 * 24 * 60 * 60 * 1000L)
            .validate()
            .build();

        assertFalse(LicenseVerifier.verifyLicense(tamperedLicense, Files.readAllBytes(pubKeyPath)));
    }

    public void testRandomLicenseVerification() throws Exception {
        TestUtils.LicenseSpec licenseSpec = TestUtils.generateRandomLicenseSpec(
            randomIntBetween(License.VERSION_START, License.VERSION_CURRENT));
        License generatedLicense = generateSignedLicense(licenseSpec, pubKeyPath, priKeyPath);
        assertTrue(LicenseVerifier.verifyLicense(generatedLicense, Files.readAllBytes(pubKeyPath)));
    }

    private static License generateSignedLicense(
            TestUtils.LicenseSpec spec, Path pubKeyPath, Path priKeyPath) throws Exception {
        LicenseSigner signer = new LicenseSigner(priKeyPath, pubKeyPath);
        License.Builder builder = License.builder()
            .uid(spec.uid)
            .feature(spec.feature)
            .type(spec.type)
```

```java
                    .subscriptionType(spec.subscriptionType)
                    .issuedTo(spec.issuedTo)
                    .issuer(spec.issuer)
                    .maxNodes(spec.maxNodes);

            if (spec.expiryDate != null) {
                builder.expiryDate(DateUtils.endOfTheDay(spec.expiryDate));
            } else {
                builder.expiryDate(spec.expiryDateInMillis);
            }
            if (spec.issueDate != null) {
                builder.issueDate(DateUtils.beginningOfTheDay(spec.issueDate));
            } else {
                builder.issueDate(spec.issueDateInMillis);
            }
            builder.version(spec.version);
            return signer.sign(builder.build());
        }

}
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license.licensor;

import org.elasticsearch.common.Strings;
import org.elasticsearch.common.joda.DateMathParser;
import org.elasticsearch.common.joda.FormatDateTimeFormatter;
import org.elasticsearch.common.joda.Joda;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.common.xcontent.ToXContent;
import org.elasticsearch.common.xcontent.XContentBuilder;
import org.elasticsearch.common.xcontent.XContentFactory;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.DateUtils;
import org.elasticsearch.license.License;
import org.elasticsearch.test.ESTestCase;
import org.hamcrest.MatcherAssert;
import org.joda.time.format.DateTimeFormatter;

import java.io.IOException;
import java.nio.file.Path;
import java.util.UUID;

import static com.carrotsearch.randomizedtesting.RandomizedTest.randomBoolean;
import static com.carrotsearch.randomizedtesting.RandomizedTest.randomInt;
```

```java
import static com.carrotsearch.randomizedtesting.RandomizedTest.randomIntBetween;
import static org.elasticsearch.common.xcontent.XContentFactory.jsonBuilder;
import static org.elasticsearch.test.ESTestCase.randomFrom;
import static org.hamcrest.core.IsEqual.equalTo;

public class TestUtils {

    public static final String PUBLIC_KEY_RESOURCE = "/public.key";
    public static final String PRIVATE_KEY_RESOURCE = "/private.key";

    private static final FormatDateTimeFormatter formatDateTimeFormatter =
            Joda.forPattern("yyyy-MM-dd");
    private static final DateMathParser dateMathParser =
            new DateMathParser(formatDateTimeFormatter);
    private static final DateTimeFormatter dateTimeFormatter = formatDateTimeFormatter.printer();

    public static String dumpLicense(License license) throws Exception {
        XContentBuilder builder = XContentFactory.contentBuilder(XContentType.JSON);
        builder.startObject();
        builder.startObject("license");
        license.toInnerXContent(builder, ToXContent.EMPTY_PARAMS);
        builder.endObject();
        builder.endObject();
        return Strings.toString(builder);
    }

    public static String dateMathString(String time, final long now) {
        return dateTimeFormatter.print(dateMathParser.parse(time, () -> now));
    }

    public static long dateMath(String time, final long now) {
        return dateMathParser.parse(time, () -> now);
    }

    public static LicenseSpec generateRandomLicenseSpec(int version) {
        boolean datesInMillis = randomBoolean();
        long now = System.currentTimeMillis();
        String uid = UUID.randomUUID().toString();
        String issuer = "issuer__" + randomInt();
        String issuedTo = "issuedTo__" + randomInt();
        String type = version < License.VERSION_NO_FEATURE_TYPE ?
                randomFrom("subscription", "internal", "development") :
                randomFrom("basic", "silver", "dev", "gold", "platinum");
        final String subscriptionType;
        final String feature;
        if (version < License.VERSION_NO_FEATURE_TYPE) {
            subscriptionType = randomFrom("gold", "silver", "platinum");
            feature = "feature__" + randomInt();
```

```java
      } else {
         subscriptionType = null;
         feature = null;
      }
      int maxNodes = randomIntBetween(5, 100);
      if (datesInMillis) {
         long issueDateInMillis = dateMath("now", now);
         long expiryDateInMillis = dateMath("now+10d/d", now);
         return new LicenseSpec(
              version,
              uid,
              feature,
              issueDateInMillis,
              expiryDateInMillis,
              type,
              subscriptionType,
              issuedTo,
              issuer,
              maxNodes);
      } else {
         String issueDate = dateMathString("now", now);
         String expiryDate = dateMathString("now+10d/d", now);
         return new LicenseSpec(
              version,
              uid,
              feature,
              issueDate,
              expiryDate, type,
              subscriptionType,
              issuedTo,
              issuer,
              maxNodes);
      }
   }

   public static String generateLicenseSpecString(LicenseSpec licenseSpec) throws IOException {
      XContentBuilder licenses = jsonBuilder();
      licenses.startObject();
      licenses.startObject("license")
            .field("uid", licenseSpec.uid)
            .field("type", licenseSpec.type)
            .field("subscription_type", licenseSpec.subscriptionType)
            .field("issued_to", licenseSpec.issuedTo)
            .field("issuer", licenseSpec.issuer)
            .field("feature", licenseSpec.feature)
            .field("max_nodes", licenseSpec.maxNodes);

      if (licenseSpec.issueDate != null) {
```

```java
            licenses.field("issue_date", licenseSpec.issueDate);
        } else {
            licenses.field("issue_date_in_millis", licenseSpec.issueDateInMillis);
        }
        if (licenseSpec.expiryDate != null) {
            licenses.field("expiry_date", licenseSpec.expiryDate);
        } else {
            licenses.field("expiry_date_in_millis", licenseSpec.expiryDateInMillis);
        }
        licenses.field("version", licenseSpec.version);
        licenses.endObject();
        licenses.endObject();
        return Strings.toString(licenses);
    }

    public static void assertLicenseSpec(LicenseSpec spec, License license) {
        MatcherAssert.assertThat(license.uid(), equalTo(spec.uid));
        MatcherAssert.assertThat(license.issuedTo(), equalTo(spec.issuedTo));
        MatcherAssert.assertThat(license.issuer(), equalTo(spec.issuer));
        MatcherAssert.assertThat(license.type(), equalTo(spec.type));
        MatcherAssert.assertThat(license.maxNodes(), equalTo(spec.maxNodes));
        if (spec.issueDate != null) {
            MatcherAssert.assertThat(
                    license.issueDate(),
                    equalTo(DateUtils.beginningOfTheDay(spec.issueDate)));
        } else {
            MatcherAssert.assertThat(license.issueDate(), equalTo(spec.issueDateInMillis));
        }
        if (spec.expiryDate != null) {
            MatcherAssert.assertThat(
                    license.expiryDate(),
                    equalTo(DateUtils.endOfTheDay(spec.expiryDate)));
        } else {
            MatcherAssert.assertThat(license.expiryDate(), equalTo(spec.expiryDateInMillis));
        }
    }

    public static License generateSignedLicense(
            TimeValue expiryDuration, Path pubKeyPath, Path priKeyPath) throws Exception {
        long issue = System.currentTimeMillis();
        int version = ESTestCase.randomIntBetween(License.VERSION_START, License.VERSION_CURRENT);
        String type = version < License.VERSION_NO_FEATURE_TYPE ?
                randomFrom("subscription", "internal", "development") :
                randomFrom("trial", "basic", "silver", "dev", "gold", "platinum");
        final License.Builder builder = License.builder()
                .uid(UUID.randomUUID().toString())
                .expiryDate(issue + expiryDuration.getMillis())
                .issueDate(issue)
```

```
                .version(version)
                .type(type)
                .issuedTo("customer")
                .issuer("elasticsearch")
                .maxNodes(5);
        if (version == License.VERSION_START) {
            builder.subscriptionType(randomFrom("dev", "gold", "platinum", "silver"));
            builder.feature(ESTestCase.randomAlphaOfLength(10));
        }
        LicenseSigner signer = new LicenseSigner(priKeyPath, pubKeyPath);
        return signer.sign(builder.build());
    }

    public static class LicenseSpec {
        public final int version;
        public final String feature;
        public final String issueDate;
        public final long issueDateInMillis;
        public final String expiryDate;
        public final long expiryDateInMillis;
        public final String uid;
        public final String type;
        public final String subscriptionType;
        public final String issuedTo;
        public final String issuer;
        public final int maxNodes;

        public LicenseSpec(
                int version,
                String uid,
                String feature,
                long issueDateInMillis,
                long expiryDateInMillis,
                String type,
                String subscriptionType,
                String issuedTo,
                String issuer,
                int maxNodes) {
            this.version = version;
            this.feature = feature;
            this.issueDateInMillis = issueDateInMillis;
            this.issueDate = null;
            this.expiryDateInMillis = expiryDateInMillis;
            this.expiryDate = null;
            this.uid = uid;
            this.type = type;
            this.subscriptionType = subscriptionType;
            this.issuedTo = issuedTo;
```

```java
        this.issuer = issuer;
        this.maxNodes = maxNodes;
    }

    public LicenseSpec(
            int version,
            String uid,
            String feature,
            String issueDate,
            String expiryDate,
            String type,
            String subscriptionType,
            String issuedTo,
            String issuer,
            int maxNodes) {
        this.version = version;
        this.feature = feature;
        this.issueDate = issueDate;
        this.issueDateInMillis = -1;
        this.expiryDate = expiryDate;
        this.expiryDateInMillis = -1;
        this.uid = uid;
        this.type = type;
        this.subscriptionType = subscriptionType;
        this.issuedTo = issuedTo;
        this.issuer = issuer;
        this.maxNodes = maxNodes;
    }
}

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor.tools;

import org.elasticsearch.cli.Command;
import org.elasticsearch.cli.CommandTestCase;
import org.elasticsearch.cli.ExitCodes;
import org.elasticsearch.cli.UserException;

import java.nio.file.Files;
import java.nio.file.Path;

import static org.hamcrest.CoreMatchers.containsString;
```

```java
public class KeyPairGenerationToolTests extends CommandTestCase {

    @Override
    protected Command newCommand() {
        return new KeyPairGeneratorTool();
    }

    public void testMissingKeyPaths() throws Exception {
        Path exists = createTempFile("", "existing");
        Path dne = createTempDir().resolve("dne");
        UserException e = expectThrows(
            UserException.class,
            () -> execute(
                "--publicKeyPath",
                exists.toString(),
                "--privateKeyPath",
                dne.toString()));
        assertThat(e.getMessage(), containsString("existing"));
        assertEquals(ExitCodes.USAGE, e.exitCode);
        e = expectThrows(
            UserException.class,
            () -> execute(
                "--publicKeyPath",
                dne.toString(),
                "--privateKeyPath",
                exists.toString()));
        assertThat(e.getMessage(), containsString("existing"));
        assertEquals(ExitCodes.USAGE, e.exitCode);
    }

    public void testTool() throws Exception {
        Path keysDir = createTempDir();
        Path publicKeyFilePath = keysDir.resolve("public");
        Path privateKeyFilePath = keysDir.resolve("private");

        execute(
            "--publicKeyPath",
            publicKeyFilePath.toString(),
            "--privateKeyPath",
            privateKeyFilePath.toString());
        assertTrue(publicKeyFilePath.toString(), Files.exists(publicKeyFilePath));
        assertTrue(privateKeyFilePath.toString(), Files.exists(privateKeyFilePath));
    }

}
/*
* Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
* or more contributor license agreements. Licensed under the Elastic License;
```

```java
* you may not use this file except in compliance with the Elastic License.
*/
package org.elasticsearch.license.licensor.tools;

import org.elasticsearch.cli.Command;
import org.elasticsearch.cli.CommandTestCase;
import org.elasticsearch.cli.ExitCodes;
import org.elasticsearch.cli.UserException;
import org.elasticsearch.common.unit.TimeValue;
import org.elasticsearch.license.License;
import org.elasticsearch.license.licensor.TestUtils;
import org.junit.Before;

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;

public class LicenseVerificationToolTests extends CommandTestCase {
    protected Path pubKeyPath = null;
    protected Path priKeyPath = null;

    @Before
    public void setup() throws Exception {
        logger.error("project.basedir [{}]", System.getProperty("project.basedir"));
        pubKeyPath = getDataPath(TestUtils.PUBLIC_KEY_RESOURCE);
        priKeyPath = getDataPath(TestUtils.PRIVATE_KEY_RESOURCE);
    }

    @Override
    protected Command newCommand() {
        return new LicenseVerificationTool();
    }

    public void testMissingKeyPath() throws Exception {
        Path pub = createTempDir().resolve("pub");
        UserException e = expectThrows(
            UserException.class,
            () -> execute("--publicKeyPath", pub.toString()));
        assertTrue(e.getMessage(), e.getMessage().contains("pub does not exist"));
        assertEquals(ExitCodes.USAGE, e.exitCode);
    }

    public void testMissingLicenseSpec() throws Exception {
        UserException e = expectThrows(UserException.class, () -> {
            execute("--publicKeyPath", pubKeyPath.toString());
        });
        assertTrue(
            e.getMessage(),
```

```java
                    e.getMessage().contains("Must specify either --license or --licenseFile"));
            assertEquals(ExitCodes.USAGE, e.exitCode);
        }


    public void testBrokenLicense() throws Exception {
        final TimeValue oneHour = TimeValue.timeValueHours(1);
        License signedLicense = TestUtils.generateSignedLicense(oneHour, pubKeyPath, priKeyPath);
        License tamperedLicense = License.builder()
            .fromLicenseSpec(signedLicense, signedLicense.signature())
            .expiryDate(signedLicense.expiryDate() + randomIntBetween(1, 1000)).build();
        UserException e = expectThrows(
                UserException.class,
                () -> execute(
                    "--publicKeyPath",
                    pubKeyPath.toString(),
                    "--license",
                    TestUtils.dumpLicense(tamperedLicense)));
        assertEquals("Invalid License!", e.getMessage());
        assertEquals(ExitCodes.DATA_ERROR, e.exitCode);
    }

    public void testLicenseSpecString() throws Exception {
        final TimeValue oneHour = TimeValue.timeValueHours(1);
        License signedLicense = TestUtils.generateSignedLicense(oneHour, pubKeyPath, priKeyPath);
        String output = execute(
                "--publicKeyPath",
                pubKeyPath.toString(),
                "--license",
                TestUtils.dumpLicense(signedLicense));
        assertFalse(output, output.isEmpty());
    }

    public void testLicenseSpecFile() throws Exception {
        final TimeValue oneHour = TimeValue.timeValueHours(1);
        License signedLicense = TestUtils.generateSignedLicense(oneHour, pubKeyPath, priKeyPath);
        Path licenseSpecFile = createTempFile();
        Files.write(
                licenseSpecFile,
                TestUtils.dumpLicense(signedLicense).getBytes(StandardCharsets.UTF_8));
        String output = execute(
                "--publicKeyPath",
                pubKeyPath.toString(),
                "--licenseFile",
                licenseSpecFile.toString());
        assertFalse(output, output.isEmpty());
    }

}
```

```java
/*
 * Copyright Elasticsearch B.V. and/or licensed to Elasticsearch B.V. under one
 * or more contributor license agreements. Licensed under the Elastic License;
 * you may not use this file except in compliance with the Elastic License.
 */
package org.elasticsearch.license.licensor.tools;

import org.elasticsearch.cli.Command;
import org.elasticsearch.cli.CommandTestCase;
import org.elasticsearch.cli.ExitCodes;
import org.elasticsearch.cli.UserException;
import org.elasticsearch.common.bytes.BytesArray;
import org.elasticsearch.common.xcontent.XContentType;
import org.elasticsearch.license.License;
import org.elasticsearch.license.licensor.TestUtils;
import org.junit.Before;

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;

public class LicenseGenerationToolTests extends CommandTestCase {

    protected Path pubKeyPath = null;
    protected Path priKeyPath = null;

    @Before
    public void setup() throws Exception {
        pubKeyPath = getDataPath(TestUtils.PUBLIC_KEY_RESOURCE);
        priKeyPath = getDataPath(TestUtils.PRIVATE_KEY_RESOURCE);
    }

    @Override
    protected Command newCommand() {
        return new LicenseGeneratorTool();
    }

    public void testMissingKeyPaths() throws Exception {
        Path pub = createTempDir().resolve("pub");
        Path pri = createTempDir().resolve("pri");
        UserException e = expectThrows(
            UserException.class,
            () -> execute(
                "--publicKeyPath",
                pub.toString(),
                "--privateKeyPath",
                pri.toString()));
        assertTrue(e.getMessage(), e.getMessage().contains("pri does not exist"));
```

```java
        assertEquals(ExitCodes.USAGE, e.exitCode);

        Files.createFile(pri);
        e = expectThrows(
            UserException.class,
            () -> execute(
                "--publicKeyPath",
                pub.toString(),
                "--privateKeyPath",
                pri.toString()));
        assertTrue(e.getMessage(), e.getMessage().contains("pub does not exist"));
        assertEquals(ExitCodes.USAGE, e.exitCode);
    }

    public void testMissingLicenseSpec() throws Exception {
        UserException e = expectThrows(
            UserException.class,
            () -> execute(
                "--publicKeyPath",
                pubKeyPath.toString(),
                "--privateKeyPath",
                priKeyPath.toString()));
        assertTrue(
            e.getMessage(),
            e.getMessage().contains("Must specify either --license or --licenseFile"));
        assertEquals(ExitCodes.USAGE, e.exitCode);
    }

    public void testLicenseSpecString() throws Exception {
        TestUtils.LicenseSpec inputLicenseSpec =
            TestUtils.generateRandomLicenseSpec(License.VERSION_CURRENT);
        String licenseSpecString = TestUtils.generateLicenseSpecString(inputLicenseSpec);
        String output = execute(
            "--publicKeyPath",
            pubKeyPath.toString(),
            "--privateKeyPath",
            priKeyPath.toString(),
            "--license",
            licenseSpecString);
        final BytesArray bytes = new BytesArray(output.getBytes(StandardCharsets.UTF_8));
        License outputLicense = License.fromSource(bytes, XContentType.JSON);
        TestUtils.assertLicenseSpec(inputLicenseSpec, outputLicense);
    }

    public void testLicenseSpecFile() throws Exception {
        TestUtils.LicenseSpec inputLicenseSpec =
            TestUtils.generateRandomLicenseSpec(License.VERSION_CURRENT);
        String licenseSpecString = TestUtils.generateLicenseSpecString(inputLicenseSpec);
```

```
            Path licenseSpecFile = createTempFile();
            Files.write(licenseSpecFile, licenseSpecString.getBytes(StandardCharsets.UTF_8));
            String output = execute(
                  "--publicKeyPath",
                  pubKeyPath.toString(),
                  "--privateKeyPath",
                  priKeyPath.toString(),
                  "--licenseFile",
                  licenseSpecFile.toString());
            final BytesArray bytes = new BytesArray(output.getBytes(StandardCharsets.UTF_8));
            License outputLicense = License.fromSource(bytes, XContentType.JSON);
            TestUtils.assertLicenseSpec(inputLicenseSpec, outputLicense);
      }

}
apply plugin: 'elasticsearch.build'

dependencies {
 compile project(xpackModule('core'))
 compile "org.elasticsearch:elasticsearch:${version}"
 testCompile "org.elasticsearch.test:framework:${version}"
}

project.forbiddenPatterns {
 exclude '**/*.key'
}

dependencyLicenses.enabled = false

task buildZip(type: Zip, dependsOn: jar) {
 String parentDir = "license-tools-${version}"
 into(parentDir + '/lib') {
   from jar
   from configurations.runtime
 }
 into(parentDir + '/bin') {
   from 'bin'
 }
}

assemble.dependsOn buildZip
Elasticsearch X-Pack
Copyright 2009-2017 Elasticsearch
Apache Lucene
Copyright 2001-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
```

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
 http://snowball.tartarus.org/

The KStem stemmer in
 analysis/common/src/org/apache/lucene/analysis/en

was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, Hindi and Bengali analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bn/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary

(http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

 mecab-ipadic-2.7.0-20070801

which can be obtained from

 http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

 http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801 Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise

contravene any of the laws and regulations of the countries having jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the research and development conducted during the project and is provided to users as so produced on an experimental basis. Accordingly, the program is provided without any warranty whatsoever, whether express, implied, statutory or otherwise. The term "warranty" used herein includes, but is not limited to, any warranty of the quality, performance, merchantability and fitness for a particular purpose of the program and the nonexistence of any infringement or violation of any right of any third party.

Each user of the program will agree and understand, and be deemed to have agreed and understood, that there is no warranty whatsoever for the program and, accordingly, the entire risk arising from or otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other organization that participated in or was otherwise related to the development of the program and their respective officials, directors, officers and other employees shall be held liable for any and all damages, including, without limitation, general, special, incidental and consequential damages, arising out of or otherwise in connection with the use or inability to use the program or any product, material or result produced or otherwise obtained by using the program, regardless of whether they have been advised of, or otherwise had knowledge of, the possibility of such damages at any time during the project or thereafter. Each user will be deemed to have agreed to the foregoing by his or her commencement of use of the program. The term "use" as used herein includes, but is not limited to, the use, modification, copying and distribution of the program and the production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

===
Nori Korean Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

 mecab-ko-dic-2.0.3-20170922

which can be obtained from

 https://bitbucket.org/eunjeon/mecab-ko-dic/downloads/mecab-ko-dic-2.0.3-20170922.tar.gz
Apache Tika
Copyright 2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Copyright 1993-2010 University Corporation for Atmospheric Research/Unidata
This software contains code derived from UCAR/Unidata's NetCDF library.

Tika-server component uses CDDL-licensed dependencies: jersey (http://jersey.java.net/) and
Grizzly (http://grizzly.java.net/)

Tika-parsers component uses CDDL/LGPL dual-licensed dependency: jhighlight
(https://github.com/codelibs/jhighlight)

OpenCSV: Copyright 2005 Bytecode Pty Ltd. Licensed under the Apache License, Version 2.0

IPTC Photo Metadata descriptions Copyright 2010 International Press Telecommunications Council.
 ==
 == NOTICE file corresponding to section 4(d) of the Apache License,   ==
 == Version 2.0, in this case for the Apache XmlBeans distribution.   ==
 ==

 This product includes software developed by
 The Apache Software Foundation (http://www.apache.org/).

 Portions of this software were originally based on the following:
  - software copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

 Aside from contributions to the Apache XMLBeans project, this
 software also includes:

  - one or more source files from the Apache Xerces-J and Apache Axis
    products, Copyright (c) 1999-2003 Apache Software Foundation

  - W3C XML Schema documents Copyright 2001-2003 (c) World Wide Web
    Consortium (Massachusetts Institute of Technology, European Research
    Consortium for Informatics and Mathematics, Keio University)

  - resolver.jar from Apache Xml Commons project,
    Copyright (c) 2001-2003 Apache Software Foundation

- Piccolo XML Parser for Java from http://piccolo.sourceforge.net/,
  Copyright 2002 Yuval Oren under the terms of the Apache Software License 2.0

- JSR-173 Streaming API for XML from http://sourceforge.net/projects/xmlpullparser/,
  Copyright 2005 BEA under the terms of the Apache Software License 2.0
                        Apache License
                    Version 2.0, January 2004
                  http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications

represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without

modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
    distribution, then any Derivative Works that You distribute must
    include a readable copy of the attribution notices contained
    within such NOTICE file, excluding those notices that do not
    pertain to any part of the Derivative Works, in at least one
    of the following places: within a NOTICE text file distributed
    as part of the Derivative Works; within the Source form or
    documentation, if provided along with the Derivative Works; or,
    within a display generated by the Derivative Works, if and
    wherever such third-party notices normally appear. The contents
    of the NOTICE file are for informational purposes only and
    do not modify the License. You may add Your own attribution
    notices within Derivative Works that You distribute, alongside
    or as an addendum to the NOTICE text from the Work, provided
    that such additional attribution notices cannot be construed
    as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade

names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
   Apache License
                  Version 2.0, January 2004
              http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You

institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Apache POI
Copyright 2003-2015 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).

This product contains parts that were originally based on software from BEA.
Copyright (c) 2000-2003, BEA Systems, <http://www.bea.com/>.

This product contains W3C XML Schema documents. Copyright 2001-2003 (c)
World Wide Web Consortium (Massachusetts Institute of Technology, European
Research Consortium for Informatics and Mathematics, Keio University)

This product contains the Piccolo XML Parser for Java
(http://piccolo.sourceforge.net/). Copyright 2002 Yuval Oren.

This product contains the chunks_parse_cmds.tbl file from the vsdump program.
Copyright (C) 2006-2007 Valek Filippov (frob@df.ru)

This product contains parts of the eID Applet project
(http://eid-applet.googlecode.com). Copyright (c) 2009-2014
FedICT (federal ICT department of Belgium), e-Contract.be BVBA (https://www.e-contract.be),
Bart Hanssens from FedICT

Apache Commons IO
Copyright 2002-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache PDFBox
Copyright 2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Based on source code originally developed in the PDFBox and
FontBox projects.

Copyright (c) 2002-2007, www.pdfbox.org

Based on source code originally developed in the PaDaF project.
Copyright (c) 2010 Atos Worldline SAS

Includes the Adobe Glyph List
Copyright 1997, 1998, 2002, 2007, 2010 Adobe Systems Incorporated.

Includes the Zapf Dingbats Glyph List

Copyright 2002, 2010 Adobe Systems Incorporated.

Includes OSXAdapter
Copyright (C) 2003-2007 Apple, Inc., All Rights Reserved
The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
                    (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

EXTERNAL COMPONENTS

Apache PDFBox includes a number of components with separate copyright notices
and license terms. Your use of these components is subject to the terms and
conditions of the following licenses.

Contributions made to the original PDFBox and FontBox projects:

Copyright (c) 2002-2007, www.pdfbox.org
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

3. Neither the name of pdfbox; nor the names of its contributors may be
   used to endorse or promote products derived from this software without
   specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Adobe Font Metrics (AFM) for PDF Core 14 Fonts

This file and the 14 PostScript(R) AFM files it accompanies may be used,
copied, and distributed for any purpose and without charge, with or without
modification, provided that all copyright notices are retained; that the
AFM files are not distributed without this file; that all modifications
to this file or any of the AFM files are prominently noted in the modified
file(s); and that this paragraph is not modified. Adobe Systems has no
responsibility or obligation to support the use of the AFM files.

CMaps for PDF Fonts (http://opensource.adobe.com/wiki/display/cmap/Downloads)

Copyright 1990-2009 Adobe Systems Incorporated.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

Neither the name of Adobe Systems Incorporated nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

PaDaF PDF/A preflight (http://sourceforge.net/projects/padaf)

Copyright 2010 Atos Worldline SAS

Licensed by Atos Worldline SAS under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
Atos Worldline SAS licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

OSXAdapter

Version: 2.0

Disclaimer: IMPORTANT: This Apple software is supplied to you by
Apple Inc. ("Apple") in consideration of your agreement to the
following terms, and your use, installation, modification or
redistribution of this Apple software constitutes acceptance of these
terms. If you do not agree with these terms, please do not use,

install, modify or redistribute this Apple software.

In consideration of your agreement to abide by the following terms, and
subject to these terms, Apple grants you a personal, non-exclusive
license, under Apple's copyrights in this original Apple software (the
"Apple Software"), to use, reproduce, modify and redistribute the Apple
Software, with or without modifications, in source and/or binary forms;
provided that if you redistribute the Apple Software in its entirety and
without modifications, you must retain this notice and the following
text and disclaimers in all such redistributions of the Apple Software.
Neither the name, trademarks, service marks or logos of Apple Inc.
may be used to endorse or promote products derived from the Apple
Software without specific prior written permission from Apple. Except
as expressly stated in this notice, no other rights or licenses, express
or implied, are granted by Apple herein, including but not limited to
any patent rights that may be infringed by your derivative works or by
other works in which the Apple Software may be incorporated.

The Apple Software is provided by Apple on an "AS IS" basis. APPLE
MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION
THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND
OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION,
MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED
AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE),
STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2003-2007 Apple, Inc., All Rights Reserved
Apache Commons Logging
Copyright 2003-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.

APACHE POI SUBCOMPONENTS:

Apache POI includes subcomponents with separate copyright notices and
license terms. Your use of these subcomponents is subject to the terms
and conditions of the following licenses:

Office Open XML schemas (ooxml-schemas-1.1.jar)

  The Office Open XML schema definitions used by Apache POI are
  a part of the Office Open XML ECMA Specification (ECMA-376, [1]).
  As defined in section 9.4 of the ECMA bylaws [2], this specification
  is available to all interested parties without restriction:

    9.4 All documents when approved shall be made available to
      all interested parties without restriction.

  Furthermore, both Microsoft and Adobe have granted patent licenses
  to this work [3,4,5].

  [1] http://www.ecma-international.org/publications/standards/Ecma-376.htm
  [2] http://www.ecma-international.org/memento/Ecmabylaws.htm
  [3] http://www.microsoft.com/openspecifications/en/us/programs/osp/default.aspx
  [4] http://www.ecma-international.org/publications/files/ECMA-
ST/Ecma%20PATENT/Patent%20statements%20ok/ECMA-
376%20Edition%202%20Microsoft%20Patent%20Declaration.pdf
  [5] http://www.ecma-international.org/publications/files/ECMA-
ST/Ecma%20PATENT/Patent%20statements%20ok/ECMA-376%20Adobe%20Patent%20Declaration.pdf

JUnit test library (junit-4.11.jar)

  Common Public License - v 1.0

  THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON
  PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION
  OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

  1. DEFINITIONS

  "Contribution" means:

  a) in the case of the initial Contributor, the initial code and
    documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from
and are distributed by that particular Contributor. A Contribution
'originates' from a Contributor if it was added to the Program by
such Contributor itself or anyone acting on such Contributor's behalf.
Contributions do not include additions to the Program which: (i) are
separate modules of software distributed in conjunction with the
Program under their own license agreement, and (ii) are not derivative
works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which
are necessarily infringed by the use or sale of its Contribution alone
or when combined with the Program.

"Program" means the Contributions distributed in accordance with this
Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants
   Recipient a non-exclusive, worldwide, royalty-free copyright license
   to reproduce, prepare derivative works of, publicly display, publicly
   perform, distribute and sublicense the Contribution of such
   Contributor, if any, and such derivative works, in source code and
   object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants
   Recipient a non-exclusive, worldwide, royalty-free patent license under
   Licensed Patents to make, use, sell, offer to sell, import and
   otherwise transfer the Contribution of such Contributor, if any, in
   source code and object code form. This patent license shall apply to
   the combination of the Contribution and the Program if, at the time
   the Contribution is added by the Contributor, such addition of the
   Contribution causes such combination to be covered by the Licensed
   Patents. The patent license shall not apply to any other combinations
   which include the Contribution. No hardware per se is licensed
   hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

  i)   effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

  ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

  iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

  iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of
   the Program.

Contributors may not remove or alter any copyright notices contained
within the Program.

Each Contributor must identify itself as the originator of its
Contribution, if any, in a manner that reasonably allows subsequent
Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities
with respect to end users, business partners and the like. While this
license is intended to facilitate the commercial use of the Program,
the Contributor who includes the Program in a commercial product offering
should do so in a manner which does not create potential liability for
other Contributors. Therefore, if a Contributor includes the Program
in a commercial product offering, such Contributor ("Commercial
Contributor") hereby agrees to defend and indemnify every other
Contributor ("Indemnified Contributor") against any losses, damages
and costs (collectively "Losses") arising from claims, lawsuits and
other legal actions brought by a third party against the Indemnified
Contributor to the extent caused by the acts or omissions of such
Commercial Contributor in connection with its distribution of the
Program in a commercial product offering. The obligations in this
section do not apply to any claims or Losses relating to any actual
or alleged intellectual property infringement. In order to qualify,
an Indemnified Contributor must: a) promptly notify the Commercial
Contributor in writing of such claim, and b) allow the Commercial
Contributor to control, and cooperate with the Commercial Contributor
in, the defense and any related settlement negotiations. The Indemnified
Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial
product offering, Product X. That Contributor is then a Commercial
Contributor. If that Commercial Contributor then makes performance
claims, or offers warranties related to Product X, those performance
claims and warranties are such Commercial Contributor's responsibility
alone. Under this section, the Commercial Contributor would have to
defend claims against the other Contributors related to those
performance claims and warranties, and if a court requires any other
Contributor to pay any damages as a result, the Commercial Contributor
must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient

relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement,
but in order to avoid inconsistency the Agreement is copyrighted and may
only be modified in the following manner. The Agreement Steward reserves
the right to publish new versions (including revisions) of this Agreement
from time to time. No one other than the Agreement Steward has the right
to modify this Agreement. IBM is the initial Agreement Steward. IBM may
assign the responsibility to serve as the Agreement Steward to a suitable
separate entity. Each new version of the Agreement will be given a
distinguishing version number. The Program (including Contributions) may
always be distributed subject to the version of the Agreement under which
it was received. In addition, after a new version of the Agreement is
published, Contributor may elect to distribute the Program (including
its Contributions) under the new version. Except as expressly stated in
Sections 2(a) and 2(b) above, Recipient receives no rights or licenses
to the intellectual property of any Contributor under this Agreement,
whether expressly, by implication, estoppel or otherwise. All rights in
the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to
this Agreement will bring a legal action under this Agreement more than
one year after the cause of action arose. Each party waives its rights
to a jury trial in any resulting litigation.
Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache Commons CLI
Copyright 2001-2009 The Apache Software Foundation

This product includes software developed by
The Apache Software Foundation (http://www.apache.org/).
                        Apache License
                   Version 2.0, January 2004
                  http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

THIS PRODUCT ALSO INCLUDES THIRD PARTY SOFTWARE REDISTRIBUTED UNDER THE
FOLLOWING LICENSES:

Apache Commons Logging,
  The Apache Software License, Version 1.1 (commons-logging-1.1.1.jar)

The Apache Software License, Version 1.1

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the
   distribution.

3. The end-user documentation included with the redistribution,
   if any, must include the following acknowledgment:
     "This product includes software developed by the
      Apache Software Foundation (http://www.apache.org/)."
   Alternately, this acknowledgment may appear in the software itself,

if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" must
   not be used to endorse or promote products derived from this
   software without prior written permission. For written
   permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache",
   nor may "Apache" appear in their name, without prior written
   permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Test messages from the Perl-MIME-Tools project,

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a
Package may be copied, such that the Copyright Holder maintains some
semblance of artistic control over the development of the package,
while giving the users of the package the right to use and distribute
the Package in a more-or-less customary fashion, plus the right to make
reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the
Copyright Holder, and derivatives of that collection of files
created through textual modification.

"Standard Version" refers to such a Package if it has not been
modified, or has been modified in accordance with the wishes
of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or
copyrights for the package.

"You" is you, if you're thinking about copying or distributing
this Package.

"Reasonable copying fee" is whatever you can justify on the
basis of media cost, duplication charges, time of people involved,
and so on. (You will not be required to justify it to the
Copyright Holder, but only to the computing community at large
as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item
itself, though there may be fees involved in handling the item.
It also means that recipients of the item may redistribute it
under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications
derived from the Public Domain or from the Copyright Holder. A Package
modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided
that you insert a prominent notice in each changed file stating how and
when you changed that file, and provided that you do at least ONE of the
following:

   a) place your modifications in the Public Domain or otherwise make them
   Freely Available, such as by posting said modifications to Usenet or
   an equivalent medium, or placing the modifications on a major archive
   site such as uunet.uu.net, or by allowing the Copyright Holder to include
   your modifications in the Standard Version of the Package.

   b) use the modified Package only within your corporation or organization.

   c) rename any non-standard executables so the names do not conflict
   with standard executables, which must also be provided, and provide
   a separate manual page for each non-standard executable that clearly
   documents how it differs from the Standard Version.

   d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or
executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be

construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

    The End

Licensing of XZ for Java
=======================

  All the files in this package have been written by Lasse Collin
  and/or Igor Pavlov. All these files have been put into the
  public domain. You can do whatever you want with these files.

  This software is provided "as is", without any warranty.
Copyright (c) 2004-2014 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,   FITNESS   FOR   A   PARTICULAR   PURPOSE   AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

                    Apache License
                Version 2.0, January 2004
              http://www.apache.org/licenses/

  TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.

APACHE TIKA SUBCOMPONENTS

Apache Tika includes a number of subcomponents with separate copyright notices
and license terms. Your use of these subcomponents is subject to the terms and
conditions of the following licenses.

MIME type information from file-4.26.tar.gz (http://www.darwinsys.com/file/)

Copyright (c) Ian F. Darwin 1986, 1987, 1989, 1990, 1991, 1992, 1994, 1995.
Software written by Ian F. Darwin and others;
maintained 1994- Christos Zoulas.

This software is not subject to any export provision of the United States
Department of Commerce, and may be exported to any country or planet.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice immediately at the beginning of the file, without modification,
   this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Charset detection code from ICU4J (http://site.icu-project.org/)

Copyright (c) 1995-2009 International Business Machines Corporation
and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, and/or sell copies of the Software, and to permit persons
to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies
of the Software and that both the above copyright notice(s) and this
permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE
BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES,
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall
not be used in advertising or otherwise to promote the sale, use or other
dealings in this Software without prior written authorization of the
copyright holder.

Parsing functionality provided by the NetCDF Java Library (http://www.unidata.ucar.edu/software/netcdf-java/)

Copyright 1993-2010 University Corporation for Atmospheric Research/Unidata

Portions of this software were developed by the Unidata Program at the University
Corporation for Atmospheric Research.

Access and use of this software shall impose the following obligations and understandings
on the user. The user is granted the right, without any fee or cost, to use, copy, modify,
alter, enhance and distribute this software, and any derivative works thereof, and its
supporting documentation for any purpose whatsoever, provided that this entire notice
appears in all copies of the software, derivative works and supporting documentation. Further,
UCAR requests that the user credit UCAR/Unidata in any publications that result from the use
of this software or in any product that includes this software, although this is not an obligation.
The names UCAR and/or Unidata, however, may not be used in any advertising or publicity to endorse
or promote any products or commercial entity unless specific written permission is obtained from
UCAR/Unidata. The user also understands that UCAR/Unidata is not obligated to provide the user with
any support, consulting, training or assistance of any kind with regard to the use, operation and
performance of this software nor to provide the user with any updates, revisions, new versions or
"bug fixes."

IPTC Photo Metadata descriptions are taken from the IPTC Photo Metadata Standard, July 2010, Copyright 2010 International Press Telecommunications Council.

PRODUCT LIABILITY OR OTHERWISE.

6. The IPTC takes no position regarding the validity or scope of any Intellectual Property or other rights that might be claimed to pertain to the implementation or use of the technology described in the Document or the extent to which any license under such rights might or might not be available. The IPTC does not represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication, assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of the Specifications and Materials, can be obtained from the Managing Director of the IPTC.

7. By using the Specifications and Materials including the Document in any manner or for any purpose, you release the IPTC from all liabilities, claims, causes of action, allegations, losses, injuries, damages, or detriments of any nature arising from or relating to the use of the Specifications, Materials or any portion thereof. You further agree not to file a lawsuit, make a claim, or take any other formal or informal legal action against the IPTC, resulting from your acquisition, use, duplication, distribution, or exploitation of the Specifications, Materials or any portion thereof. Finally, you hereby agree that the IPTC is not liable for any direct, indirect, special or consequential damages arising from or relating to your acquisition, use, duplication, distribution, or exploitation of the Specifications, Materials or any portion thereof.

8. Specifications and Materials may be downloaded or copied provided that ALL copies retain the ownership, copyright and license notices.

9. Materials may not be edited, modified, or presented in a context that creates a misleading or false impression or statement as to the positions, actions, or statements of the IPTC.

10. The name and trademarks of the IPTC may not be used in advertising, publicity, or in relation to products or services and their names without the specific, written prior permission of the IPTC. Any permitted use of the trademarks of the IPTC, whether registered or not, shall be accompanied by an appropriate mark and attribution, as agreed with the IPTC.

11. Specifications may be extended by both members and non-members to provide additional functionality (Extension Specifications) provided that there is a clear recognition of the IPTC IP and its ownership in the Extension Specifications and the related documentation and provided that the extensions are clearly identified and provided that a perpetual license is granted by the creator of the Extension Specifications for other members and non-members to use the Extension Specifications and to continue extensions of the Extension Specifications. The IPTC does not waive any of its rights in the Specifications and Materials in this context. The Extension Specifications may be considered the intellectual property of their creator. The IPTC expressly disclaims any responsibility for damage caused by an extension to the Specifications.

12. Specifications and Materials may be included in derivative work of both members and non-members provided that there is a clear recognition of the IPTC IP and its ownership in the derivative work and its related documentation. The IPTC does not waive any of its rights in the Specifications and Materials in this context. Derivative work in its entirety may be considered the intellectual property of the creator of the work .The IPTC expressly disclaims any responsibility for damage caused when its IP is used in a derivative context.

13. This Specifications License Agreement is perpetual subject to your conformance to the terms of this Agreement. The IPTC may terminate this Specifications License Agreement immediately upon your breach of this Agreement and, upon such termination you will cease all use, duplication, distribution, and/or exploitation in any manner of the Specifications and Materials.

14. This Specifications License Agreement reflects the entire agreement of the parties regarding the subject matter hereof and supersedes all prior agreements or representations regarding such matters, whether written or oral. To the extent any portion or provision of this Specifications License Agreement is found to be illegal or unenforceable, then the remaining provisions of this Specifications License Agreement will remain in full force and effect and the illegal or unenforceable provision will be construed to give it such effect as it may properly have that is consistent with the intentions of the parties.

15. This Specifications License Agreement may only be modified in writing signed by an authorized representative

of the IPTC.

16. This Specifications License Agreement is governed by the law of United Kingdom, as such law is applied to contracts made and fully performed in the United Kingdom. Any disputes arising from or relating to this Specifications License Agreement will be resolved in the courts of the United Kingdom. You consent to the jurisdiction of such courts over you and covenant not to assert before such courts any objection to proceeding in such forums.

JUnRAR (https://github.com/edmund-wagner/junrar/)

JUnRAR is based on the UnRAR tool, and covered by the same license
It was formerly available from http://java-unrar.svn.sourceforge.net/

```
******  *****  ******  UnRAR - free utility for RAR archives
** ** ** ** ** **  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
****** ******* ******   License for use and distribution of
** ** ** ** ** **  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
** ** ** ** ** **      FREE portable version
                  ~~~~~~~~~~~~~~~~~~~~
```

The source code of UnRAR utility is freeware. This means:

1. All copyrights to RAR and the utility UnRAR are exclusively
   owned by the author - Alexander Roshal.

2. The UnRAR sources may be used in any software to handle RAR
   archives without limitations free of charge, but cannot be used
   to re-create the RAR compression algorithm, which is proprietary.
   Distribution of modified UnRAR sources in separate form or as a
   part of other software is permitted, provided that it is clearly
   stated in the documentation and source comments that the code may
   not be used to develop a RAR (WinRAR) compatible archiver.

3. The UnRAR utility may be freely distributed. It is allowed
   to distribute UnRAR inside of other software packages.

4. THE RAR ARCHIVER AND THE UnRAR UTILITY ARE DISTRIBUTED "AS IS".
   NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT
   YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS,
   DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING
   OR MISUSING THIS SOFTWARE.

5. Installing and using the UnRAR utility signifies acceptance of
   these terms and conditions of the license.

6. If you don't agree with terms of the license you must remove
   UnRAR files from your storage devices and cease to use the
   utility.

Thank you for your interest in RAR and UnRAR. Alexander L. Roshal

Sqlite (bundled in org.xerial's sqlite-jdbc)
   This product bundles Sqlite, which is in the Public Domain. For details
   see: https://www.sqlite.org/copyright.html
Apache Commons Compress
Copyright 2002-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

The files in the package org.apache.commons.compress.archivers.sevenz
were derived from the LZMA SDK, version 9.20 (C/ and CPP/7zip/),
which has been placed in the public domain:

"LZMA SDK is placed in the public domain." (http://www.7-zip.org/sdk.html)
The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
                  (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
                  MOZILLA PUBLIC LICENSE
                     Version 1.1

                     ----

1. Definitions.

   1.0.1. "Commercial Use" means distribution or otherwise making the
   Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:
    A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

    B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or

hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:
   (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

   (b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

   (c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

   (d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused

by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d)   Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this

License or the recipients' rights hereunder. However, You may include
an additional document offering the additional rights described in
Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be
made available in Source Code form under the terms of this License
either on the same media as an Executable version or via an accepted
Electronic Distribution Mechanism to anyone to whom you made an
Executable version available; and if made available via Electronic
Distribution Mechanism, must remain available for at least twelve (12)
months after the date it initially became available, or at least six
(6) months after a subsequent version of that particular Modification
has been made available to such recipients. You are responsible for
ensuring that the Source Code version remains available even if the
Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a
file documenting the changes You made to create that Covered Code and
the date of any change. You must include a prominent statement that
the Modification is derived, directly or indirectly, from Original
Code provided by the Initial Developer and including the name of the
Initial Developer in (a) the Source Code, and (b) in any notice in an
Executable version or related documentation in which You describe the
origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
    (a) Third Party Claims.
    If Contributor has knowledge that a license under a third party's
    intellectual property rights is required to exercise the rights
    granted by such Contributor under Sections 2.1 or 2.2,
    Contributor must include a text file with the Source Code
    distribution titled "LEGAL" which describes the claim and the
    party making the claim in sufficient detail that a recipient will
    know whom to contact. If Contributor obtains such knowledge after
    the Modification is made available as described in Section 3.2,
    Contributor shall promptly modify the LEGAL file in all copies
    Contributor makes available thereafter and shall take other steps
    (such as notifying appropriate mailing lists or newsgroups)
    reasonably calculated to inform those who received the Covered
    Code that new knowledge has been obtained.

    (b) Contributor APIs.
    If Contributor's Modifications include an application programming
    interface and Contributor has knowledge of patent licenses which
    are reasonably necessary to implement that API, Contributor must
    also include this information in the LEGAL file.

(c)   Representations.

Contributor represents that, except as disclosed pursuant to
Section 3.4(a) above, Contributor believes that Contributor's
Modifications are Contributor's original creation(s) and/or
Contributor has sufficient rights to grant the rights conveyed by
this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source
Code. If it is not possible to put such notice in a particular Source
Code file due to its structure, then You must include such notice in a
location (such as a relevant directory) where a user would be likely
to look for such a notice. If You created one or more Modification(s)
You may add your name as a Contributor to the notice described in
Exhibit A. You must also duplicate this License in any documentation
for the Source Code where You describe recipients' rights or ownership
rights relating to Covered Code. You may choose to offer, and to
charge a fee for, warranty, support, indemnity or liability
obligations to one or more recipients of Covered Code. However, You
may do so only on Your own behalf, and not on behalf of the Initial
Developer or any Contributor. You must make it absolutely clear than
any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial
Developer and every Contributor for any liability incurred by the
Initial Developer or such Contributor as a result of warranty,
support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the
requirements of Section 3.1-3.5 have been met for that Covered Code,
and if You include a notice stating that the Source Code version of
the Covered Code is available under the terms of this License,
including a description of how and where You have fulfilled the
obligations of Section 3.2. The notice must be conspicuously included
in any notice in an Executable version, related documentation or
collateral in which You describe recipients' rights relating to the
Covered Code. You may distribute the Executable version of Covered
Code or ownership rights under a license of Your choice, which may
contain terms different from this License, provided that You are in
compliance with the terms of this License and that the license for the
Executable version does not attempt to limit or alter the recipient's
rights in the Source Code version from the rights set forth in this
License. If You distribute the Executable version under a different
license You must make it absolutely clear that any terms which differ
from this License are offered by You alone, not by the Initial
Developer or any Contributor. You hereby agree to indemnify the
Initial Developer and every Contributor for any liability incurred by

the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your

license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under

Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995),

all U.S. Government End Users acquire Covered Code with only those
rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. This License shall be governed by
California law provisions (except to the extent applicable law, if
any, provides otherwise), excluding its conflict-of-law provisions.
With respect to disputes in which at least one party is a citizen of,
or an entity chartered or registered to do business in the United
States of America, any litigation relating to this License shall be
subject to the jurisdiction of the Federal Courts of the Northern
District of California, with venue lying in Santa Clara County,
California, with the losing party responsible for costs, including
without limitation, court costs and reasonable attorneys' fees and
expenses. The application of the United Nations Convention on
Contracts for the International Sale of Goods is expressly excluded.
Any law or regulation which provides that the language of a contract
shall be construed against the drafter shall not apply to this
License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is
responsible for claims and damages arising, directly or indirectly,
out of its utilization of rights under this License and You agree to
work with Initial Developer and Contributors to distribute such
responsibility on an equitable basis. Nothing herein is intended or
shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as
"Multiple-Licensed". "Multiple-Licensed" means that the Initial
Developer permits you to utilize portions of the Covered Code under
Your choice of the NPL or the alternative licenses, if any, specified
by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License
Version 1.1 (the "License"); you may not use this file except in
compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS"
basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
License for the specific language governing rights and limitations
under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____.
Portions created by _____ are Copyright (C) _____
_____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms
of the _____ license (the "[___] License"), in which case the
provisions of [_____] License are applicable instead of those
above. If you wish to allow use of your version of this file only
under the terms of the [____] License and not to allow others to use
your version of this file under the MPL, indicate your decision by
deleting the provisions above and replace them with the notice and
other provisions required by the [___] License. If you do not delete
the provisions above, a recipient may use your version of this file
under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of
the notices in the Source Code files of the Original Code. You should
use the text of this Exhibit A rather than the text found in the
Original Code Source Code for Your Modifications.]

==
==   NOTICE file for use with the Apache License, Version 2.0,   ==
==

Apache JAMES Mime4j
Copyright 2004-2010 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product test suite includes data (mimetools-testmsgs folder) developed
by Eryq and ZeeGee Software Inc as part of the "MIME-tools" Perl5 toolkit
and licensed under the Artistic License
Apache HttpComponents Core
Copyright 2005-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This project contains annotations derived from JCIP-ANNOTATIONS

Copyright (c) 2005 Brian Goetz and Tim Peierls. See http://www.jcip.net

AWS SDK for Java

Copyright 2010-2014 Amazon.com, Inc. or its affiliates. All Rights Reserved.

This product includes software developed by

Amazon Technologies, Inc (http://www.amazon.com/).

****

THIRD PARTY COMPONENTS

****

This software includes third party software subject to the following copyrights:

- XML parsing and utility functions from JetS3t - Copyright 2006-2009 James Murty.

- JSON parsing and utility functions from JSON.org - Copyright 2002 JSON.org.

- PKCS#1 PEM encoded private key parsing and utility functions from oauth.googlecode.com - Copyright 1998-2010 AOL Inc.

The licenses for these third party components are included in LICENSE.txt

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

  1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

  1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

  1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

  1.4. Executable. means the Covered Software in any form other than Source Code.

  1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

  1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

  1.7. License. means this document.

  1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent

Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.
Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.
The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.
You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.
You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.
You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided

that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY

INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

  The Covered Software is a .commercial item,. as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

  This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you

distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source

code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

  a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

  b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

  c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program

with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

 a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

 b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

 c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in

full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be

a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM

(INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

  One line to give the program's name and a brief idea of what it does.

  Copyright (C)

  This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

  This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

  You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

  Gnomovision version 69, Copyright (C) year name of author
  Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

  signature of Ty Coon, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do

not wish to do so, delete this exception statement from your version.

Apache Commons Codec

Copyright 2002-2015 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java

contains test data from http://aspell.net/test/orig/batch0.tab.

Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

===

The content of package org.apache.commons.codec.language.bm has been translated

from the original php source code available at http://stevemorse.org/phoneticinfo.htm

with permission from the original authors.

Original source copyright:

Copyright (c) 2008 Alexander Beider & Stephen P. Morse.

Apache HttpComponents Client

Copyright 1999-2015 The Apache Software Foundation

This product includes software developed at

The Apache Software Foundation (http://www.apache.org/).

Apache License

Version 2.0, January 2004

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding

those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Note: Other license terms may apply to certain, identified software files contained within or distributed with the accompanying software if such terms are included in the directory containing the accompanying software. Such

other license terms will then apply in lieu of the terms of the software license above.

JSON processing code subject to the JSON License from JSON.org:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

The Software shall be used for Good, not Evil.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

===

This project contains annotations in the package org.apache.http.annotation
which are derived from JCIP-ANNOTATIONS
Copyright (c) 2005 Brian Goetz and Tim Peierls.
See http://www.jcip.net and the Creative Commons Attribution License
(http://creativecommons.org/licenses/by/2.5)
Full text: http://creativecommons.org/licenses/by/2.5/legalcode

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS
PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR
OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS
LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE
BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED
HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

   "Collective Work" means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its
entirety in unmodified form, along with a number of other contributions, constituting separate and independent
works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be
considered a Derivative Work (as defined below) for the purposes of this License.
   "Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works, such as a
translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art
reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or
adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the
purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording,
the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative
Work for the purpose of this License.
   "Licensor" means the individual or entity that offers the Work under the terms of this License.
   "Original Author" means the individual or entity who created the Work.
   "Work" means the copyrightable work of authorship offered under the terms of this License.
   "You" means an individual or entity exercising rights under this License who has not previously violated the terms
of this License with respect to the Work, or who has received express permission from the Licensor to exercise
rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use,
first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable

laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

to create and reproduce Derivative Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

For the avoidance of doubt, where the work is a musical composition:

Performance Royalties Under Blanket Licenses. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

Mechanical Rights and Statutory Royalties. Licensor waives the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

Webcasting Rights and Statutory Royalties. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions.The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any credit as required by clause 4(b), as requested.

If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative

Works or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

ICU4J, (under lucene/analysis/icu) is licensed under an MIT style license (modules/analysis/icu/lib/icu4j-LICENSE-BSD_LIKE.txt) and Copyright (c) 1995-2012 International Business Machines Corporation and others

ICU License - ICU 1.8.1 and later

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2012 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, and/or sell copies of the
Software, and to permit persons to whom the Software is furnished to do so,
provided that the above copyright notice(s) and this permission notice appear
in all copies of the Software and that both the above copyright notice(s) and
this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE
LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other
dealings in this Software without prior written authorization of the
copyright holder.

All trademarks and registered trademarks mentioned herein are the property of
their respective owners.

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

Apache HTrace (incubating) is Apache 2.0 Licensed. See below for licensing of dependencies that are NOT Apache Licensed.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of,

the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

(a) You must give any other recipients of the Work or
   Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
   stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
   that You distribute, all copyright, patent, trademark, and
   attribution notices from the Source form of the Work,
   excluding those notices that do not pertain to any part of
   the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
   distribution, then any Derivative Works that You distribute must
   include a readable copy of the attribution notices contained
   within such NOTICE file, excluding those notices that do not
   pertain to any part of the Derivative Works, in at least one
   of the following places: within a NOTICE text file distributed
   as part of the Derivative Works; within the Source form or
   documentation, if provided along with the Derivative Works; or,
   within a display generated by the Derivative Works, if and
   wherever such third-party notices normally appear. The contents
   of the NOTICE file are for informational purposes only and
   do not modify the License. You may add Your own attribution
   notices within Derivative Works that You distribute, alongside
   or as an addendum to the NOTICE text from the Work, provided
   that such additional attribution notices cannot be construed
   as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Units, a string formatting go library, is Copyright (c) 2014 Alec Thomas
and MIT licensed: https://github.com/alecthomas/units/blob/master/COPYING

D3, a javascript library for manipulating data, used by htrace-hbase
is Copyright 2010-2014, Michael Bostock and BSD licensed:
https://github.com/mbostock/d3/blob/master/LICENSE

Bootstrap, an html, css, and javascript framework, is
Copyright (c) 2011-2015 Twitter, Inc and MIT licensed:
https://github.com/twbs/bootstrap/blob/master/LICENSE

underscore, a javascript library of functional programming helpers, is
(c) 2009-2014 Jeremy Ashkenas, DocumentCloud and Investigative Reporters
& Editors and an MIT license:
https://github.com/jashkenas/underscore/blob/master/LICENSE

jquery, a javascript library, is Copyright jQuery Foundation and other
contributors, https://jquery.org/. The software consists of
voluntary contributions made by many individuals. For exact
contribution history, see the revision history
available at https://github.com/jquery/jquery
It is MIT licensed:
https://github.com/jquery/jquery/blob/master/LICENSE.txt

backbone, is a javascript library, that is Copyright (c) 2010-2014
Jeremy Ashkenas, DocumentCloud. It is MIT licensed:
https://github.com/jashkenas/backbone/blob/master/LICENSE

moment.js is a front end time conversion project.
It is (c) 2011-2014 Tim Wood, Iskren Chernev, Moment.js contributors
and shared under the MIT license:
https://github.com/moment/moment/blob/develop/LICENSE

CMP is an implementation of the MessagePack serialization format in
C. It is licensed under the MIT license:

https://github.com/camgunz/cmp/blob/master/LICENSE

go-codec is an implementation of several serialization and deserialization
codecs in Go. It is licensed under the MIT license:
https://github.com/ugorji/go/blob/master/LICENSE
Copyright (c) <YEAR>, <OWNER>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following
disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Apache Commons Lang
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())
Apache HTrace
Copyright 2015 The Apache Software Foundation

This product includes software developed at The Apache Software
Foundation (http://www.apache.org/).

In addition, this product includes software dependencies. See
the accompanying LICENSE.txt for a listing of dependencies
that are NOT Apache licensed (with pointers to their licensing)

Apache HTrace includes an Apache Thrift connector to Zipkin. Zipkin
is a distributed tracing system that is Apache 2.0 Licensed.
Copyright 2012 Twitter, Inc.

Our Owl logo we took from http://www.clker.com/clipart-13653.html.
It is public domain/free.
Apache Commons Codec
Copyright 2002-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

src/test/org/apache/commons/codec/language/DoubleMetaphoneTest.java
contains test data from http://aspell.net/test/orig/batch0.tab.
Copyright (C) 2002 Kevin Atkinson (kevina@gnu.org)

==

The content of package org.apache.commons.codec.language.bm has been translated
from the original php source code available at http://stevemorse.org/phoneticinfo.htm
with permission from the original authors.
Original source copyright:
Copyright (c) 2008 Alexander Beider & Stephen P. Morse.
Copyright (c) 2004-2014 QOS.ch
All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,   FITNESS   FOR   A   PARTICULAR   PURPOSE   AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Apache Commons Collections
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
 COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0 1.

Definitions.

1.1. Contributor means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable means the Covered Software in any form other than Source Code.

1.5. Initial Developer means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License means this document.

1.8. Licensable means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications means the Source Code and Executable form of any of the following: A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications; B. Any new file that contains any part of the Original Software or previous Modification; or C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You (or Your) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, You includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, control means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof);

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License;

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant. Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code. Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications. The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have

sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices. You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms. You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions. You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipients rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works. You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions. Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions. You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions. When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY. COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN AS IS BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as Participant) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTYS NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS. The Covered Software is a commercial item, as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of commercial computer software (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and commercial computer software documentation as such terms are used in 48 C.F.R. 12.212

(Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdictions conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.
Apache Commons Configuration
Copyright 2001-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache Commons Lang
Copyright 2001-2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the Spring Framework,
under the Apache License 2.0 (see: StringUtils.containsWhitespace())
Apache License
                    Version 2.0, January 2004
                 http://www.apache.org/licenses/

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and
limitations under the License.
Copyright 2016, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2016, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

   * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
   * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
   * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

  "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

  "Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
/*
* Copyright (c) 2007-present, Stephen Colebourne & Michael Nascimento Santos
*

ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
This software is Copyright (c) 2013 by MaxMind, Inc.

This is free software, licensed under the Apache License, Version 2.0.
This software is Copyright (c) 2014 by MaxMind, Inc.

This is free software, licensed under the Apache License, Version 2.0.
COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)Version 1.1

1. Definitions.

   1.1. "Contributor" means each individual or entity that creates or contributes to the creation of Modifications.

   1.2. "Contributor Version" means the combination of the Original Software, prior Modifications used by a
Contributor (if any), and the Modifications made by that particular Contributor.

   1.3. "Covered Software" means (a) the Original Software, or (b) Modifications, or (c) the combination of files
containing Original Software with files containing Modifications, in each case including portions thereof.

   1.4. "Executable" means the Covered Software in any form other than Source Code.

   1.5. "Initial Developer" means the individual or entity that first makes Original Software available under this
License.

   1.6. "Larger Work" means a work which combines Covered Software or portions thereof with code not governed
by the terms of this License.

   1.7. "License" means this document.

   1.8. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the
initial grant or subsequently acquired, any and all of the rights conveyed herein.

   1.9. "Modifications" means the Source Code and Executable form of any of the following:

   A. Any file that results from an addition to, deletion from or modification of the contents of a file containing
Original Software or previous Modifications;

   B. Any new file that contains any part of the Original Software or previous Modification; or

   C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. "Original Software" means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. "Source Code" means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell,

offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients' rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient's rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Oracle is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE

PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as "Participant") alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. If You assert a patent infringement claim against Participant alleging that the Participant Software directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

6.4. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting

of "commercial computer software" (as that term is defined at 48 C.F.R. ? 252.227-7014(a)(1)) and "commercial computer software documentation" as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction's conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)
The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its

contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

  a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

  b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

  c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

  a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER

PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

  One line to give the program's name and a brief idea of what it does.

  Copyright (C)

  This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

  This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

  You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

  Gnomovision version 69, Copyright (C) year name of author
  Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

  signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Oracle are subject to the following clarification and special exception to the GPL Version 2, but only where Oracle has expressly included in the particular source file's header the words "Oracle designates this particular file as subject to the "Classpath" exception as provided by Oracle in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library. If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Copyright 2006 Envoi Solutions LLC

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Source code in this repository is variously licensed under the Apache License
Version 2.0, an Apache compatible license, or the Elastic License. Outside of
the "x-pack" folder, source code in a given file is licensed under the Apache
License Version 2.0, unless otherwise noted at the beginning of the file or a
LICENSE file present in the directory subtree declares a separate license.
Within the "x-pack" folder, source code in a given file is licensed under the
Elastic License, unless otherwise noted at the beginning of the file or a
LICENSE file present in the directory subtree declares a separate license.

The build produces two sets of binaries - one set that falls under the Elastic
License and another set that falls under Apache License Version 2.0. The
binaries that contain `-oss` in the artifact name are licensed under the Apache
License Version 2.0.
Source code in this repository is variously licensed under the Apache License
Version 2.0, an Apache compatible license, or the Elastic License. Outside of
the "x-pack" folder, source code in a given file is licensed under the Apache
License Version 2.0, unless otherwise noted at the beginning of the file or a
LICENSE file present in the directory subtree declares a separate license.
Within the "x-pack" folder, source code in a given file is licensed under the
Elastic License, unless otherwise noted at the beginning of the file or a
LICENSE file present in the directory subtree declares a separate license.

The build produces two sets of binaries - one set that falls under the Elastic
License and another set that falls under Apache License Version 2.0. The
binaries that contain `-oss` in the artifact name are licensed under the Apache

This product includes software developed by The Apache Software
Foundation (http://www.apache.org/).

1.43 elasticsearch-http-client 6.2.1

1.43.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"

replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.44 enzyme 3.2.0

1.44.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Airbnb, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.45 enzyme-adapter-react-16 1.1.0

1.46 enzyme-to-json 3.2.2
1.46.1 Available under license :

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.47 eslint 4.8.0

1.47.1 Available under license :

ESLint

Copyright JS Foundation and other contributors, https://js.foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright JS Foundation and other contributors, https://js.foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN

THE SOFTWARE.

1.48 eslint-config-airbnb 16.1.0
1.48.1 Available under license :

MIT License

Copyright (c) 2012 Airbnb

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.49 eslint-plugin-extra-rules 0.0.0-development

1.50 eslint-plugin-import 2.8.0
1.50.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Ben Mosher

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.51 eslint-plugin-jsx-a11y 6.0.3

1.51.1 Available under license :

The MIT License (MIT)
Copyright (c) 2016 Ethan Cohen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.52 eslint-plugin-react 7.4.0

1.52.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Yannick Croissant

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.53 express 4.15.4

1.53.1 Available under license :

(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>
Copyright (c) 2013-2014 Roman Shtylman <shtylman+expressjs@gmail.com>
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.54 fscreen 1.0.2

1.54.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Rafael Pedicini

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.55 gson 2.8.2

1.55.1 Available under license :

Google Gson

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2008-2011 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and

limitations under the License.

1.56 history 4.6.2

1.56.1 Available under license :

MIT License

Copyright (c) React Training 2016-2018

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.57 httpsnippet 1.16.5

1.57.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Mashape (https://www.mashape.com)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.58 identity-obj-proxy 3.0.0

1.58.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Keyan Zhang

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.59 immutability-helper 2.4.0

1.59.1 Available under license :

MIT License

Copyright (c) 2017 Moshe Kolodny

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.60 istanbul *

1.60.1 Available under license :

Copyright 2012 Yahoo! Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
    * Redistributions of source code must retain the above copyright
      notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.
    * Neither the name of the Yahoo! Inc. nor the
      names of its contributors may be used to endorse or promote products
      derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL YAHOO! INC. BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.61 jackson-annotations 2.7.5

1.61.1 Available under license :

This copy of Jackson JSON processor annotations is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.62 jackson-databind 2.9.9

1.62.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0
Jackson JSON processor

Jackson is a high-performance, Free/Open Source JSON processing library.
It was originally written by Tatu Saloranta (tatu.saloranta@iki.fi), and has
been in development since 2007.
It is currently developed by a community of developers, as well as supported
commercially by FasterXML.com.

Licensing

Jackson core and extension components may be licensed under different licenses.
To find the details that apply to this artifact see the accompanying LICENSE file.
For more information, including possible other licensing options, contact
FasterXML.com (http://fasterxml.com).

Credits

A list of contributors may be found from CREDITS file, which is included
in some artifacts (usually source distributions); but is always available
from the source code management (SCM) system project uses.

1.63 jacksonmodulejsonSchema 2.7.5

1.63.1 Available under license :

This copy of Jackson JSON processor databind module is licensed under the
Apache (Software) License, version 2.0 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

1.64 javax-mail 1.4.7

1.64.1 Available under license :

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0

1. Definitions.

1.1. Contributor. means each individual or entity that creates or contributes to the creation of Modifications.

1.2. Contributor Version. means the combination of the Original Software, prior Modifications used by a Contributor (if any), and the Modifications made by that particular Contributor.

1.3. Covered Software. means (a) the Original Software, or (b) Modifications, or (c) the combination of files containing Original Software with files containing Modifications, in each case including portions thereof.

1.4. Executable. means the Covered Software in any form other than Source Code.

1.5. Initial Developer. means the individual or entity that first makes Original Software available under this License.

1.6. Larger Work. means a work which combines Covered Software or portions thereof with code not governed by the terms of this License.

1.7. License. means this document.

1.8. Licensable. means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. Modifications. means the Source Code and Executable form of any of the following:

A. Any file that results from an addition to, deletion from or modification of the contents of a file containing Original Software or previous Modifications;

B. Any new file that contains any part of the Original Software or previous Modification; or

C. Any new file that is contributed or otherwise made available under the terms of this License.

1.10. Original Software. means the Source Code and Executable form of computer software code that is originally released under this License.

1.11. Patent Claims. means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.12. Source Code. means (a) the common form of computer software code in which modifications are made and (b) associated documentation included in or with such code.

1.13. You. (or .Your.) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, .You. includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, .control. means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. License Grants.

2.1. The Initial Developer Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, the Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer, to use, reproduce, modify, display, perform, sublicense and distribute the Original Software (or portions thereof), with or without Modifications, and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using or selling of Original Software, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Software (or portions thereof).

(c) The licenses granted in Sections 2.1(a) and (b) are effective on the date Initial Developer first distributes or otherwise makes the Original Software available to a third party under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: (1) for code that You delete from the Original Software, or (2) for infringements caused by: (i) the modification of the Original Software, or (ii) the combination of the Original Software with other software or devices.

2.2. Contributor Grant.

Conditioned upon Your compliance with Section 3.1 below and subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof), either on an unmodified basis, with other Modifications, as Covered Software and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: (1) Modifications made by that Contributor (or portions thereof); and (2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) The licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first distributes or otherwise makes the Modifications available to a third party.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: (1) for any code that Contributor has

deleted from the Contributor Version; (2) for infringements caused by: (i) third party modifications of Contributor Version, or (ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or (3) under Patent Claims infringed by Covered Software in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Availability of Source Code.

Any Covered Software that You distribute or otherwise make available in Executable form must also be made available in Source Code form and that Source Code form must be distributed only under the terms of this License. You must include a copy of this License with every copy of the Source Code form of the Covered Software You distribute or otherwise make available. You must inform recipients of any such Covered Software in Executable form as to how they can obtain such Covered Software in Source Code form in a reasonable manner on or through a medium customarily used for software exchange.

3.2. Modifications.

The Modifications that You create or to which You contribute are governed by the terms of this License. You represent that You believe Your Modifications are Your original creation(s) and/or You have sufficient rights to grant the rights conveyed by this License.

3.3. Required Notices.

You must include a notice in each of Your Modifications that identifies You as the Contributor of the Modification. You may not remove or alter any copyright, patent or trademark notices contained within the Covered Software, or any notices of licensing or any descriptive text giving attribution to any Contributor or the Initial Developer.

3.4. Application of Additional Terms.

You may not offer or impose any terms on any Covered Software in Source Code form that alters or restricts the applicable version of this License or the recipients. rights hereunder. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, you may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear that any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.5. Distribution of Executable Versions.

You may distribute the Executable form of the Covered Software under the terms of this License or under the terms of a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable form does not attempt to limit or alter the recipient.s rights in the Source Code form from the rights set forth in this License. If You distribute the Covered Software in Executable form under a different license, You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.6. Larger Works.

You may create a Larger Work by combining Covered Software with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the

requirements of this License are fulfilled for the Covered Software.

4. Versions of the License.

4.1. New Versions.

Sun Microsystems, Inc. is the initial license steward and may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Except as provided in Section 4.3, no one other than the license steward has the right to modify this License.

4.2. Effect of New Versions.

You may always continue to use, distribute or otherwise make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. If the Initial Developer includes a notice in the Original Software prohibiting it from being distributed or otherwise made available under any subsequent version of the License, You must distribute and make the Covered Software available under the terms of the version of the License under which You originally received the Covered Software. Otherwise, You may also choose to use, distribute or otherwise make the Covered Software available under the terms of any subsequent version of the License published by the license steward.

4.3. Modified Versions.

When You are an Initial Developer and You want to create a new license for Your Original Software, You may create and use a modified version of this License if You: (a) rename the license and remove any references to the name of the license steward (except to note that the license differs from this License); and (b) otherwise make it clear that the license contains terms which differ from this License.

5. DISCLAIMER OF WARRANTY.

  COVERED SOFTWARE IS PROVIDED UNDER THIS LICENSE ON AN .AS IS. BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED SOFTWARE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED SOFTWARE IS WITH YOU. SHOULD ANY COVERED SOFTWARE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

6. TERMINATION.

6.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

6.2. If You assert a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You assert such claim is referred to as .Participant.) alleging that the Participant Software (meaning the Contributor Version where the Participant is a Contributor or the Original Software where the Participant is the Initial Developer) directly or indirectly infringes any patent, then any and all rights granted directly or indirectly to You by such Participant, the Initial Developer (if the Initial Developer is not the Participant) and all Contributors under Sections 2.1 and/or 2.2 of this License shall,

upon 60 days notice from Participant terminate prospectively and automatically at the expiration of such 60 day notice period, unless if within such 60 day period You withdraw Your claim with respect to the Participant Software against such Participant either unilaterally or pursuant to a written agreement with Participant.

6.3. In the event of termination under Sections 6.1 or 6.2 above, all end user licenses that have been validly granted by You or any distributor hereunder prior to termination (excluding licenses granted to You by any distributor) shall survive termination.

7. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED SOFTWARE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOST PROFITS, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY.S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

8. U.S. GOVERNMENT END USERS.

The Covered Software is a .commercial item,. as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of .commercial computer software. (as that term is defined at 48 C.F.R. 252.227-7014(a)(1)) and .commercial computer software documentation. as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Software with only those rights set forth herein. This U.S. Government Rights clause is in lieu of, and supersedes, any other FAR, DFAR, or other clause or provision that addresses Government rights in computer software under this License.

9. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by the law of the jurisdiction specified in a notice contained within the Original Software (except to the extent applicable law, if any, provides otherwise), excluding such jurisdiction.s conflict-of-law provisions. Any litigation relating to this License shall be subject to the jurisdiction of the courts located in the jurisdiction and venue specified in a notice contained within the Original Software, with the losing party responsible for costs, including, without limitation, court costs and reasonable attorneys. fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License. You agree that You alone are responsible for compliance with the United States export administration regulations (and the export control laws and regulation of any other countries) when You use, distribute or otherwise make available any Covered Software.

10. RESPONSIBILITY FOR CLAIMS.

  As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

  NOTICE PURSUANT TO SECTION 9 OF THE COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL)

  The code released under the CDDL shall be governed by the laws of the State of California (excluding conflict-of-law provisions). Any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California and the state courts of the State of California, with venue lying in Santa Clara County, California.

The GNU General Public License (GPL) Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you

legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

  a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

  b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

  c) If the modified program normally reads commands interactively when run, you must cause it, when started

running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

  a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

  b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

  c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically

terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

  One line to give the program's name and a brief idea of what it does.

  Copyright (C)

  This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

  This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the

implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

  You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

  Gnomovision version 69, Copyright (C) year name of author
  Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

  signature of Ty Coon, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL VERSION 2

Certain source files distributed by Sun Microsystems, Inc. are subject to the following clarification and special exception to the GPL Version 2, but only where Sun has expressly included in the particular source file's header the words

"Sun designates this particular file as subject to the "Classpath" exception as provided by Sun in the License file that accompanied this code."

Linking this library statically or dynamically with other modules is making a combined work based on this library. Thus, the terms and conditions of the GNU General Public License Version 2 cover the whole combination.

As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module.? An independent module is a module

which is not derived from or based on this library.? If you modify this library, you may extend this exception to your version of the library, but you are not obligated to do so.? If you do not wish to do so, delete this exception statement from your version.

1.65 jest 21.2.1

1.65.1 Available under license :

MIT License

For Jest software

Copyright (c) 2014-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.66 jna 3.5.1

1.66.1 Available under license :

This copy of JNA is licensed under the
Lesser General Public License (LGPL), version 2.1 ("the License").
See the License for details about distribution rights, and the
specific rights regarding derivate works.

You may obtain a copy of the License at:

http://www.gnu.org/licenses/licenses.html

A copy is also included in the downloadable source code package
containing JNA, in file "LGPL2.1", under the same directory
as this file.

1~

V

W

XYZ[

V\

]

^_

`

a

bcd e f g h ij

V kl

V m nocpqrsnameLjava/lang/String;emailurlorganizationorganizationUrlrolesLjava/util/List;

Signature$Ljava/util/List<Ljava/lang/String;>;timezone

propertiesLjava/util/Properties;<init>()VCodeLineNumberTableLocalVariableTablethis$Lorg/apache/maven/model/

Contributor;addProperty'(Ljava/lang/String;Ljava/lang/String;)VkeyvalueaddRole(Ljava/lang/String;)VstringgetEm

ail()Ljava/lang/String;getNamegetOrganizationgetOrganizationUrl

getProperties()Ljava/util/Properties;getRoles()Ljava/util/List;&()Ljava/util/List<Ljava/lang/String;>;getTimezonege

tUrl

removeRolesetEmailsetNamesetOrganizationsetOrganizationUrl

setProperties(Ljava/util/Properties;)VsetRoles(Ljava/util/List;)VLocalVariableTypeTable'(Ljava/util/List<Ljava/lan

g/String;>;)VsetTimezonesetUrl

SourceFileContributor.java-

.@Atujava/lang/Stringjava/lang/ClassCastExceptionjava/lang/StringBuilder:Contributor.addRoles(string) parameter

must be instanceof vwx=<y<-9BCz{|"!

!$!%!+,java/util/Properties&'java/util/ArrayList*!#!=Contributor.removeRoles(string) parameter must be instanceof

}|"org/apache/maven/model/Contributorjava/lang/Objectjava/io/Serializableput8(Ljava/lang/Object;Ljava/lang/Obje

ct;)Ljava/lang/Object;append-

(Ljava/lang/String;)Ljava/lang/StringBuilder;java/lang/ClasstoStringjava/util/Listadd(Ljava/lang/Object;)Zremove!

!"!#!$!%!&'()*!+,-.//*123045/M*+,W1 236!7!0

P

Q89/r2+"YY

*

+W12232:!0Z\&^1_;<//*1230h=<//*1230r></*1230|?</*1230@A/I**Y*1230BC/I**Y*1230(DE</*1230F</*12

30G9/r2+"YY

*

+W12232:!0&1H9/>*+123"!0

I9/>*+123 !0

J9/>*+123$!0

K9/>*+123%!0

LM/>*+123+,0

NO/P*+123&'P&)0

 (QR9/>*+123*!0

S9/>*+123#!0

TU

1)

! " # $ %&'(nameLjava/lang/String;urldistributioncomments<init>()VCodeLineNumberTableLocalVariableTablethis

Lorg/apache/maven/model/License;getComments()Ljava/lang/String;getDistributiongetNamegetUrlsetComments(Lj

ava/lang/String;)VsetDistributionsetNamesetUrl

SourceFileLicense.java

org/apache/maven/model/Licensejava/lang/Objectjava/io/Serializable!

/*/*J/*]/*g/*q>*+

{|>*+

>*+

>*+

1
'U
VW &X &Y &Z
V[
&\ &]
'^
_`
Vab
Vc
d`
Ve
Vf
dghihjk
Ulm
n
o
pqhrs
Ut
uvw
xy
z|} classNameLjava/lang/String;nameactionStringactionsLjava/util/Set;
Signature#Ljava/util/Set<Ljava/lang/String;>;<init>()VCodeLineNumberTableLocalVariableTablethis
PermissionInnerClasses3Lorg/apache/tools/ant/types/Permissions$Permission;setClass(Ljava/lang/String;)VaClassg
etClassName()Ljava/lang/String;setNameaNamegetName
setActions
getActionsmatches(Ljava/security/Permission;)ZassizeIpermLjava/security/Permission;LocalVariableTypeTablepar
seActions#(Ljava/lang/String;)Ljava/util/Set;itemresulttkLjava/util/StringTokenizer;7(Ljava/lang/String;)Ljava/util/
Set<Ljava/lang/String;>;toString
SourceFilePermissions.java01~=()*)+)KL,-
@=*B=Fjava/util/HashSetjava/util/StringTokenizer,0=java/lang/StringBuilderPermission: ("",
"")R=1org/apache/tools/ant/types/Permissions$Permissionjava/lang/Objectjava/lang/Stringtrimlength()IgetClass()Lj

ava/lang/Class;java/lang/Classequals(Ljava/lang/Object;)ZendsWith(Ljava/lang/String;)Zjava/security/Permission
substring(II)Ljava/lang/String;
startsWith
java/util/Set removeAll(Ljava/util/Collection;)Z'(Ljava/lang/String;Ljava/lang/String;)V
hasMoreTokens()Z nextTokenaddappend-(Ljava/lang/String;)Ljava/lang/StringBuilder;-
(Ljava/lang/Object;)Ljava/lang/StringBuilder;&org/apache/tools/ant/types/Permissions!&'()*)+),-./
012/*34589:2A *+3

4 58 ;)<=2/*3458>:2A *+3

4 58 ?)@=2/*3458A:2V*++**+3 "458,)B=2/*3)458CD2*+

<

+**d*+***+M,>,*W,3>1245&6A7C:Q;S?Z@cAjBuCEH4*cE-jFG58HIJcE/KL2;YMY+N-"-:,W,3"PQRS#T-
U6W9X44#M);58;,)3N-(OPJ3N/.QR=2^4Y * ! * " *#$ %3a4458ST7

&{6

==
=========================

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from 'org.apache.velocity' to 'clover.org.apache.velocity'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.

==
=========================

                            Apache License
                       Version 2.0, January 2004
                      http://www.apache.org/licenses/

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

    1. Definitions.

       "License" shall mean the terms and conditions for use, reproduction,
       and distribution as defined by Sections 1 through 9 of this document.

       "Licensor" shall mean the copyright owner or entity authorized by
       the copyright owner that is granting the License.

       "Legal Entity" shall mean the union of the acting entity and all
       other entities that control, are controlled by, or are under common
       control with that entity. For the purposes of this definition,
       "control" means (i) the power, direct or indirect, to cause the
       direction or management of such entity, whether by contract or
       otherwise, or (ii) ownership of fifty percent (50%) or more of the
       outstanding shares, or (iii) beneficial ownership of such entity.

       "You" (or "Your") shall mean an individual or Legal Entity
       exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed

as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this

===
=========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from the 'antlr.*' to the 'clover.antlr.*'. This was necessary to avoid potential name
conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.
===
=========================

SOFTWARE RIGHTS

ANTLR 1989-2006 Developed by Terence Parr
Partially supported by University of San Francisco & jGuru.com

We reserve no legal rights to the ANTLR--it is fully in the
public domain. An individual or company may do whatever
they wish with source code distributed with ANTLR or the
code generated by ANTLR, including the incorporation of
ANTLR, or its output, into commerical software.

We encourage users to develop software with ANTLR. However,
we do ask that credit is given to us for developing
ANTLR. By "credit", we mean that if you use ANTLR or
incorporate any source code into one of your programs
(commercial product, research project, or otherwise) that
you acknowledge this fact somewhere in the documentation,
research report, etc... If you like ANTLR and have
developed a nice tool with the output, please mention that
you developed it using ANTLR. In addition, we ask that the
headers remain intact in our source code. As long as these
guidelines are kept, we expect to continue enhancing this
system and expect to make other tools available as they are
completed.

The primary ANTLR guy:

Terence Parr
parrt@cs.usfca.edu
parrt@antlr.org
==
==========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from 'it.unimi.dsi.fastutil' to 'clover.it.unimi.dsi.fastutil'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.
==
==========================

   GNU LESSER GENERAL PUBLIC LICENSE
        Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
   59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

Finally, software patents pose a constant threat to the existence of

any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the

Library.

 You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

 2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices
  stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no
  charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a
  table of data to be supplied by an application program that uses
  the facility, other than as an argument passed when the facility
  is invoked, then you must make a good faith effort to ensure that,
  in the event an application does not supply such function or
  table, the facility still operates, and performs whatever part of
  its purpose remains meaningful.

  (For example, a function in a library to compute square roots has
  a purpose that is entirely well-defined independent of the
  application. Therefore, Subsection 2d requires that any
  application-supplied function or table used by this function must
  be optional: if the application does not supply it, the square
  root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or

collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a

copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

 9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

 10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

 11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made

generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

    NO WARRANTY

  15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE

LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

 16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

 END OF TERMS AND CONDITIONS

 How to Apply These Terms to Your New Libraries

 If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

 To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

 <one line to give the library's name and a brief idea of what it does.>
 Copyright (C) <year> <name of author>

 This library is free software; you can redistribute it and/or
 modify it under the terms of the GNU Lesser General Public
 License as published by the Free Software Foundation; either
 version 2.1 of the License, or (at your option) any later version.

 This library is distributed in the hope that it will be useful,
 but WITHOUT ANY WARRANTY; without even the implied warranty of
 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 Lesser General Public License for more details.

 You should have received a copy of the GNU Lesser General Public
 License along with this library; if not, write to the Free Software
 Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

 Yoyodyne, Inc., hereby disclaims all copyright interest in the
 library `Frob' (a library for tweaking knobs) written by James Random Hacker.

 <signature of Ty Coon>, 1 April 1990
 Ty Coon, President of Vice

That's all there is to it!
/*--

$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp $

Copyright (C) 2000-2004 Jason Hunter & Brett McLaughlin.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions, and the disclaimer that follows
   these conditions in the documentation and/or other materials
   provided with the distribution.

3. The name "JDOM" must not be used to endorse or promote products
   derived from this software without prior written permission. For
   written permission, please contact <request_AT_jdom_DOT_org>.

4. Products derived from this software may not be called "JDOM", nor
   may "JDOM" appear in their name, without prior written permission
   from the JDOM Project Management <request_AT_jdom_DOT_org>.

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
     JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

                        Apache License
                   Version 2.0, January 2004
                  http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

     "License" shall mean the terms and conditions for use, reproduction,
     and distribution as defined by Sections 1 through 9 of this document.

     "Licensor" shall mean the copyright owner or entity authorized by
     the copyright owner that is granting the License.

     "Legal Entity" shall mean the union of the acting entity and all
     other entities that control, are controlled by, or are under common
     control with that entity. For the purposes of this definition,
     "control" means (i) the power, direct or indirect, to cause the
     direction or management of such entity, whether by contract or
     otherwise, or (ii) ownership of fifty percent (50%) or more of the
     outstanding shares, or (iii) beneficial ownership of such entity.

     "You" (or "Your") shall mean an individual or Legal Entity
     exercising permissions granted by this License.

     "Source" form shall mean the preferred form for making modifications,
     including but not limited to software source code, documentation
     source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==
=========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from 'com.keypoint/org.jfree' to 'clover.com.keypoint/clover.org.jfree'. This was
necessary to avoid potential name conflicts during instrumentation of a code using the original library when
using Clover. No source code of the original library was modified.
==
=========================

  GNU LESSER GENERAL PUBLIC LICENSE
       Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
   59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts

as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is

linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

 The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

   GNU LESSER GENERAL PUBLIC LICENSE
 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

 A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

 The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

 "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

 Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

 1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact

all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

 You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

 2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices
  stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no
  charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a
  table of data to be supplied by an application program that uses
  the facility, other than as an argument passed when the facility
  is invoked, then you must make a good faith effort to ensure that,
  in the event an application does not supply such function or
  table, the facility still operates, and performs whatever part of
  its purpose remains meaningful.

  (For example, a function in a library to compute square roots has
  a purpose that is entirely well-defined independent of the
  application. Therefore, Subsection 2d requires that any
  application-supplied function or table used by this function must
  be optional: if the application does not supply it, the square
  root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest

your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

 3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

 Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

 This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

 4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

 If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

 5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

 However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

 When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

 If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

 Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

 6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

 You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

  a) Accompany the work with the complete corresponding
  machine-readable source code for the Library including whatever
  changes were used in the work (which must be distributed under
  Sections 1 and 2 above); and, if the work is an executable linked
  with the Library, with the complete machine-readable "work that
  uses the Library", as object code and/or source code, so that the
  user can modify the Library and then relink to produce a modified
  executable containing the modified Library. (It is understood
  that the user who changes the contents of definitions files in the
  Library will not necessarily be able to recompile the application
  to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

 For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

 It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

 7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact

that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the

integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

 12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

 13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

 14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

  NO WARRANTY

 15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

 16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

 END OF TERMS AND CONDITIONS

 How to Apply These Terms to Your New Libraries

 If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

 To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

 <one line to give the library's name and a brief idea of what it does.>
 Copyright (C) <year> <name of author>

 This library is free software; you can redistribute it and/or
 modify it under the terms of the GNU Lesser General Public
 License as published by the Free Software Foundation; either
 version 2.1 of the License, or (at your option) any later version.

 This library is distributed in the hope that it will be useful,
 but WITHOUT ANY WARRANTY; without even the implied warranty of
 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 Lesser General Public License for more details.

 You should have received a copy of the GNU Lesser General Public
 License along with this library; if not, write to the Free Software
 Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

 Yoyodyne, Inc., hereby disclaims all copyright interest in the
 library `Frob' (a library for tweaking knobs) written by James Random Hacker.

 <signature of Ty Coon>, 1 April 1990
 Ty Coon, President of Vice

That's all there is to it!

<div align="center">

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

</div>

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

 1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/
Open Source License for the overlibmws Package

1. License coverage

Note that this license only covers the script library (javascript core and plugin modules) and not any supporting
material such as the overlibmws website or its online documentation and support files. You may not reproduce the
website or its online material without explicit written permission from the author, but can freely incorporate scripts
and procedures which are demonstrated in that material into your own HTML or XML documents.

2. License (Artistic)

  Preamble
  The intent of this document is to state the conditions under which a Package may be copied, such that the
Copyright Holder maintains some semblance of artistic control over the development of the package, while giving
the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the
right to make reasonable modifications.

  Definitions:
  "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of
files created through textual modification.

  "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with
the wishes of the Copyright Holder.

  "Copyright Holder" is whoever is named in the copyright or copyrights for the package.

  "You" is you, if you're thinking about copying or distributing this Package.

  "Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of
people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing
community at large as a market that must bear the fee.)

  "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling
the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

  You may make and give away verbatim copies of the source form of the Standard Version of this Package without
restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
  You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the
Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
  You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in
each changed file stating how and when you changed that file, and provided that you do at least ONE of the
following:

place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

use the modified Package only within your corporation or organization.

rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

make other distribution arrangements with the Copyright Holder.

You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

accompany the distribution with the machine-readable source of the Package with your modifications.

accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

make other distribution arrangements with the Copyright Holder.

You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.

The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

==

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'com.lowagie' to 'clover.com.lowagie'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

==

MOZILLA PUBLIC LICENSE
Version 1.1

----

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

   A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

   B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this

License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:
   (a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

   (b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

   (c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a

copy of this License with every copy of the Source Code You
distribute. You may not offer or impose any terms on any Source Code
version that alters or restricts the applicable version of this
License or the recipients' rights hereunder. However, You may include
an additional document offering the additional rights described in
Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be
made available in Source Code form under the terms of this License
either on the same media as an Executable version or via an accepted
Electronic Distribution Mechanism to anyone to whom you made an
Executable version available; and if made available via Electronic
Distribution Mechanism, must remain available for at least twelve (12)
months after the date it initially became available, or at least six
(6) months after a subsequent version of that particular Modification
has been made available to such recipients. You are responsible for
ensuring that the Source Code version remains available even if the
Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a
file documenting the changes You made to create that Covered Code and
the date of any change. You must include a prominent statement that
the Modification is derived, directly or indirectly, from Original
Code provided by the Initial Developer and including the name of the
Initial Developer in (a) the Source Code, and (b) in any notice in an
Executable version or related documentation in which You describe the
origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
   (a) Third Party Claims.
   If Contributor has knowledge that a license under a third party's
   intellectual property rights is required to exercise the rights
   granted by such Contributor under Sections 2.1 or 2.2,
   Contributor must include a text file with the Source Code
   distribution titled "LEGAL" which describes the claim and the
   party making the claim in sufficient detail that a recipient will
   know whom to contact. If Contributor obtains such knowledge after
   the Modification is made available as described in Section 3.2,
   Contributor shall promptly modify the LEGAL file in all copies
   Contributor makes available thereafter and shall take other steps
   (such as notifying appropriate mailing lists or newsgroups)
   reasonably calculated to inform those who received the Covered
   Code that new knowledge has been obtained.

   (b) Contributor APIs.
   If Contributor's Modifications include an application programming

interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c)   Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.

You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ

from this License are offered by You alone, not by the Initial
Developer or any Contributor. You hereby agree to indemnify the
Initial Developer and every Contributor for any liability incurred by
the Initial Developer or such Contributor as a result of any such
terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code
not governed by the terms of this License and distribute the Larger
Work as a single product. In such a case, You must make sure the
requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this
License with respect to some or all of the Covered Code due to
statute, judicial order, or regulation then You must: (a) comply with
the terms of this License to the maximum extent possible; and (b)
describe the limitations and the code they affect. Such description
must be included in the LEGAL file described in Section 3.4 and must
be included with all distributions of the Source Code. Except to the
extent prohibited by statute or regulation, such description must be
sufficiently detailed for a recipient of ordinary skill to be able to
understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has
attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised
and/or new versions of the License from time to time. Each version
will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the
License, You may always continue to use it under the terms of that
version. You may also choose to use such Covered Code under the terms
of any subsequent version of the License published by Netscape. No one
other than Netscape has the right to modify the terms applicable to
Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may
only do in order to apply it to code which is not already Covered Code

governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days

of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer

software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in

compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS"
basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
License for the specific language governing rights and limitations
under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____.
Portions created by _____ are Copyright (C) _____
_____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms
of the _____ license (the "[___] License"), in which case the
provisions of [_____] License are applicable instead of those
above. If you wish to allow use of your version of this file only
under the terms of the [____] License and not to allow others to use
your version of this file under the MPL, indicate your decision by
deleting the provisions above and replace them with the notice and
other provisions required by the [___] License. If you do not delete
the provisions above, a recipient may use your version of this file
under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of
the notices in the Source Code files of the Original Code. You should
use the text of this Exhibit A rather than the text found in the
Original Code Source Code for Your Modifications.]
==
=========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from the 'org.apache.commons' to the 'clover.org.apache.commons'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.
==
=========================

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Public Domain Dedication
This license is acceptable for Free Cultural Works.

Copyright-Only Dedication (based on United States law) or Public Domain Certification

The person or persons who have associated work with this document (the "Dedicator" or "Certifier") hereby either (a) certifies that, to the best of his knowledge, the work of authorship identified is in the public domain of the country from which the work is published, or (b) hereby dedicates whatever copyright the dedicators holds in the work of authorship identified below (the "Work") to the public domain. A certifier, moreover, dedicates any copyright interest he may have in the associated work, and for these purposes, is described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.
==
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package name from 'com.google.json' to 'clover.com.google.json'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.
==

                          Apache License
                    Version 2.0, January 2004
                  http://www.apache.org/licenses/

  TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
    and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
============================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package names from 'org.jfree' to 'clover.org.jfree'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.
==
============================

   GNU LESSER GENERAL PUBLIC LICENSE
       Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
   59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

       Preamble

 The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

 This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

 When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

 To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for

you if you distribute copies of the library or if you modify it.

 For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

 We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

 To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

 Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

 Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

 When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

 We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many

libraries. However, the Lesser license provides advantages in certain
special circumstances.

 For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

 In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

 Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

 The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

  GNU LESSER GENERAL PUBLIC LICENSE
 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

 A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

 The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is

included without limitation in the term "modification".)

 "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

 Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

 1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

 You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

 2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices
  stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no
  charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a
  table of data to be supplied by an application program that uses
  the facility, other than as an argument passed when the facility
  is invoked, then you must make a good faith effort to ensure that,
  in the event an application does not supply such function or
  table, the facility still operates, and performs whatever part of
  its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

 3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

 Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

 This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

 4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany

it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by

the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

 14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

   NO WARRANTY

 15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

 16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

   END OF TERMS AND CONDITIONS

   How to Apply These Terms to Your New Libraries

 If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

 To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively

convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

 Yoyodyne, Inc., hereby disclaims all copyright interest in the
 library `Frob' (a library for tweaking knobs) written by James Random Hacker.

 <signature of Ty Coon>, 1 April 1990
 Ty Coon, President of Vice

That's all there is to it!
==
==========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package name from 'com.google.common' to 'clover.com.google.common'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.
==
==========================

                           Apache License
                     Version 2.0, January 2004
                   http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

==

=========================

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed

their package name from 'org.apache.commons' to 'clover.org.apache.commons'. This was necessary to

avoid potential name conflicts during instrumentation of a code using the original library when using Clover.

No source code of the original library was modified.

==

=========================

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,

"control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but

not limited to compiled object code, generated documentation,

and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)

with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
/*
File: Core.js

Description:

Provides common utility functions and the Class object used internally by the library.

Also provides the <TreeUtil> object for manipulating JSON tree structures

Some of the Basic utility functions and the Class system are based in the MooTools Framework
<http://mootools.net>. Copyright (c) 2006-2009 Valerio Proietti, <http://mad4milk.net/>. MIT license
<http://mootools.net/license.txt>.

Author:

Nicolas Garcia Belmonte

Copyright:

Copyright 2008-2009 by Nicolas Garcia Belmonte.

Homepage:

<http://thejit.org>

Version:

1.1.2

License:

BSD License

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE

WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR

COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR

OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

http://www.atlassian.com/dms/wac/Atlassian_EULA_4-2.pdf

/*

* The Apache Software License, Version 1.1

*

* Copyright (c) 2000-2003 The Apache Software Foundation. All rights

* reserved.

*

* Redistribution and use in source and binary forms, with or without

* modification, are permitted provided that the following conditions

* are met:

*

* 1. Redistributions of source code must retain the above copyright

*   notice, this list of conditions and the following disclaimer.

*

```
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in
*    the documentation and/or other materials provided with the
*    distribution.
*
* 3. The end-user documentation included with the redistribution, if
*    any, must include the following acknowlegement:
*       "This product includes software developed by the
*        Apache Software Foundation (http://www.apache.org/)."
*    Alternately, this acknowlegement may appear in the software itself,
*    if and wherever such third-party acknowlegements normally appear.
*
* 4. The names "Ant" and "Apache Software
*    Foundation" must not be used to endorse or promote products derived
*    from this software without prior written permission. For written
*    permission, please contact apache@apache.org.
*
* 5. Products derived from this software may not be called "Apache"
*    nor may "Apache" appear in their names without prior written
*    permission of the Apache Group.
*
* THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
* WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
* ====================================================================
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation.  For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*/
        GNU LESSER GENERAL PUBLIC LICENSE
                   Version 3, 29 June 2007
```

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the

function or data, the facility still operates, and performs
whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of
this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the
Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license
document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

a) Give prominent notice with each copy of the Combined Work that
the Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license
document.

c) For a Combined Work that displays copyright notices during
execution, include the copyright notice for the Library among
these notices, as well as a reference directing the user to the
copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this
License, and the Corresponding Application Code in a form
suitable for, and under terms that permit, the user to
recombine or relink the Application with a modified version of

the Linked Version to produce a modified Combined Work, in the
manner specified by section 6 of the GNU GPL for conveying
Corresponding Source.

1) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (a) uses at run time
a copy of the Library already present on the user's computer
system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.

e) Provide Installation Information, but only if you would otherwise
be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based
on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version

of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

 If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
Javolution - Java(TM) Solution for Real-Time and Embedded Systems
Copyright (c) 2006, Javolution (http://javolution.org)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright notice,
    this list of conditions and the following disclaimer in the documentation
    and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in
   the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any,
   must include the following acknowledgment:

 "This product includes software developed by the Indiana University
  Extreme! Lab (http://www.extreme.indiana.edu/)."

Alternately, this acknowledgment may appear in the software itself,
if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab"
must not be used to endorse or promote products derived from this
software without prior written permission. For written permission,
please contact http://www.extreme.indiana.edu/.

5. Products derived from this software may not use "Indiana Univeristy"
name nor may "Indiana Univeristy" appear in their name, without prior
written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

/**
* CruiseControl, a Continuous Integration Toolkit
* Copyright (c) 2001-2003, ThoughtWorks, Inc.
* 651 W Washington Ave. Suite 500
* Chicago, IL 60661 USA
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
*   + Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.

```
*
*     + Redistributions in binary form must reproduce the above
*       copyright notice, this list of conditions and the following
*       disclaimer in the documentation and/or other materials provided
*       with the distribution.
*
*     + Neither the name of ThoughtWorks, Inc., CruiseControl, nor the
*       names of its contributors may be used to endorse or promote
*       products derived from this software without specific prior
*       written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
* A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
******************************************************************************/
```

Apache Velocity

Copyright (C) 2000-2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
Apache Ant
Copyright 1999-2015 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

The <sync> task is based on code Copyright (c) 2002, Landmark
Graphics Corp that has been kindly donated to the Apache Software
Foundation.

```
/*
*                                 Apache License
*                           Version 2.0, January 2004
*                        http://www.apache.org/licenses/
*
*   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
*
*   1. Definitions.
*
*      "License" shall mean the terms and conditions for use, reproduction,
```

```
 *     and distribution as defined by Sections 1 through 9 of this document.
 *
 *     "Licensor" shall mean the copyright owner or entity authorized by
 *     the copyright owner that is granting the License.
 *
 *     "Legal Entity" shall mean the union of the acting entity and all
 *     other entities that control, are controlled by, or are under common
 *     control with that entity. For the purposes of this definition,
 *     "control" means (i) the power, direct or indirect, to cause the
 *     direction or management of such entity, whether by contract or
 *     otherwise, or (ii) ownership of fifty percent (50%) or more of the
 *     outstanding shares, or (iii) beneficial ownership of such entity.
 *
 *     "You" (or "Your") shall mean an individual or Legal Entity
 *     exercising permissions granted by this License.
 *
 *     "Source" form shall mean the preferred form for making modifications,
 *     including but not limited to software source code, documentation
 *     source, and configuration files.
 *
 *     "Object" form shall mean any form resulting from mechanical
 *     transformation or translation of a Source form, including but
 *     not limited to compiled object code, generated documentation,
 *     and conversions to other media types.
 *
 *     "Work" shall mean the work of authorship, whether in Source or
 *     Object form, made available under the License, as indicated by a
 *     copyright notice that is included in or attached to the work
 *     (an example is provided in the Appendix below).
 *
 *     "Derivative Works" shall mean any work, whether in Source or Object
 *     form, that is based on (or derived from) the Work and for which the
 *     editorial revisions, annotations, elaborations, or other modifications
 *     represent, as a whole, an original work of authorship. For the purposes
 *     of this License, Derivative Works shall not include works that remain
 *     separable from, or merely link (or bind by name) to the interfaces of,
 *     the Work and Derivative Works thereof.
 *
 *     "Contribution" shall mean any work of authorship, including
 *     the original version of the Work and any modifications or additions
 *     to that Work or Derivative Works thereof, that is intentionally
 *     submitted to Licensor for inclusion in the Work by the copyright owner
 *     or by an individual or Legal Entity authorized to submit on behalf of
 *     the copyright owner. For the purposes of this definition, "submitted"
 *     means any form of electronic, verbal, or written communication sent
 *     to the Licensor or its representatives, including but not limited to
 *     communication on electronic mailing lists, source code control systems,
 *     and issue tracking systems that are managed by, or on behalf of, the
```

```
*      Licensor for the purpose of discussing and improving the Work, but
*      excluding communication that is conspicuously marked or otherwise
*      designated in writing by the copyright owner as "Not a Contribution."
*
*      "Contributor" shall mean Licensor and any individual or Legal Entity
*      on behalf of whom a Contribution has been received by Licensor and
*      subsequently incorporated within the Work.
*
*   2. Grant of Copyright License. Subject to the terms and conditions of
*      this License, each Contributor hereby grants to You a perpetual,
*      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
*      copyright license to reproduce, prepare Derivative Works of,
*      publicly display, publicly perform, sublicense, and distribute the
*      Work and such Derivative Works in Source or Object form.
*
*   3. Grant of Patent License. Subject to the terms and conditions of
*      this License, each Contributor hereby grants to You a perpetual,
*      worldwide, non-exclusive, no-charge, royalty-free, irrevocable
*      (except as stated in this section) patent license to make, have made,
*      use, offer to sell, sell, import, and otherwise transfer the Work,
*      where such license applies only to those patent claims licensable
*      by such Contributor that are necessarily infringed by their
*      Contribution(s) alone or by combination of their Contribution(s)
*      with the Work to which such Contribution(s) was submitted. If You
*      institute patent litigation against any entity (including a
*      cross-claim or counterclaim in a lawsuit) alleging that the Work
*      or a Contribution incorporated within the Work constitutes direct
*      or contributory patent infringement, then any patent licenses
*      granted to You under this License for that Work shall terminate
*      as of the date such litigation is filed.
*
*   4. Redistribution. You may reproduce and distribute copies of the
*      Work or Derivative Works thereof in any medium, with or without
*      modifications, and in Source or Object form, provided that You
*      meet the following conditions:
*
*      (a) You must give any other recipients of the Work or
*          Derivative Works a copy of this License; and
*
*      (b) You must cause any modified files to carry prominent notices
*          stating that You changed the files; and
*
*      (c) You must retain, in the Source form of any Derivative Works
*          that You distribute, all copyright, patent, trademark, and
*          attribution notices from the Source form of the Work,
*          excluding those notices that do not pertain to any part of
*          the Derivative Works; and
*
```

```
 *      (d) If the Work includes a "NOTICE" text file as part of its
 *          distribution, then any Derivative Works that You distribute must
 *          include a readable copy of the attribution notices contained
 *          within such NOTICE file, excluding those notices that do not
 *          pertain to any part of the Derivative Works, in at least one
 *          of the following places: within a NOTICE text file distributed
 *          as part of the Derivative Works; within the Source form or
 *          documentation, if provided along with the Derivative Works; or,
 *          within a display generated by the Derivative Works, if and
 *          wherever such third-party notices normally appear. The contents
 *          of the NOTICE file are for informational purposes only and
 *          do not modify the License. You may add Your own attribution
 *          notices within Derivative Works that You distribute, alongside
 *          or as an addendum to the NOTICE text from the Work, provided
 *          that such additional attribution notices cannot be construed
 *          as modifying the License.
 *
 *      You may add Your own copyright statement to Your modifications and
 *      may provide additional or different license terms and conditions
 *      for use, reproduction, or distribution of Your modifications, or
 *      for any such Derivative Works as a whole, provided Your use,
 *      reproduction, and distribution of the Work otherwise complies with
 *      the conditions stated in this License.
 *
 *   5. Submission of Contributions. Unless You explicitly state otherwise,
 *      any Contribution intentionally submitted for inclusion in the Work
 *      by You to the Licensor shall be under the terms and conditions of
 *      this License, without any additional terms or conditions.
 *      Notwithstanding the above, nothing herein shall supersede or modify
 *      the terms of any separate license agreement you may have executed
 *      with Licensor regarding such Contributions.
 *
 *   6. Trademarks. This License does not grant permission to use the trade
 *      names, trademarks, service marks, or product names of the Licensor,
 *      except as required for reasonable and customary use in describing the
 *      origin of the Work and reproducing the content of the NOTICE file.
 *
 *   7. Disclaimer of Warranty. Unless required by applicable law or
 *      agreed to in writing, Licensor provides the Work (and each
 *      Contributor provides its Contributions) on an "AS IS" BASIS,
 *      WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 *      implied, including, without limitation, any warranties or conditions
 *      of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
 *      PARTICULAR PURPOSE. You are solely responsible for determining the
 *      appropriateness of using or redistributing the Work and assume any
 *      risks associated with Your exercise of permissions under this License.
 *
 *   8. Limitation of Liability. In no event and under no legal theory,
```

```
 *      whether in tort (including negligence), contract, or otherwise,
 *      unless required by applicable law (such as deliberate and grossly
 *      negligent acts) or agreed to in writing, shall any Contributor be
 *      liable to You for damages, including any direct, indirect, special,
 *      incidental, or consequential damages of any character arising as a
 *      result of this License or out of the use or inability to use the
 *      Work (including but not limited to damages for loss of goodwill,
 *      work stoppage, computer failure or malfunction, or any and all
 *      other commercial damages or losses), even if such Contributor
 *      has been advised of the possibility of such damages.
 *
 *   9. Accepting Warranty or Additional Liability. While redistributing
 *      the Work or Derivative Works thereof, You may choose to offer,
 *      and charge a fee for, acceptance of support, warranty, indemnity,
 *      or other liability obligations and/or rights consistent with this
 *      License. However, in accepting such obligations, You may act only
 *      on Your own behalf and on Your sole responsibility, not on behalf
 *      of any other Contributor, and only if You agree to indemnify,
 *      defend, and hold each Contributor harmless for any liability
 *      incurred by, or claims asserted against, such Contributor by reason
 *      of your accepting any such warranty or additional liability.
 *
 *   END OF TERMS AND CONDITIONS
 *
 *   APPENDIX: How to apply the Apache License to your work.
 *
 *      To apply the Apache License to your work, attach the following
 *      boilerplate notice, with the fields enclosed by brackets "[]"
 *      replaced with your own identifying information. (Don't include
 *      the brackets!)  The text should be enclosed in the appropriate
 *      comment syntax for the file format. We also recommend that a
 *      file or class name and description of purpose be included on the
 *      same "printed page" as the copyright notice for easier
 *      identification within third-party archives.
 *
 *   Copyright [yyyy] [name of copyright owner]
 *
 *   Licensed under the Apache License, Version 2.0 (the "License");
 *   you may not use this file except in compliance with the License.
 *   You may obtain a copy of the License at
 *
 *       http://www.apache.org/licenses/LICENSE-2.0
 *
 *   Unless required by applicable law or agreed to in writing, software
 *   distributed under the License is distributed on an "AS IS" BASIS,
 *   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 *   See the License for the specific language governing permissions and
 *   limitations under the License.
```

*/

W3C SOFTWARE NOTICE AND LICENSE

http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

This work (and included software, documentation such as READMEs, or other
related items) is being provided by the copyright holders under the following
license. By obtaining, using and/or copying this work, you (the licensee) agree
that you have read, understood, and will comply with the following terms and
conditions.

Permission to copy, modify, and distribute this software and its documentation,
with or without modification, for any purpose and without fee or royalty is
hereby granted, provided that you include the following on ALL copies of the
software and documentation or portions thereof, including modifications:

  1. The full text of this NOTICE in a location viewable to users of the
     redistributed or derivative work.
  2. Any pre-existing intellectual property disclaimers, notices, or terms
     and conditions. If none exist, the W3C Software Short Notice should be
     included (hypertext is preferred, text is permitted) within the body
     of any redistributed or derivative code.
  3. Notice of any changes or modifications to the files, including the date
     changes were made. (We recommend you provide URIs to the location from
     which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE
NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT
THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY
PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or
publicity pertaining to the software without specific, written prior permission.
Title to copyright in this software and any associated documentation will at
all times remain with copyright holders.

_____

This formulation of W3C's notice and license became active on December 31 2002.
This version removes the copyright ownership notice such that this license can
be used with materials other than those owned by the W3C, reflects that ERCIM
is now a host of the W3C, includes references to this specific dated version of
the license, and removes the ambiguous grant of "use". Otherwise, this version
is the same as the previous version and is written so as to preserve the Free

Software Foundation's assessment of GPL compatibility and OSI's certification under the Open Source Definition. Please see our Copyright FAQ for common questions about using materials from our site, including specific terms and conditions for packages like libwww, Amaya, and Jigsaw. Other questions about this notice can be directed to site-policy@w3.org.

Joseph Reagle <site-policy@w3.org>

This license came from: http://www.megginson.com/SAX/copying.html
 However please note future versions of SAX may be covered
 under http://saxproject.org/?selected=pd

SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for XML), and release all of the SAX 2.0 source code, compiled code, and documentation contained in this distribution into the Public Domain. SAX comes with NO WARRANTY or guarantee of fitness for any purpose.

David Megginson, david@megginson.com
2000-05-05
.!"clover/org/jfree/ui/about/Licencesjava/lang/Object
Licences.javaGPLLjava/lang/String;DGNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.
To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically

terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are

different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
LGPLeSGNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to

use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".
A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.
1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

*   a) The modified work must itself be a software library.

* b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

* c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

* d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

*   a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
*   b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
*   c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
*   d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
*   e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
* b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
 singleton$Lclover/org/jfree/ui/about/Licences;<init>()V
thisgetInstance&()Lclover/org/jfree/ui/about/Licences;

getGPL()Ljava/lang/String;getLGPL
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!

/*9 4

Y$%'- 1-;

.%clover/org/jfree/ui/about/Contributorjava/lang/ObjectContributor.javanameLjava/lang/String;email<init>'(Ljava/lang/String;Ljava/lang/String;)V()V

this'Lclover/org/jfree/ui/about/Contributor;getName()Ljava/lang/String;getEmailCodeLocalVariableTableLineNumberTable

SourceFile!

Y*

+, DE FG/*O/*X

GNU LIBRARY GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is
numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some
specially designated Free Software Foundation software, and to any
other libraries whose authors decide to use it. You can use it for
your libraries, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if
you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link a program with the library, you must provide

complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.
.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free

libraries that are incorporated in them. (We have not seen how to achieve
this as regards changes in header files, but we have achieved it as regards
changes in the actual functions of the Library.) The hope is that this
will lead to faster development of free libraries.

 The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, while the latter only
works together with the library.

 Note that it is possible for a library to be covered by the ordinary
General Public License rather than by this special one.
.

   GNU LIBRARY GENERAL PUBLIC LICENSE
 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library which
contains a notice placed by the copyright holder or other authorized
party saying it may be distributed under the terms of this Library
General Public License (also called "this License"). Each licensee is
addressed as "you".

 A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

 The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

 "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

 Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

 You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.
.
 2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices
  stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no
  charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a
  table of data to be supplied by an application program that uses
  the facility, other than as an argument passed when the facility
  is invoked, then you must make a good faith effort to ensure that,
  in the event an application does not supply such function or
  table, the facility still operates, and performs whatever part of
  its purpose remains meaningful.

  (For example, a function in a library to compute square roots has
  a purpose that is entirely well-defined independent of the
  application. Therefore, Subsection 2d requires that any
  application-supplied function or table used by this function must
  be optional: if the application does not supply it, the square
  root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

 3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.
.
 Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

 This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

 4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

 If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

 5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.
.
6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified

executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

d) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

 For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the source code distributed need not include anything that is normally
distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

 It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.
.
 7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

  a) Accompany the combined library with a copy of the same work
  based on the Library, uncombined with any other library
  facilities. This must be distributed under the terms of the
  Sections above.

  b) Give prominent notice with the combined library of the fact
  that part of it is a work based on the Library, and explaining
  where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed

through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

 12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

 13. The Free Software Foundation may publish revised and/or new
versions of the Library General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.
.
 14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

  NO WARRANTY

 15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME

THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

 16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
MOZILLA PUBLIC LICENSE
Version 1.1

----

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

    A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

    B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

  2. Source Code License.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of

the Covered Code.

(d)   Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the

origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

   (a) Third Party Claims.

If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

   (b) Contributor APIs.

If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

   (c)   Representations.

Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the

Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the LEGAL file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has

attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised
and/or new versions of the License from time to time. Each version
will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the
License, You may always continue to use it under the terms of that
version. You may also choose to use such Covered Code under the terms
of any subsequent version of the License published by Netscape. No one
other than Netscape has the right to modify the terms applicable to
Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may
only do in order to apply it to code which is not already Covered Code
governed by this License), You must (a) rename Your license so that
the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape",
"MPL", "NPL" or any confusingly similar phrase do not appear in your
license (except to note that your license differs from this License)
and (b) otherwise make it clear that Your version of the license
contains terms which differ from the Mozilla Public License and
Netscape Public License. (Filling in the name of the Initial
Developer, Original Code or Contributor in the notice described in
Exhibit A shall not of themselves be deemed to be modifications of
this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS,
WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING,
WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF
DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING.
THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE
IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT,
YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE
COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER
OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF
ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate
automatically if You fail to comply with terms herein and fail to cure

such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____.
Portions created by _____ are Copyright (C) _____
_____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [_____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of

the notices in the Source Code files of the Original Code. You should

use the text of this Exhibit A rather than the text found in the

Original Code Source Code for Your Modifications.]

/*

* The contents of this file are subject to the Mozilla Public License Version 1.1

* (the "License"); you may not use this file except in compliance with the License.

* You may obtain a copy of the License at http://www.mozilla.org/MPL/

*

* Software distributed under the License is distributed on an "AS IS" basis,

* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License

* for the specific language governing rights and limitations under the License.

*

* The Original Code is 'iText, a free JAVA-PDF library'.

*

* The Initial Developer of the Original Code is Bruno Lowagie. Portions created by

* the Initial Developer are Copyright (C) 1999, 2000, 2001, 2002 by Bruno Lowagie.

* All Rights Reserved.

* Co-Developer of the code is Paulo Soares. Portions created by the Co-Developer

* are Copyright (C) 2000, 2001, 2002 by Paulo Soares. All Rights Reserved.

*

* Contributor(s): all the names of the contributors are added in the source code

* where applicable.

*

* Alternatively, the contents of this file may be used under the terms of the

* LGPL license (the "GNU LIBRARY GENERAL PUBLIC LICENSE"), in which case the

* provisions of LGPL are applicable instead of those above. If you wish to

* allow use of your version of this file only under the terms of the LGPL

* License and not to allow others to use your version of this file under

* the MPL, indicate your decision by deleting the provisions above and

* replace them with the notice and other provisions required by the LGPL.

* If you do not delete the provisions above, a recipient may use your version

* of this file under either the MPL or the GNU LIBRARY GENERAL PUBLIC LICENSE.

*

* This library is free software; you can redistribute it and/or modify it

* under the terms of the MPL as stated above or under the terms of the GNU

* Library General Public License as published by the Free Software Foundation;

* either version 2 of the License, or any later version.

*

* This library is distributed in the hope that it will be useful, but WITHOUT

* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS

* FOR A PARTICULAR PURPOSE. See the GNU Library general Public License for more

* details.

*

* If you didn't download this code from the following link, you should check if

* you aren't using an obsolete version:

* http://www.lowagie.com/iText/

*

* This class is generated based on a grammar file privided by SUN, and updated

* by Carsten Hammer. SUN's license agreement can be found at this URL:
* http://java.sun.com/products/java-media/2D/samples/samples-license.html
* See also the file sun.txt in directory com.lowagie.text.pdf
*/
0(clover/com/atlassian/license/LicensePairjava/lang/ObjectLicensePair.javaNEW_LICENSE_PREFIX[Blicensehash
originalLicenseStringLjava/lang/String;isNGZ<init>([B[B)V-clover/com/atlassian/license/LicenseException()V

startsWith([B[B)Z

 =clover/com/atlassian/extras/decoder/v2/Version2LicenseDecoder!packLicense([B[B)Ljava/lang/String;#$
"%
packV1License'$
(
*this*Lclover/com/atlassian/license/LicensePair;([B[BLjava/lang/String;)VtextoriginalStringiItargetprefix'(Ljava/la
ng/String;Ljava/lang/String;)Vjava/lang/Exception6 License string or hash are null.8(Ljava/lang/String;)V:
;)clover/com/atlassian/license/LicenseUtils=getBytes(Ljava/lang/String;)[B?@
>Ajava/lang/StringBufferC
DException generating license: Fappend,(Ljava/lang/String;)Ljava/lang/StringBuffer;HI
DJ,(Ljava/lang/Object;)Ljava/lang/StringBuffer;HL
DMtoString()Ljava/lang/String;OP
DQeLjava/lang/Exception;contactLicense was
nullUsplitVersion2License>(Ljava/lang/String;)Lclover/com/atlassian/license/LicensePair;WX
Y=clover/com/atlassian/extras/decoder/v1/Version1LicenseDecoder[splitLicense]X
\^
concatLicensepair getString([B)Ljava/lang/String;bc
>djava/lang/Stringflength()Ihi
gj substring(II)Ljava/lang/String;lm
gn
p(I)Ljava/lang/String;lr
gssbLjava/lang/StringBuffer;
hashString
lineLength
licenseStrjava/io/IOExceptionz
" canDecode(Ljava/lang/String;)Z}~
"lastIndexOf(I)I
g()[B?
g-clover/org/apache/commons/codec/binary/Base64decodeBase64([B)[B
java/io/ByteArrayInputStream([B)V
java/io/DataInputStream(Ljava/io/InputStream;)V
readInti
read([B)I
 availablei
.
(Ljava/lang/Throwable;)V
licenseContentdecodedBytesinLjava/io/ByteArrayInputStream;dInLjava/io/DataInputStream;
textLengthlicenseTextLjava/io/IOException;encodedLicensepos()Z
getLicensegetLicenseStringgetHash
getHashStringgetOriginalLicenseString<clinit>CodeLocalVariableTableLineNumberTable

Exceptions
SourceFile!

C**+*,*** ** **&***)+ C,-CC "$ %&'B(.#**+*,*** *-+*#,-#/# #0./ 012"3$,+>,+3,3*12$,-$3$478 9;<9">5z*+,
Y9<*+B*,B*** ** **&***)+NYDYEGK-NR<Z]7*^STz,-zz .FGINO&P5QZV]S^UyW:B*+
YV<*+ZM,+_M*,++*,*,*, B,-B`*a-.`acfhj!m)n1o9pAq'$8DYEN,e:kl6k#-oKW-qKWt:-KW-qKW+e:k#-oKW-
qKWt:-KW-qKW-RH,- uvw{x2U=yFuwxz!|.}5~AHOU_lsWXH
z"Y|++X=+oN-:Y:Y:6:W: WY +NY-op{pR'I2>=3D,2J&[q Sz,-zc2>'2=DJR[cpq/* ,-/*,-P2*e,-/*,-P2*e,-P/*+,-
OP/*+,-8 Y
TYTYTY
TYT
0!2clover/com/atlassian/license/MemoryLicenseRegistryjava/lang/Object,clover/com/atlassian/license/LicenseRegis
tryMemoryLicenseRegistry.javaLICENSELjava/lang/String;HASH<init>()V

this4Lclover/com/atlassian/license/MemoryLicenseRegistry;setLicenseMessage(Ljava/lang/String;)V
licenseMessagesetLicenseHash
  licenseHashgetLicenseMessage()Ljava/lang/String;getLicenseHashCodeLocalVariableTableLineNumberTable
SourceFile!

 /*=+

=+
..
03clover/com/atlassian/license/decoder/LicenseAdaptorjava/lang/Object$clover/com/atlassian/license/LicenseLicen
seAdaptor.java_1_YEARJ\licenseType*Lclover/com/atlassian/license/LicenseType;creationDateLjava/util/Date;pur
chaseDate
expiryDatemaintenanceExpiryDate
evaluationZsupportEntitlementNumberLjava/lang/String;permittedClusterNodesIorganisationpartnermaximumNum
berOfUsers<init>h(Lclover/com/atlassian/extras/common/util/LicenseProperties;Lclover/com/atlassian/license/Lice
nseType;)V()V

!CreationDate#9clover/com/atlassian/extras/common/util/LicenseProperties%getProperty&(Ljava/lang/String;)Ljav
a/lang/String;'(&)-clover/com/atlassian/extras/common/DateEditor+getDate$(Ljava/lang/String;)Ljava/util/Date;-.
,/ 1LicenseExpiryDate3 5
Evaluation7java/lang/Boolean9valueOf'(Ljava/lang/String;)Ljava/lang/Boolean;;<
:=booleanValue()Z?@
:A CPurchaseDateE GMaintenanceExpiryDateI KSENM
ONumberOfClusterNodesQgetInt(Ljava/lang/String;I)IST&U WOrganisationY [PartnerName] _
NumberOfUsersa
cthis5Lclover/com/atlassian/license/decoder/LicenseAdaptor;licenseProperties;Lclover/com/atlassian/extras/commo
n/util/LicenseProperties;maxUsersgetDateCreated()Ljava/util/Date;java/util/DatelgetTime()Jno
mp(J)Vr
msupdatedCreationDategetDatePurchased
getExpiryDate(clover/com/atlassian/license/LicenseTypexexpiresz@y{getLicenseDurationgetLicenseId()Ljava/lang
/String;getLicenseType,()Lclover/com/atlassian/license/LicenseType;getOrganisationgetPartnerNamegetPermittedC
lusteredNodes()IgetUsersrequiresUserLimit@y isExpiredjava/lang/SystemcurrentTimeMilliso
isLicenseLevel(Ljava/util/Collection;)Z

getDescriptionyjava/lang/StringtoLowerCase

java/util/Collectioniterator()Ljava/util/Iterator;java/util/IteratorhasNext@next()Ljava/lang/Object;indexOf(Ljava/lang/String;)I

levelLjava/util/Iterator;levelsLjava/util/Collection;descriptiongetSupportEntitlementNumber

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile!

* *,"*+$*02*+4*06*+8*>BD*+F*0H*+J*0L*+N*P*+RVX*+Z**+^*`+bV>* 'd*efgh

i: !'"9#H$W%c&p'|()*+jkx.*LmY*Lq

etLmY*2qtL+ u.ef,u026,8vk/*Hef=wkF*"|

*D

*6efB}o, efG~/*PefL/*"efQ/*\efV/*`ef[/*Xef`E*"*defegk@E*6*6qefq<*M+N- -:,4(&<ef<

/v

wy(z5|7~:/*Pef

03clover/com/atlassian/license/decoder/LicenseDecoderjava/lang/ObjectLicenseDecoder.java1clover/com/atlassian/extras/common/log/Logger$Log-

clover/com/atlassian/extras/common/log/LoggerLoglog3Lclover/com/atlassian/extras/common/log/Logger$Log;DURATION_PREFIXLjava/lang/String; Duration:JIRA_APPLICATION_NAMEJIRACONF_APPLICATION_NAME

CONFLUENCE9class$clover$com$atlassian$license$decoder$LicenseDecoderLjava/lang/Class;<init>()V

this5Lclover/com/atlassian/license/decoder/LicenseDecoder;

getLicensed(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)Lclover/com/atlassian/license/License;-clover/com/atlassian/license/LicenseException!getPublicKey-(Ljava/lang/String;)Ljava/security/PublicKey;#$

%loadLicense}(Lclover/com/atlassian/license/LicensePair;Ljava/security/PublicKey;Ljava/lang/String;)Lclover/com/atlassian/license/License;'(

)java/lang/RuntimeException+

getMessage()Ljava/lang/String;-.

"/*(Ljava/lang/String;Ljava/lang/Throwable;)V1

,2e/Lclover/com/atlassian/license/LicenseException;pair*Lclover/com/atlassian/license/LicensePair;applicationNam

eisValid?(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)Z

;java/lang/Exception=getPublicKeyFilename&(Ljava/lang/String;)Ljava/lang/String;?@

ACjava/lang/StringEequals(Ljava/lang/Object;)ZGH

FI]The filename for the public key is null. This must be set before a public key can be

located.K(Ljava/lang/String;)VM

"NloadPublicKeyFromFileP$

Q Sjava/lang/StringBufferU

V!Exception looking up public key: Xappend,(Ljava/lang/String;)Ljava/lang/StringBuffer;Z[

V\

>/toString_.

V`error*(Ljava/lang/Object;Ljava/lang/Throwable;)VbcdfException getting verification from file - possible

classloader problem, or corrupt JIRA installation

fLjava/lang/Exception;publicKeyFileNamejava/io/IOExceptionj&java/security/NoSuchAlgorithmExceptionl*java/security/spec/InvalidKeySpecExceptionnjava/lang/Threadp

currentThread()Ljava/lang/Thread;rs

qtgetContextClassLoader()Ljava/lang/ClassLoader;vw

qxjava/lang/ClassLoaderzgetResourceAsStream)(Ljava/lang/String;)Ljava/io/InputStream;|}

{~ getClass()Ljava/lang/Class;

java/lang/ClassgetComponentType

getClassLoaderw

)clover/com/atlassian/license/LicenseUtilsreadKey(Ljava/io/InputStream;)[B

java/io/InputStreamclose

%java/security/spec/X509EncodedKeySpec([B)V

DSAjava/security/KeyFactorygetInstance.(Ljava/lang/String;)Ljava/security/KeyFactory;

generatePublic7(Ljava/security/spec/KeySpec;)Ljava/security/PublicKey;

keyfisLjava/io/InputStream;

contextLoaderLjava/lang/ClassLoader;encKey[B

pubKeySpec'Ljava/security/spec/X509EncodedKeySpec;

keyFactoryLjava/security/KeyFactory;(clover/com/atlassian/license/LicensePairisNG()Z

parseNewLicense

parseOldLicense(

(Ljava/lang/Object;)Vb publicKeyLjava/security/PublicKey;!java/security/InvalidKeyException

java/security/SignatureExceptionSHA1withDSAjava/security/Signature-

(Ljava/lang/String;)Ljava/security/Signature;

initVerify(Ljava/security/PublicKey;)V

()[B

update

getHash

verify([B)Z

getDecodedMessage([B)Ljava/lang/String;

java/util/StringTokenizer^'(Ljava/lang/String;Ljava/lang/String;)V

hasMoreTokens

 nextToken.

java/lang/IntegerparseInt(Ljava/lang/String;)I

+clover/com/atlassian/license/LicenseManager/()Lclover/com/atlassian/license/LicenseManager;

getLicenseType?(Ljava/lang/String;I)Lclover/com/atlassian/license/LicenseType;

-clover/com/atlassian/extras/common/DateEditorgetDate$(Ljava/lang/String;)Ljava/util/Date;

(clover/com/atlassian/license/LicenseTypeexpires

getOriginalLicenseString.

getLicenseIdFromLicenseString@

requiresUserLimit\|split'(Ljava/lang/String;)[Ljava/lang/String;

F%License contained invalid user

limit:+clover/com/atlassian/license/DefaultLicense(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String

;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;ILjava/lang/String;)V

"Signature did not verify

properly."warn$%usersAndClustersLimitsparts[Ljava/lang/String;licenseTypeCodeIlicenseType*Lclover/com/atlas

sian/license/LicenseType;dateCreatedLjava/util/Date;

datePurchaseddateExpiresorganisation licenseIdusersclusterCountpartnerName

messageString tokenizerLjava/util/StringTokenizer;

signatureLjava/security/Signature;=clover/com/atlassian/extras/decoder/v2/Version2LicenseDecoder<

= canDecode(Ljava/lang/String;)Z?@

=A Failed to decode as V2 license:

C,(Ljava/lang/Object;)Ljava/lang/StringBuffer;ZE

VFdecode*(Ljava/lang/String;)Ljava/util/Properties;HI

=J

lookupProduct=(Ljava/lang/String;)Lclover/com/atlassian/extras/api/Product;LM

N@clover/com/atlassian/extras/common/util/ProductLicensePropertiesPB(Lclover/com/atlassian/extras/api/Product;Ljava/util/Properties;)VR

QSLicenseEditionUgetPropertyW@

QXLicenseTypeZP(Ljava/lang/String;Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;\

]LicenseTypeName_

Evaluationajava/lang/BooleancvalueOf'(Ljava/lang/String;)Ljava/lang/Boolean;ef

dgbooleanValuei

dj@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolverlgetLicenseEditionD(Ljava/lang/String;)Lclover/com/atlassian/extras/api/LicenseEdition;no

mp>clover/com/atlassian/extras/legacy/util/OldLicenseTypeResolverr(Lclover/com/atlassian/extras/api/Product;Ljava/lang/String;ZLclover/com/atlassian/extras/api/LicenseEdition;)Lclover/com/atlassian/license/LicenseType;t

su3clover/com/atlassian/license/decoder/LicenseAdaptorwh(Lclover/com/atlassian/extras/common/util/LicenseProperties;Lclover/com/atlassian/license/LicenseType;)Vy

xzlicenseDecoder?Lclover/com/atlassian/extras/decoder/v2/Version2LicenseDecoder;propLjava/util/Properties;product)Lclover/com/atlassian/extras/api/Product;productPropertiesBLclover/com/atlassian/extras/common/util/ProductLicenseProperties;editionNamelicenseTypeString'clover/com/atlassian/extras/api/Productvalues,()[Lclover/com/atlassian/extras/api/Product;

getName.

equalsIgnoreCase@

FgetNamespace.

replace(CC)Ljava/lang/String;

F"java/lang/IllegalArgumentException Could not find product for key <>

Narr$*[Lclover/com/atlassian/extras/api/Product;len$i$keylength()I

F

-clover/com/atlassian/license/util/StringUtils

replaceAllJ(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;

  substring(II)Ljava/lang/String;

FtoUpperCase.

ForiginalLicenseString$java/io/UnsupportedEncodingExceptionUTF-8([BLjava/lang/String;)V

F(Ljava/lang/Throwable;)V

,&Ljava/io/UnsupportedEncodingException;messageX(Lclover/com/atlassian/license/LicensePair;Ljava/security/PublicKey;Ljava/lang/String;)ZgetLicenseTypeStoreC(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseTypeStore;

-clover/com/atlassian/license/LicenseTypeStoregetPublicKeyFileName.

<clinit>F(Ljava/lang/Class;)Lclover/com/atlassian/extras/common/log/Logger$Log;

ConstantValue SyntheticCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses!

/*# f*+&+*M,Y,0,3

"

45678.

02 9:B*+<6788

#$K*BL+D+J

"YLO+RMTVYWY],^]a,e"YgO !> ")4hK8Fi=?@D!F"HAI" P$WLuyM, ,*L+!YWY*L+N+Y-::>WiU N:HO.OP R

SVW5Z:[>]H^O`kmo

'(* *,*+,NT->
>*4h 67 8gk
ortu (-N-+-*-**:Y:6,::: :

:
:*:6
6X::' "YVYW]]aO26
26:
:Y

!T#&N'D()F*+Q,-[./e0/ h1/
234+
5+*6'7489-67--8':;!{|}'4<FQ[ehr| +
m"
X
=Y>M,*B"YVYWD]*GaO,*KN+O:QY-T:VY:[Y:+^:.`YbYhkqv:xY{f
n,-678|}8p~>jJ^TT,-^J2/8>JT^cq"
LM]L+=>0+2:** _YVYW]*]]a4"74+ 2+]25;
@f4*0*F'*DDD;ED4
1
ZFY*L,Y+
 4
 9M*+,* 678
?@5*85YWYT%

0Cclover/com/atlassian/license/applications/jira/JiraLicenseTypeStore-clover/com/atlassian/license/LicenseTypeStoreJiraLicenseTypeStore.javaAPPLICATION_NAMELjava/lang/String; JIRAJIRA_STANDARD_ACADEMIC*Lclover/com/atlassian/license/LicenseType;JIRA_STANDARD_EVALUATIONJIRA_STANDARD_NON_PROFITJIRA_STANDARD_FULL_LICENSEJIRA_STANDARD_COMMUNITYJIRA_STANDARD_OPEN_SOURCEJIRA_STANDARD_DEVELOPERJIRA_STANDARD_DEMONSTRATIONJIRA_STANDARD_PERSONALJIRA_STANDARD_STARTERJIRA_PROFESSIONAL_ACADEMICJIRA_PROFESSIONAL_EVALUATIONJIRA_PROFESSIONAL_NON_PROFITJIRA_PROFESSIONAL_FULL_LICENSEJIRA_PROFESSIONAL_COMMUNITYJIRA_PROFESSIONAL_OPEN_SOURCEJIRA_PROFESSIONAL_DEVELOPERJIRA_PROFESSIONAL_DEMONSTRATIONJIRA_PROFESSIONAL_PERSONALJIRA_PROFESSIONAL_STARTERJIRA_ENTERPRISE_ACADEMICJIRA_ENTERPRISE_EVALUATIONJIRA_ENTERPRISE_NON_PROFITJIRA_ENTERPRISE_FULL_LICENSEJIRA_ENTERPRISE_HOSTEDJIRA_ENTERPRISE_COMMUNITYJIRA_ENTERPRISE_OPEN_SOURCEJIRA_ENTERPRISE_DEVELOPERJIRA_ENTERPRISE_DEMONSTRATIONJIRA_ENTERPRISE_TESTINGJIRA_ENTERPRISE_PERSONALJIRA_ENTERPRISE_STARTERpublicKeyFileNameprivateKeyFileName<init>()V-.
/applicationLicenseTypesLjava/util/ArrayList;12 3
 5java/util/ArrayList7add(Ljava/lang/Object;)Z9:
8; =
 ? A C E G I K M O Q S U W Y [ ] _ a c e! g" i# k$ m% o& q' s(u) w*
ythisELclover/com/atlassian/license/applications/jira/JiraLicenseTypeStore;getAllLicenses()Ljava/util/Collection;getPublicKeyFileName()Ljava/lang/String;+ getPrivateKeyFileName,
<clinit>/clover/com/atlassian/license/DefaultLicenseTypeJIRA Standard:
Academic+clover/com/atlassian/extras/api/LicenseTypeACADEMIC-

Lclover/com/atlassian/extras/api/LicenseType; name

.clover/com/atlassian/extras/api/LicenseEditionSTANDARD0Lclover/com/atlassian/extras/api/LicenseEdition;

Z(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V-

JIRA Standard: Evaluation

COMMERCIAL 'JIRA Standard: Non-Profit / Open Source

NON_PROFIT JIRA Standard: Commercial ServerJIRA Standard: Community COMMUNITY JIRA Standard: Open

SourceOPEN_SOURCE JIRA Standard: Developer DEVELOPER JIRA Standard: Demonstration

DEMONSTRATION JIRA Standard: PersonalPERSONAL JIRA Standard: StarterSTARTER JIRA Professional:

AcademicPROFESSIONAL JIRA Professional: Evaluation+JIRA Professional: Non-Profit / Open Source$JIRA

Professional: Commercial ServerJIRA Professional: CommunityJIRA Professional: Open SourceJIRA Professional:

Developer JIRA Professional: DemonstrationJIRA Professional: PersonalJIRA Professional: StarterJIRA Enterprise:

Academic

ENTERPRISE JIRA Enterprise: Evaluation)JIRA Enterprise: Non-Profit / Open Source"JIRA Enterprise:

Commercial ServerJIRA Enterprise: HostedHOSTED JIRA Enterprise: CommunityJIRA Enterprise: Open

SourceJIRA Enterprise: DeveloperJIRA Enterprise: DemonstrationJIRA Enterprise: TestingTESTING

[(ILjava/lang/String;ZZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V-

JIRA Enterprise: PersonalJIRA Enterprise: Starter"clover/com/atlassian/jira/leaf.keyjira/jira.byte

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile!#

        ! " # $ % & ' ( ) * + ,-

.e*0*46<W*4><W*4@<W*4B<W*4D<W*4F<W*4H<W*4J<W*4L<W*4N<W*4P<W*4R<W*4T<W*4V<W*4

X<W*4Z<W*4\<W*4^<W*4`<W*4b<W*4d<W*4f<W*4h<W*4j<W*4l<W*4n<W*4p<W*4r<W*4t<W*4v<W*

4x<W*4z<We{| "6789%:0;;<F=Q>\?g@rB}CDEFGHIJKMNOPQR"S-T8UCVNWYXdY}~/*4{|].{| b.{| g.

BY6Ym>Y@YBYkDYzFYHYJYLYNY[PY/RYLTYWVY*XY'ZYR\Yc^YY`YZbY

dYfYhYjYlYnYpYrYtYvY!xY"z "3Mg3Le~ !"$%/&I'c(})*+,-./52;3

0nQclover/com/atlassian/license/applications/greenhopper/GreenHopperLicenseTypeStore-

clover/com/atlassian/license/LicenseTypeStore

GreenHopperLicenseTypeStore.javaNAMELjava/lang/String;GreenHopper!GREENHOPPER_STANDARD_FULL

_LICENSE*Lclover/com/atlassian/license/LicenseType;%GREENHOPPER_PROFESSIONAL_FULL_LICENSE#

GREENHOPPER_ENTERPRISE_FULL_LICENSE!GREENHOPPER_ENTERPRISE_EVALUATIONGREENH

OPPER_ENTERPRISE_ACADEMIC"GREENHOPPER_ENTERPRISE_OPEN_SOURCEGREENHOPPER_ENT

ERPRISE_PERSONAL<init>()V

applicationLicenseTypesLjava/util/ArrayList;

 java/util/ArrayListadd(Ljava/lang/Object;)Z

 "

 $ & (*

,thisSLclover/com/atlassian/license/applications/greenhopper/GreenHopperLicenseTypeStore;getPublicKeyFileNam

e()Ljava/lang/String;)clover/com/atlassian/greenhopper/leaf.key2getPrivateKeyFileNamegreenhopper/greenhopper.

byte5<clinit>/clover/com/atlassian/license/DefaultLicenseType8'GreenHopper Standard: Commercial

Server:+clover/com/atlassian/extras/api/LicenseType<

COMMERCIAL-Lclover/com/atlassian/extras/api/LicenseType;>? =@nameB1

=C.clover/com/atlassian/extras/api/LicenseEditionESTANDARD0Lclover/com/atlassian/extras/api/LicenseEdition;

GH FIZ(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)VK

9L+GreenHopper Professional: Commercial ServerNPROFESSIONALPH FQ)GreenHopper Enterprise:

Commercial ServerS

ENTERPRISEUH FV"GreenHopper Enterprise: EvaluationX GreenHopper Enterprise: AcademicZACADEMIC\?

=]#GreenHopper Enterprise: Open Source_OPEN_SOURCEa? =b GreenHopper Enterprise:

PersonaldPERSONALf? =g
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!i

jR**!W*#!W*%!W*'!W*)!W*+!W*-!WkR./l& %0; F!Q"01j-3k./l&41j-
6k./l+7j9Yx;ADJM9YyOADRM#9YzTADWM%9Y{YADWM'9Y|[^DWM)9Y}`cDWM+9Y~ehDWM-l4Nhm
0tKclover/com/atlassian/license/applications/crucible/CrucibleLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreCrucibleLicenseTypeStore.javaCRUCIBLE_ACADEMIC*Lclover/c
om/atlassian/license/LicenseType;CRUCIBLE_COMMERCIALCRUCIBLE_COMMUNITYCRUCIBLE_EVALU
ATIONCRUCIBLE_OPEN_SOURCECRUCIBLE_DEVELOPERCRUCIBLE_STARTERCRUCIBLE_DEMONS
TRATIONpublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
 " $
 & (*
 ,
.thisMLclover/com/atlassian/license/applications/crucible/CrucibleLicenseTypeStore;getAllLicenses()Ljava/util/Col
lection;getPublicKeyFileName()Ljava/lang/String; 6getPrivateKeyFileName
9<clinit>/clover/com/atlassian/license/DefaultLicenseType<Crucible:
Academic>+clover/com/atlassian/extras/api/LicenseType@ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;BC ADnameF5
AG*(ILjava/lang/String;ZZLjava/lang/String;)VI
=JCrucible: CommercialL
COMMERCIALNC AOCrucible: CommunityQ COMMUNITYSC ATCrucible: EvaluationVCrucible: Open
SourceXOPEN_SOURCEZC A[Crucible: Developer] DEVELOPER_C A`Crucible: StarterbSTARTERdC
AeCrucible: Demonstriationg
DEMONSTRATIONiC
Aj&clover/com/atlassian/crucible/leaf.keylcrucible/crucible.bytenCodeLocalVariableTableLineNumberTable
SourceFile!

  p]**!W*#!W*%!W*'!W*)!W*+!W*-!W*/!Wq]01r*
%0;
F!Q"\#23p/*q01r'45p.7q01r,85p.:q01r1;p=YL?EHK=YVMPHK#=Y`RUHK%=YjWPHK'=YtY\HK)=Y~^aHK+=Y
cfHK-=YhkHK/m7o:r*

.E\ss
0yUclover/com/atlassian/license/applications/sharepoint/SharePointPluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStore%SharePointPluginLicenseTypeStore.javaSP_PLUGIN_APPNAME
Ljava/lang/String;SharePoint
PluginAPPLICATION_NAMESHAREPOINT_ACADEMIC*Lclover/com/atlassian/license/LicenseType;SHAREP
OINT_EVALUATIONSHAREPOINT_DEMONSTRATIONSHAREPOINT_NON_PROFITSHAREPOINT_COM
MUNITYSHAREPOINT_DEVELOPERSHAREPOINT_OPEN_SOURCESHAREPOINT_FULL_LICENSEpublic
KeyFileNameprivateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayList add(Ljava/lang/Object;)Z"#
 !$
 & (* , . 0
2thisWLclover/com/atlassian/license/applications/sharepoint/SharePointPluginLicenseTypeStore;getAllLicenses()Lj
ava/util/Collection;getPublicKeyFileName()Ljava/lang/String; :getPrivateKeyFileName

=<clinit>/clover/com/atlassian/license/DefaultLicenseType@SharePoint:
AcademicB+clover/com/atlassian/extras/api/LicenseTypeDACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;FG EHnameJ9
EK*(ILjava/lang/String;ZZLjava/lang/String;)VM
ANSharePoint: EvaluationP
COMMERCIALRG ESSharePoint: DemonstrationU
DEMONSTRATIONWG EX$SharePoint: Non-Profit / Open SourceZ
NON_PROFIT\G E]SharePoint: Community_ COMMUNITYaG EbSharePoint: Developerd DEVELOPERfG
EgSharePoint: Open SourceiOPEN_SOURCEkG ElSharePoint: Commercialn-
clover/com/atlassian/confluence/page/Page.keypconfluence/confluence.byter
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!t
t
    u]**%W*'%W*)%W*+%W*-%W*/%W*1%W*3%Wv]45w*
%'()%*0+;,F-Q.\/67u/*v45w389u.;v45w8<9u.>v45w=?uAYCILOAYQTLO'AY(VYLO/AY2[^LO)AY<`cLO-
AYFehLO1AYPjmLO3AYZoTLO+q;s>w*
.E\s!"x
0tMclover/com/atlassian/license/applications/vssplugin/VSSPluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreVSSPluginLicenseTypeStore.javaVSS_ACADEMIC*Lclover/com/at
lassian/license/LicenseType;VSS_EVALUATIONVSS_NON_PROFITVSS_FULL_LICENSE
VSS_COMMUNITY
VSS_DEVELOPERVSS_DEMONSTRATIONVSS_OPEN_SOURCEpublicKeyFileNameLjava/lang/String;private
KeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
" $
& (*
,
.thisOLclover/com/atlassian/license/applications/vssplugin/VSSPluginLicenseTypeStore;getAllLicenses()Ljava/util/
Collection;getPublicKeyFileName()Ljava/lang/String; 6getPrivateKeyFileName
9<clinit>/clover/com/atlassian/license/DefaultLicenseType<
VSS: Academic>+clover/com/atlassian/extras/api/LicenseType@ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;BC ADnameF5
AG*(ILjava/lang/String;ZZLjava/lang/String;)VI
=JVSS: EvaluationL
COMMERCIALNC AOVSS: Non-Profit / Open SourceQ
NON_PROFITSC ATVSS: CommercialVVSS: CommunityX COMMUNITYZC A[VSS: Developer]
DEVELOPER_C A`VSS: Demonstrationb
DEMONSTRATIONdC AeVSS: Open SourcegOPEN_SOURCEiC
AjCclover/com/atlassian/license/applications/jira/JiraLicenseTypeStorel m6
m9CodeLocalVariableTableLineNumberTable
SourceFile!

  p]**!W*#!W*%!W*'!W*)!W*+!W*-!W*/!Wq]01r*
!"#%$0%;&F'Q(\)23p/*q01r-
45p.7q01r285p.:q01r7;p=Y@?EHK=YAMPHK#=YBRUHK%=YCWPHK'=YDY\HK)=YE^aHK+=YFcfHK-
=YGhkHK/n7o:r*
.E\ss

0lEclover/com/atlassian/license/applications/crowd/CrowdLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreCrowdLicenseTypeStore.javaCROWD_ACADEMIC*Lclover/com/a
tlassian/license/LicenseType;CROWD_COMMERCIALCROWD_COMMUNITYCROWD_EVALUATIONCRO
WD_OPEN_SOURCECROWD_DEVELOPERCROWD_DEMONSTRATIONpublicKeyFileNameLjava/lang/Strin
g;privateKeyFileName<init>()V

applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z

 ! #

 % ')

+thisGLclover/com/atlassian/license/applications/crowd/CrowdLicenseTypeStore;getAllLicenses()Ljava/util/Collect
ion;getPublicKeyFileName()Ljava/lang/String; 3getPrivateKeyFileName

6<clinit>/clover/com/atlassian/license/DefaultLicenseType9Crowd:

Academic;+clover/com/atlassian/extras/api/LicenseType=ACADEMIC-

Lclover/com/atlassian/extras/api/LicenseType;?@ >AnameC2

>D*(ILjava/lang/String;ZZLjava/lang/String;)VF

:GCrowd: CommercialI

COMMERCIALK@ >LCrowd: CommunityN COMMUNITYP@ >QCrowd: EvaluationSCrowd: Open

SourceUOPEN_SOURCEW@ >XCrowd: DeveloperZ DEVELOPER\@ >]Crowd: Demonstration_

DEMONSTRATIONa@

>b#clover/com/atlassian/crowd/leaf.keydcrowd/crowd.bytefCodeLocalVariableTableLineNumberTable

SourceFile!

 hR** W*" W*\$ W*& W*(W** W*, WiR-.j& %0; F!Q"/0h/*i-.j&12h.4i-.j+52h.7i-
.j08h:YY<BEH:YaJMEH":YiOREH\$:YqTMEH&:YyVYEH(:Y[^EH*:Y`cEH,e4g7j&

.E\sk

0tWclover/com/atlassian/license/applications/perforceplugin/PerforcePluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStore#PerforcePluginLicenseTypeStore.javaPERFORCE_ACADEMIC*L
clover/com/atlassian/license/LicenseType;PERFORCE_EVALUATIONPERFORCE_DEMONSTRATIONPERFO
RCE_NON_PROFITPERFORCE_COMMUNITYPERFORCE_DEVELOPERPERFORCE_OPEN_SOURCEPERF
ORCE_FULL_LICENSEpublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z

 "

 \$ & (*,

.thisYLclover/com/atlassian/license/applications/perforceplugin/PerforcePluginLicenseTypeStore;getAllLicenses()L
java/util/Collection;getPublicKeyFileName()Ljava/lang/String; 6getPrivateKeyFileName

9<clinit>/clover/com/atlassian/license/DefaultLicenseType<Perforce:

Academic>+clover/com/atlassian/extras/api/LicenseType@ACADEMIC-

Lclover/com/atlassian/extras/api/LicenseType;BC ADnameF5

AG*(ILjava/lang/String;ZZLjava/lang/String;)VI

=JPerforce: EvaluationL

COMMERCIALNC AOPerforce: DemonstrationQ

DEMONSTRATIONSC AT"Perforce: Non-Profit / Open SourceV

NON_PROFITXC AYPerforce: Community[COMMUNITY]C A^Perforce: Developer` DEVELOPERbC AcPerforce:

Open SourceeOPEN_SOURCEgC AhPerforce:

CommercialjCclover/com/atlassian/license/applications/jira/JiraLicenseTypeStorel m6

m9CodeLocalVariableTableLineNumberTable

SourceFile!

p]**!W*#!W*%!W*'!W*)!W*+!W*-!W*/!Wq]01r*

!"%#0$;%F&Q'\(23p/*q01r,45p.7q01r185p.:q01r6;p=Y3?EHK=Y7MPHK#=Y8RUHK+=Y9WZHK%=Y:_HK)=Y
;adHK-=Y<fiHK/=Y=kPHK'n7o:r*

.E\ss

0lGclover/com/atlassian/license/applications/clover/CloverLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreCloverLicenseTypeStore.javaCLOVER_ACADEMIC*Lclover/com/
atlassian/license/LicenseType;CLOVER_COMMERCIALCLOVER_COMMUNITYCLOVER_EVALUATIONCL
OVER_OPEN_SOURCECLOVER_DEVELOPERCLOVER_DEMONSTRATIONpublicKeyFileNameLjava/lang/
String;privateKeyFileName<init>()V

applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z

 ! #

 % ')

+thisILclover/com/atlassian/license/applications/clover/CloverLicenseTypeStore;getAllLicenses()Ljava/util/Collecti
on;getPublicKeyFileName()Ljava/lang/String; 3getPrivateKeyFileName

6<clinit>/clover/com/atlassian/license/DefaultLicenseType9Clover:
Academic;+clover/com/atlassian/extras/api/LicenseType=ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;?@ >AnameC2

>D*(ILjava/lang/String;ZZLjava/lang/String;)VF

:GClover: CommercialI

COMMERCIALK@ >LClover: CommunityN COMMUNITYP@ >QClover: EvaluationSClover: Open
SourceUOPEN_SOURCEW@ >XClover: DeveloperZ DEVELOPER\@ >]Clover: Demonstration_
DEMONSTRATIONa@

>b$clover/com/atlassian/clover/leaf.keydclover/clover.bytefCodeLocalVariableTableLineNumberTable
SourceFile!

 hR** W*" W*$ W*& W*(W** W*, WiR-.j& %0; F!Q"/0h/*i-.j&12h.4i-.j+52h.7i-
.j08h:Y<BEH:YJMEH":YOREH$:YTMEH&:YVYEH(:Y[^EH*:Y$`cEH,e4g7j&

.E\sk

0Oclover/com/atlassian/license/applications/confluence/ConfluenceLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreConfluenceLicenseTypeStore.javaACADEMIC*Lclover/com/atlassia
n/license/LicenseType;

EVALUATIONTESTINGHOSTED_EVALUATION
NON_PROFITFULL_LICENSEPERSONALSTARTERHOSTED COMMUNITYOPEN_SOURCE DEVELOPER
DEMONSTRATIONpublicKeyFileNameLjava/lang/String;-
clover/com/atlassian/confluence/page/Page.keyprivateKeyFileNameconfluence/confluence.byteAPPLICATION_N
AMECONF<init>()V

applicationLicenseTypesLjava/util/ArrayList;"# $ &java/util/ArrayList(add(Ljava/lang/Object;)Z*+

), . 0

 2 4 6

 8 : < > @ B

DthisQLclover/com/atlassian/license/applications/confluence/ConfluenceLicenseTypeStore;getAllLicenses()Ljava/u
til/Collection;getPublicKeyFileName()Ljava/lang/String;getPrivateKeyFileName<clinit>/clover/com/atlassian/licen
se/DefaultLicenseTypeNConfluence: AcademicP+clover/com/atlassian/extras/api/LicenseTypeR-
Lclover/com/atlassian/extras/api/LicenseType;T SUnameWK

SX*(ILjava/lang/String;ZZLjava/lang/String;)VZ

O[Confluence: Evaluation]

COMMERCIAL_T S`Confluence: Testingb T Sd+(ILjava/lang/String;ZZZLjava/lang/String;)Vf

OgConfluence: Hosted EvaluationiT Sk$Confluence: Non-Profit / Open SourcemT SoConfluence: Commercial

ServerqConfluence: Personal Servers
T SuConfluence: StarterwT SyConfluence: Commercial Hosted{Confluence: Community}T SConfluence: Open
SourceT SConfluence: DeveloperT SConfluence: DemonstrationT S
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!

   !%'-W*%/-W*%1-W*%3-W*%5-W*%7-W*%9-W*%;-W*%=-W*%?-W*%A-W*%C-W*%E-WFG>
!"#%$0%;&F'Q(\)g*r+},-.HI/*%FG2JK-FG7LK-FG<Mo'OYQVY\'OY
^aY\/OY0ceYh1OY@jlY\3OYNnpY\5OYUraY\7OY`tvY\9OY0xzY\COY|lYh;OY~Y\=OY
Y\?OYY\AOY+Y\E6

,
CYo
0Iclover/com/atlassian/license/applications/fisheye/FishEyeLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreFishEyeLicenseTypeStore.javaFISHEYE_ACADEMIC*Lclover/co
m/atlassian/license/LicenseType;FISHEYE_COMMERCIALFISHEYE_COMMUNITYFISHEYE_EVALUATION
FISHEYE_OPEN_SOURCEFISHEYE_DEVELOPERFISHEYE_STARTERFISHEYE_DEMONSTRATIONFISH
EYE_TESTINGpublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
! # %
') +
- /
1thisKLclover/com/atlassian/license/applications/fisheye/FishEyeLicenseTypeStore;getAllLicenses()Ljava/util/Coll
ection;getPublicKeyFileName()Ljava/lang/String; 9getPrivateKeyFileName
<<clinit>/clover/com/atlassian/license/DefaultLicenseType?FishEye:
AcademicA+clover/com/atlassian/extras/api/LicenseTypeCACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;EF DGnameI8
DJ*(ILjava/lang/String;ZZLjava/lang/String;)VL
@MFishEye: CommercialO
COMMERCIALQF DRFishEye: CommunityT COMMUNITYVF DWFishEye: EvaluationYFishEye: Open
Source[OPEN_SOURCE]F D^FishEye: Developer` DEVELOPERbF DcFishEye: StartereSTARTERgF DhFishEye:
Demonstrationj
DEMONSTRATIONlF DmFishEye: TestingoTESTINGqF Dr+(ILjava/lang/String;ZZZLjava/lang/String;)Vt
@u%clover/com/atlassian/fisheye/leaf.keywfisheye/fisheye.byteyCodeLocalVariableTableLineNumberTable
SourceFile!

   {h**"W*$"W*&"W*("W**"W*,"W*."W*0"W*2"W|h34}.%0
;!F"Q#\$g%56{/*|34})78{.:|34}.;8{.=|34}3>{@YBHKN@YPSKN$@YUXKN&@YZSKN(@Y_KN*@YadKN,
@YfiKN.@YknKN0@YpsKv2x:z=}.
.E\s~
0QWclover/com/atlassian/license/applications/editliveplugin/EditlivePluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStore#EditlivePluginLicenseTypeStore.javaEDITLIVE_ACADEMIC*Lcl
over/com/atlassian/license/LicenseType;EDITLIVE_EVALUATIONEDITLIVE_NON_PROFITEDITLIVE_FULL
_LICENSEpublicKeyFileNameLjava/lang/String;,clover/com/atlassian/editlive/publickey.byte
privateKeyFileNameeditlive/editlive.byte<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
" $
&thisYLclover/com/atlassian/license/applications/editliveplugin/EditlivePluginLicenseTypeStore;getAllLicenses()L
java/util/Collection;getPublicKeyFileName()Ljava/lang/String;getPrivateKeyFileName<clinit>/clover/com/atlassian

/license/DefaultLicenseType0EditLive!: Academic2+clover/com/atlassian/extras/api/LicenseType4ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;67 58name:-
5;*(ILjava/lang/String;ZZLjava/lang/String;)V=
1>EditLive!: Evaluation@
COMMERCIALB7 5C#EditLive!: Non-Profit / Open SourceE
NON_PROFITG7 5HEditLive!: CommercialJ
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!
LLMo1**!W*#!W*%!W*'!WN1()O%0*+M/*N()O,-M-N()O$.-M-
N()O)/M]1Y39<?1YAD<?#1YFI<?%1YKD<?'O
.EP
0Gclover/com/atlassian/license/applications/bamboo/BambooLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreBambooLicenseTypeStore.javaBAMBOO_BASIC_EVALUATION*
Lclover/com/atlassian/license/LicenseType;BAMBOO_BASIC_ACADEMICBAMBOO_BASIC_DEMONSTRATI
ONBAMBOO_BASIC_DEVELOPERBAMBOO_BASIC_COMMUNITYBAMBOO_BASIC_OPEN_SOURCEB
AMBOO_BASIC_COMMERCIAL_SERVERBAMBOO_EVALUATIONBAMBOO_ACADEMICBAMBOO_DE
MONSTRATIONBAMBOO_DEVELOPERBAMBOO_COMMUNITYBAMBOO_OPEN_SOURCEBAMBOO_C
OMMERCIAL_SERVERBAMBOO_PROFESSIONAL_EVALUATIONBAMBOO_PROFESSIONAL_ACADEM
IC!BAMBOO_PROFESSIONAL_DEMONSTRATIONBAMBOO_PROFESSIONAL_DEVELOPERBAMBOO_P
ROFESSIONAL_COMMUNITYBAMBOO_PROFESSIONAL_OPEN_SOURCE%BAMBOO_PROFESSIONAL
_COMMERCIAL_SERVERBAMBOO_ENTERPRISE_EVALUATIONBAMBOO_ENTERPRISE_ACADEMICB
AMBOO_ENTERPRISE_DEMONSTRATIONBAMBOO_ENTERPRISE_DEVELOPERBAMBOO_ENTERPRIS
E_COMMUNITYBAMBOO_ENTERPRISE_OPEN_SOURCE#BAMBOO_ENTERPRISE_COMMERCIAL_SER
VERBAMBOO_TEMP_2_0_BETApublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V'(
)applicationLicenseTypesLjava/util/ArrayList;+, - /java/util/ArrayList1add(Ljava/lang/Object;)Z34
25 7 9 ; = ? A C E G
 I K M
 O Q S U W Y [ ] _ a c e g! i" k#
mthisILclover/com/atlassian/license/applications/bamboo/BambooLicenseTypeStore;getAllLicenses()Ljava/util/Col
lection;getPublicKeyFileName()Ljava/lang/String;$% ugetPrivateKeyFileName&%
x<clinit>/clover/com/atlassian/license/DefaultLicenseType{Bamboo Basic:
Evaluation}+clover/com/atlassian/extras/api/LicenseType
COMMERCIAL-Lclover/com/atlassian/extras/api/LicenseType; namet
.clover/com/atlassian/extras/api/LicenseEditionBASIC0Lclover/com/atlassian/extras/api/LicenseEdition;
Z(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V'
|Bamboo Basic: AcademicACADEMIC Bamboo Basic: Demonstration
DEMONSTRATION Bamboo Basic: Developer DEVELOPER Bamboo Basic: Community COMMUNITY Bamboo
Basic: Open SourceOPEN_SOURCE Bamboo Basic: Commercial ServerBamboo Standard: EvaluationSTANDARD
Bamboo Standard: AcademicBamboo Standard: DemonstrationBamboo Standard: DeveloperBamboo Standard:
CommunityBamboo Standard: Open Source"Bamboo Standard: Commercial ServerBamboo Professional:
EvaluationPROFESSIONAL Bamboo Professional: Academic"Bamboo Professional: DemonstrationBamboo
Professional: DeveloperBamboo Professional: Community Bamboo Professional: Open Source&Bamboo
Professional: Commercial ServerBamboo Enterprise: Evaluation
ENTERPRISE Bamboo Enterprise: Academic Bamboo Enterprise: DemonstrationBamboo Enterprise:
DeveloperBamboo Enterprise: CommunityBamboo Enterprise: Open Source$Bamboo Enterprise: Commercial
ServerBamboo: 2.0 betaTESTING
$clover/com/atlassian/bamboo/leaf.keybamboo/bamboo.byteCodeLocalVariableTableLineNumberTable
SourceFile!

! " # $%

&%'(D***.06W*.86W*.:6W*.<6W*.>6W*.@6W*.B6W*.D6W*.F6W*.H6W*.J6W*.L6W*.N6W*.P6W*.R6W*. T6W*.V6W*.X6W*.Z6W*.\6W*.^6W*.`6W*.b6W*.d6W*.f6W*.h6W*.j6W*.l6W*.n6WDop~79:;%<0=;>F?QA\ BgCrD}EFGIJKLMNOQRSTU"V- W8YCZqr/*.op^st.vopcwt.yophz(|Y~F|YD|YH|YJ|YL|YN|YP|Y8|Y0|Y:|Y<|Y>|Y@|YB|YT|YR|YV|YX|YZ|Y\|Y^| Yb|Y`|Yd|Yf|Yh|Yj|Yl|Ynvy~

4Nh8Rl !"#$%"(<)V*p+,-.134

0+clover/com/atlassian/license/LicenseManagerjava/lang/ObjectLicenseManager.java1clover/com/atlassian/extras/c ommon/log/Logger$Log- clover/com/atlassian/extras/common/log/LoggerLoglog3Lclover/com/atlassian/extras/common/log/Logger$Log;lice nseListLjava/util/Map;licenseConfigurationslicenseManager- Lclover/com/atlassian/license/LicenseManager;1class$clover$com$atlassian$license$LicenseManagerLjava/lang/Cl ass;<init>()V

java/util/HashMap

 thisgetInstance/()Lclover/com/atlassian/license/LicenseManager; " addLicenseConfigurationr(Ljava/lang/String;Lclover/com/atlassian/license/LicenseTypeStore;Lclover/com/atlassian /license/LicenseRegistry;)V1clover/com/atlassian/license/LicenseConfiguration`(Lclover/com/atlassian/license/Lice nseRegistry;Lclover/com/atlassian/license/LicenseTypeStore;)V)

(*

java/util/Map,put8(Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object;./- 0applicationNameLjava/lang/String;licenseTypeStore/Lclover/com/atlassian/license/LicenseTypeStore;licenseRegis try.Lclover/com/atlassian/license/LicenseRegistry;licenseConfiguration3Lclover/com/atlassian/license/LicenseConfi guration;getLicenseRegistryB(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseRegistry;getLicenseConfigu rationG(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseConfiguration;<= >0()Lclover/com/atlassian/license/LicenseRegistry;:@ (AgetLicenseTypeStoreC(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseTypeStore;1()Lclover/com/atlass ian/license/LicenseTypeStore;CE (Fget&(Ljava/lang/Object;)Ljava/lang/Object;HI-Jjava/lang/RuntimeExceptionLjava/lang/StringBufferN O&No LicenseConfiguration found for key Qappend,(Ljava/lang/String;)Ljava/lang/StringBuffer;ST OUtoString()Ljava/lang/String;WX OY(Ljava/lang/String;)V[M\lookupLicenseTypeStorehasValidLicense(Ljava/lang/String;)Z getLicense:(Ljava/lang/String;)Lclover/com/atlassian/license/License;ab c$clover/com/atlassian/license/Licensee isExpired()Zghfi licenseKey-clover/com/atlassian/license/LicenseExceptionljava/lang/ExceptionnisEmptyph- qcontainsKey(Ljava/lang/Object;)Zst-u w>There is no License Configuration defined for the application y.{error(Ljava/lang/Object;)V}~,clover/com/atlassian/license/LicenseRegistrygetLicenseMessageXgetLicenseHash X?There is no license string or hash defined for the application info~(clover/com/atlassian/license/LicensePair'(Ljava/lang/String;Ljava/lang/String;)V

Could not build a license pair*(Ljava/lang/Object;Ljava/lang/Throwable;)V}3clover/com/atlassian/license/decoder/LicenseDecoderd(Lclover /com/atlassian/license/LicensePair;Ljava/lang/String;)Lclover/com/atlassian/license/License;a

Exception getting license: ,(Ljava/lang/Object;)Ljava/lang/StringBuffer;S Oe/Lclover/com/atlassian/license/LicenseException;license&Lclover/com/atlassian/license/License; licenseStrhashpair*Lclover/com/atlassian/license/LicensePair;Ljava/lang/Exception; setLicenseL(Ljava/lang/String;Ljava/lang/String;)Lclover/com/atlassian/license/License;

\isValid?(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)Z

?(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)V

qAttempt to set invalid license. Ensure that you are calling setLicense(license, appName) - not (appName, license)warnupdatedLicenseremoveI-()[Ba

)clover/com/atlassian/license/LicenseUtils getString([B)Ljava/lang/String;

setLicenseMessage[getHash

setLicenseHash[getLicensePair>(Ljava/lang/String;)Lclover/com/atlassian/license/LicensePair; Couldn't get the LicensePair ...getLicenseTypeP(Ljava/lang/String;Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;-clover/com/atlassian/license/LicenseTypeStore>(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;

licenseTypeString?(Ljava/lang/String;I)Lclover/com/atlassian/license/LicenseType;-

(I)Lclover/com/atlassian/license/LicenseType;

licenseTypeCodeIresetclear-clearLicenseConfigurations

removeLicense<clinit> getClass()Ljava/lang/Class;

java/lang/ClassgetComponentType

F(Ljava/lang/Class;)Lclover/com/atlassian/extras/common/log/Logger$Log;

  SyntheticCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses!

Q**Y*Y !"# !4#

Y$##+-0%&s(Y-,+:*+1W423456789567:;= *+?B 23;CD= *+?G 23@<=y/*+K(M,MYOYPRV+VZ], //23!89EFH-J^Dc*+K(M,,G 2389TUW[_`O*+d*+djk3bab

r+vM*+K(N-%xOYPzV+V|VZ-B:::%xOYPV+V|VZ:Y::x+M*+,1W MxOYPV,Z,*+KfmNoOooof

)89Ut67^k3gb3323bgkl)m-oMpOrUs^tgvqxy|3NY+N-,:-, *-,:x"#o>%3333231& #%1@+>*,W*,K(N-B:++4"89#67@@@23#1?m5*+K(M,BNY--Mx,&'m48967(

5523'(3c*+K(N-G,*233 89

mc*+K(N-G*23 89

mM**# 8

*

 [D*+W23

5YWYx

0X.clover/com/atlassian/license/DefaultSIDManagerjava/lang/Object'clover/com/atlassian/license/SIDManagerDefaultSIDManager.javaCHARACTER_POOLLjava/lang/String;$ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

 BAD_WORDS[Ljava/lang/String;

KEY_LENGTHICURRENT_VERSION_INITAL_CHARCBPREVIOUS_VERSIONS_INITIAL_CHARSASEPARATOR_CHAR-randomLjava/security/SecureRandom;<init>()V([B)V

this0Lclover/com/atlassian/license/DefaultSIDManager;

"java/lang/System$currentTimeMillis()J&'

%(java/lang/String*valueOf(J)Ljava/lang/String;,-

+.java/lang/StringBuffer0

1"append,(Ljava/lang/String;)Ljava/lang/StringBuffer;34

15:7identityHashCode(Ljava/lang/Object;)I9:

%;(I)Ljava/lang/StringBuffer;3=

1>toString()Ljava/lang/String;@A

1B

getProperties()Ljava/util/Properties;DE

%Fjava/util/PropertiesH

IBgetBytes()[BKL

+Mjava/security/SecureRandomO

P RseedStrseed[BgenerateSID(C)Ljava/lang/StringBuffer;3X

1Y

nextDouble()D[\

P]length()I_`

+acharAt(I)Ccd

+egetCharacterForCRC([B)Cgh

i

isKeyClean(Ljava/lang/String;)Zkl

mindexi charCountresLjava/lang/StringBuffer;

isValidSIDvalidateStringSyntaxul

v substring(II)Ljava/lang/String;xy

+zkeyStrcrcChar checkCharsidvalidZindexOf(I)I

+java/util/zip/CRC32

"update

getValue'

bytescrc32Ljava/util/zip/CRC32;crcValueJstripDashesAndNumbers&(Ljava/lang/String;)Ljava/lang/String;

 (Ljava/lang/String;)I

+badWordkeycharKey--clover/com/atlassian/license/util/StringUtils

replaceAllJ(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;

123E45S6G7T8B9P0O<clinit>FUCKSHITCOCKDICKCUNTTWATBITCHBASTARDJIZJISMFARTCRAPASS

PORNPISSPUSSYBALLSTITSBOOBSCOOCHCUMCHOADDILDODOUCHECLITMUFFNOBPECKERPRIC

KPOONTANG QUEEFSNATCH

TWOTDYKECOONNIGFAGWANKERGOOKFUDGEPACKERQUEERRAGHEAD!SKANK#SPIC%GOD'DA

MN)FICK+SCHEISSE-SCHWANZ/FOTZE1HURE3

SCHWUCHTEL5SCHWUL7TITTEN9ARSCH;IDIOT=SAU?ASSHATATURDBURGLARCDIRTYSANCHEZE

FELCHGBLASENIWICKSERKFEUCHTMMOESEOMILCHTUETENQ

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile! S

SS SS T4*U !V

TY*#+G)/M1Y2,686,<?CM1Y2,686GJ6CM,NL*PY+QSU =T Y !YUVV"&')**+G-L0X1WATp1Y2L+BZW=>:+-

ZW=#*S^bk6+fZW+*+CNjZW*+Cn+CU4;o=pOqp !hrsV:8;<=@B%C*H;IGJJ=PO`QkStlT3*+w=*+{N*-

Nj6+f6=U>| }$

~3 !3 -VYZ

]_a$c1fulTZ=+=P+b=B+fB+f=%+f-+ f-+f-=U Z !Z XV*

pqsuwy1{6}VXghT%YM,+,B!bq6fU4% !%VoV

klT+*+M>2:,U4 !p+ !+ % V!#)Te+M,M,M,M,M,M,M,M,M,M,M,U e !e \r V2 $-

6?HQZcTB+YSYSYSYSYSYSYSYSYSY SY

SYSYSY

SYSYSYSYSYSYSYSYSYSYSYSYSYSYSYSY

SYSYSY SY!SY"SY#SY$SY%SY&SY'SY(SY)"SY*$SY+&SY,(SY-

*SY.,SY/.SY00SY12SY24SY36SY48SY5:SY6<SY7>SY8@SY9BSY:DSY;FSY<HSY=JSY>LSY?NSY@PSYA

RSVW

0-
clover/com/atlassian/license/LicenseExceptionjava/lang/ExceptionLicenseException.java<init>(Ljava/lang/String;)
V

this/Lclover/com/atlassian/license/LicenseException;messageLjava/lang/String;(Ljava/lang/Throwable;)V
causeLjava/lang/Throwable;*(Ljava/lang/String;Ljava/lang/Throwable;)V
CodeLocalVariableTableLineNumberTable
SourceFile!>*+

>*+

I*+,
0$clover/com/atlassian/license/Licensejava/lang/ObjectLicense.javagetDateCreated()Ljava/util/Date;getDatePurchas
edgetOrganisation()Ljava/lang/String;getLicenseType,()Lclover/com/atlassian/license/LicenseType; isExpired()Z
getExpiryDatetoStringisLicenseLevel(Ljava/util/Collection;)ZgetUsers()IgetPartnerNamegetLicenseIdgetPermitted
ClusteredNodesgetLicenseDuration()JgetSupportEntitlementNumber
SourceFile
0Z4clover/com/atlassian/license/AbstractLicenseRegistryjava/lang/Object,clover/com/atlassian/license/LicenseRegi
stryAbstractLicenseRegistry.java<init>()V

this6Lclover/com/atlassian/license/AbstractLicenseRegistry;getAllLicenseTypes()Ljava/util/Collection;getLicenseT
ype>(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;-clover/com/atlassian/license/LicenseException
java/util/Collectioniterator()Ljava/util/Iterator;java/util/IteratorhasNext()Z
next()Ljava/lang/Object;"#$(clover/com/atlassian/license/LicenseType&getDescription()Ljava/lang/String;()'*java/l
ang/String,toLowerCase.)
-/indexOf(Ljava/lang/String;)I12
-3java/lang/StringBuffer5
6
The license type (8append,(Ljava/lang/String;)Ljava/lang/StringBuffer;:;
6<) specified is invalid.>toString@)
6A(Ljava/lang/String;)VC
DlicenseType*Lclover/com/atlassian/license/LicenseType;licenseTypeDescLjava/lang/String;Ljava/util/Iterator;typ
e-(I)Lclover/com/atlassian/license/LicenseType;getType()IMN'O(I)Ljava/lang/StringBuffer;:Q
6RICodeLocalVariableTableLineNumberTable
Exceptions
SourceFile! U/*V
W
UZ*M,!*,%'N-+0:+04-Y6Y79=+=?=BEV4FG(HI
0JZ
ZKIW(57:XLUL*M,!,%'N-P-Y6Y79=S?=BEV*FG
"JL
LKTW#%"()),*XY
0L/clover/com/atlassian/license/DefaultLicenseTypejava/lang/Object(clover/com/atlassian/license/LicenseTypeDefa
ultLicenseType.javatypeIdescriptionLjava/lang/String;isEvaluationZrequiresUserLimitexpiresnewLicenseTypeNam
eedition0Lclover/com/atlassian/extras/api/LicenseEdition;<init>*(ILjava/lang/String;ZZLjava/lang/String;)V[(ILjav
a/lang/String;ZZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V
this1Lclover/com/atlassian/license/DefaultLicenseType;Z(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atla
ssian/extras/api/LicenseEdition;)V+(ILjava/lang/String;ZZZLjava/lang/String;)V()V

!
#
%
') +
getEdition2()Lclover/com/atlassian/extras/api/LicenseEdition;getNewLicenseTypeName()Ljava/lang/String;hashCode()Ijava/lang/String312
45getTypetoStringgetDescription90
:isEvaluationLicenseType()ZgetNiceNameequals(Ljava/lang/Object;)Z72
AAlicense*Lclover/com/atlassian/license/LicenseType;oLjava/lang/Object;CodeLocalVariableTableLineNumberTable
SourceFile!

Hn*,I>

J

Hy*,IH

J
Hy*,IH

J
!H ,** *,"*$*&*(***,IR,, ,
,
,
,
,,J& $% &'()*%++,-.H/*,IJ0/0H/**IJ512H7
* *"6`I
J:72H/* IJ?80H/*;IJD90H/*"IJI<=H/*$IJN=H/*(IJS=H/*&IJX>0H/*"IJ`?@Hk++M*B,CI DEFGJeghjmK
0)clover/com/atlassian/license/LicenseUtilsjava/lang/ObjectLicenseUtils.java1clover/com/atlassian/extras/common/log/Logger$Log-
clover/com/atlassian/extras/common/log/LoggerLoglog3Lclover/com/atlassian/extras/common/log/Logger$Log;POST_LICENSE_EVAL_PERIODJ~UPDATE_ALLOWED_PERIOD\ALMOST_EXPIRED_PERIODKPARTNER_NOT_MATCHING_BUILDLjava/lang/String;'partner not matching build partner nameLICENSE_NO_PARTNER'License does not contain a partner name/class$clover$com$atlassian$license$LicenseUtilsLjava/lang/Class;<init>()V !
"this+Lclover/com/atlassian/license/LicenseUtils; getString([B)Ljava/lang/String;rndChar(I)C()
*getCharInRange,)
- /java/lang/StringBuffer1
2"Invalid Char in stream 4append,(Ljava/lang/String;)Ljava/lang/StringBuffer;67
28(I)Ljava/lang/StringBuffer;6:
2;toString()Ljava/lang/String;=>

2?debug(Ljava/lang/Object;)VABCjava/lang/StringE([C)V G

FHaByteBiI byteArray[BcharByte[Cstrjava/lang/MathSrandom()DUV

TW@@cuZInvalid int in stream `c1getByteInRange(C)BIncorrect character in stream

e(C)Ljava/lang/StringBuffer;6g

2hCgetBytes(Ljava/lang/String;)[BtoCharArray()[Cmn

Fojava/lang/CharacterqtoLowerCase(C)Cst

rucd

w!Invalid character in byte stream ystring

charArraybytesgetSupportPeriodEnd)(Lclover/com/atlassian/license/License;)J$clover/com/atlassian/license/Licens

egetDateCreated()Ljava/util/Date;java/util/DategetTime()J

license&Lclover/com/atlassian/license/License;isLicenseTooOldForBuild9(Lclover/com/atlassian/license/License;L

java/util/Date;)Z~

 buildDateLjava/util/Date;confirmExtendLicenseExpired(Ljava/util/Date;)Z

"#getNewBuildWithOldLicenseExpiryDate(Ljava/util/Date;)J

dateConfirmed(Ljava/lang/String;)Zjava/lang/NumberFormatExceptionjava/lang/Long

parseLong(Ljava/lang/String;)J

(J)V

!getSupportPeriodAlmostExpiredDateisPartnerDetailsValidL(Lclover/com/atlassian/license/License;Ljava/lang/Stri

ng;)Ljava/lang/String;getPartnerName>equals(Ljava/lang/Object;)Z

FbuildPartnerNamelicensePartnerNamereadKey(Ljava/io/InputStream;)[Bjava/io/IOExceptionjava/io/ByteArrayOut

putStream

"java/io/InputStreamread([B)I

write([BII)V

toByteArray()[B

isLjava/io/InputStream;boutLjava/io/ByteArrayOutputStream;len<clinit> getClass()Ljava/lang/Class;

java/lang/ClassgetComponentType

getInstanceF(Ljava/lang/Class;)Lclover/com/atlassian/extras/common/log/Logger$Log;

ConstantValue SyntheticCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses!

!3*#$%

 &'a*hL=**3>!++U+*``.U++U+*`@`.Ua@++U+*`.UB@!++U+*`@d.U02Y359<@DPFY+IM,4JK

LMNOPQRN#$&')&*7,A.H/X1c3j4w689=$@A

()u/hXYk`<X[k=aA` /LM!]M ^_FG H

,)j 0`

#

dA`$=$da`><?>02Y3a9<@DjbM2PRTV&X2Z;\A^D`JbMehf

cdk09 0dAZAd

`azad$`<>>?02Y3f9i@Dkbj2nprt'v3x<zB|E~KNi

kl**pL+lM>,+4vg,+,`4xdTv+4vm,+,`4x@dTX+4vs,+,`4xT>+4vy,+,`4x@`T 02Y3z9+4i@De,*LM{|Q}O:3>Q\kv

~8*a F*+ BY* 9Y*

3 *a 9Y* 3 *e ?*M,,,+,,++ ??89< *YL=N*-Y=

+-+**"

M}O

%!5YWY0

0?-clover/com/atlassian/license/util/StringUtilsjava/lang/ObjectStringUtils.java<init>()V
this/Lclover/com/atlassian/license/util/StringUtils;
replaceAllJ(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;java/lang/Stringequals(Ljava/lang/Object;)Z
java/lang/StringBufferlength()I
(I)V
indexOf(Ljava/lang/String;)I
! substring(II)Ljava/lang/String;#$
%append,(Ljava/lang/String;)Ljava/lang/StringBuffer;'(
)(I)Ljava/lang/String;#+
,toString()Ljava/lang/String;./
0iIstrLjava/lang/String;
oldPattern
newPattern remainderbufLjava/lang/StringBuffer;CodeLocalVariableTableLineNumberTable
SourceFile!;/* <
=
;i*++**NY*h:-+"6&-&*W,*W-+`-N
-*W1<>-323i45i65i75R85&C9:=6
&-3@GV]`c!>
0(clover/com/atlassian/license/LicenseTypejava/lang/ObjectLicenseType.javaequals(Ljava/lang/Object;)ZhashCode
()IgetTypetoString()Ljava/lang/String;getDescriptiongetNiceNameisEvaluationLicenseType()ZrequiresUserLimitex
piresgetNewLicenseTypeName
getEdition2()Lclover/com/atlassian/extras/api/LicenseEdition;
SourceFile

0m-
clover/com/atlassian/license/LicenseTypeStorejava/lang/ObjectLicenseTypeStore.javaapplicationLicenseTypesLjav
a/util/ArrayList;<init>()V

java/util/ArrayList

this/Lclover/com/atlassian/license/LicenseTypeStore;getPublicKeyFileName()Ljava/lang/String;getPrivateKeyFileN
amegetLicenseType>(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;-
clover/com/atlassian/license/LicenseExceptionjava/lang/Stringequals(Ljava/lang/Object;)Z
java/lang/StringBuffer"
#
1License description must be specified; you used [%append,(Ljava/lang/String;)Ljava/lang/StringBuffer;'(
#)]+toString-
#.(Ljava/lang/String;)V0
1iterator()Ljava/util/Iterator;34

5java/util/Iterator7hasNext()Z9:8;next()Ljava/lang/Object;=>8?(clover/com/atlassian/license/LicenseTypeAgetDesc
riptionCBD:License type added with an invalid description; you used [FtoLowerCaseH
IindexOf(Ljava/lang/String;)IKL
M(LicenseType not found with description

[OlicenseType*Lclover/com/atlassian/license/LicenseType;licenseTypeDescLjava/lang/String;Ljava/util/Iterator;licenseTypeString-(I)Lclover/com/atlassian/license/LicenseType;getType()IXYBZThe license type
(\(I)Ljava/lang/StringBuffer;'^

#_) specified is

invalid.alicenseCodeIlookupLicenseTypegetAllLicenses()Ljava/util/Collection;CodeLocalVariableTableLineNumberTable

Exceptions

SourceFile! h>**

Yij

 h;++!#Y#Y$&*+*,*/2*6M,<f,@BN-E-E!(Y#Y$G*-E*,*/2-EJ:+JN-Y#Y$P*+*,*/2i4HVQRST5l3UVTj.
->H_!$&(*kWhJ*6M,<,@BN-[-Y#Y$]*`b*/2i*QR"3UJJcdj/13%4'5*6keWh,*6M,<,@BN-[-i*QR"3U,,cdj@BD%E'F*Gfgh/*ijNl

0

,clover/com/atlassian/license/LicenseRegistryjava/lang/ObjectLicenseRegistry.javasetLicenseMessage(Ljava/lang/String;)VsetLicenseHashgetLicenseMessage()Ljava/lang/String;getLicenseHash

SourceFile

0!1clover/com/atlassian/license/LicenseConfigurationjava/lang/ObjectLicenseConfiguration.javalicenseRegistry.Lclover/com/atlassian/license/LicenseRegistry;licenseTypeStore/Lclover/com/atlassian/license/LicenseTypeStore;<init>`(Lclover/com/atlassian/license/LicenseRegistry;Lclover/com/atlassian/license/LicenseTypeStore;)V()V

this3Lclover/com/atlassian/license/LicenseConfiguration;getLicenseRegistry0()Lclover/com/atlassian/license/LicenseRegistry;getLicenseTypeStore1()Lclover/com/atlassian/license/LicenseTypeStore;setLicenseRegistry1(Lclover/com/atlassian/license/LicenseRegistry;)VsetLicenseTypeStore2(Lclover/com/atlassian/license/LicenseTypeStore;)VCodeLocalVariableTableLineNumberTable

SourceFile!

Y**+*, /*/*>*+

!>*+

%&

0+clover/com/atlassian/license/DefaultLicensejava/lang/Object$clover/com/atlassian/license/LicenseDefaultLicense.javaEVALUATION_PERIODJdateCreatedLjava/util/Date;

datePurchaseddateExpiredorganisationLjava/lang/String;licenseType*Lclover/com/atlassian/license/LicenseType;usersIpartnerName

licenseIdpermittedClusteredNodesdurationsen<init>(Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;)V(Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;ILjava/lang/String;)V

this-Lclover/com/atlassian/license/DefaultLicense;(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;)V(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;ILjava/lang/String;)V#

$expiresorganisationName(Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;I)V(

)

+(Ljava/util/Date;Ljava/util/Date;JLjava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;ILjava/lang/String;)V .

0r(Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;)V(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILja

va/lang/String;)V(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;I)V4

5()V7

8

 < > @ B D F H

J(Ljava/util/Date;Ljava/util/Date;JLjava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String; Ljava/lang/String;I)VgetDateCreated()Ljava/util/Date;getDatePurchasedgetOrganisation()Ljava/lang/String;getLicenseType,()Lclover/com/atlassian/license/LicenseType;toStringjava/lang/StringBufferU

V8(clover/com/atlassian/license/LicenseTypeXgetNiceNameZQY[append,(Ljava/lang/String;)Ljava/lang/StringBuffer;]^

V_

licensed to aTQ

Vc isExpired()Z

getExpiryDategN

hjava/util/DatejgetTime()Jlm

knjava/lang/SystempcurrentTimeMillisrm

qsexpiryisEvaluationLicenseTypevfYw y(J)V{

k|getPartnerNameisLicenseLevel(Ljava/util/Collection;)Zjava/util/Collectioniterator()Ljava/util/Iterator;java/util/Iteratorhas Nextfnext()Ljava/lang/Object;java/lang/StringRS

getDescriptionQYtoLowerCaseQ

indexOf(Ljava/lang/String;)I

levelLjava/util/Iterator;levelsLjava/util/Collection;getUsers()IrequiresUserLimitfYgetLicenseIdgetPermittedClustere dNodesgetLicenseDurationgetSupportEntitlementNumber<clinit>$CodeLocalVariableTableLineNumberTable SourceFile!

 *+,-R !

"
*+,-%\ !
&'
$% *+,-** ,f
!
)*+-

*+,
*!/p !

/01#
*+,
*-1p !

5672{ *+,-*H !

?@3
 *+,-6R !
&'

DE(>*9*:/*,*+=*,?*-A*C*E*G*I*K\ > !>
>>>>>>>2HIJKL%M+N1O7P=QL *+,
**!/f
!

UVW4
*+, **-1f
!

 [\]MN/*= !aON/*? !fPQ/*A !kRS/*C !pTQM#VYW*C\`b`*A`d# !uef*iL++ot !uz{ |~gNn**1L+"*CxkY*?oza}L+*
!%u (~Q/*G !8+M,),N*-*/8 !8136E*C*E !Q/*I !/*K !m/*/ !Q/*, !7z
0'clover/com/atlassian/license/SIDManagerjava/lang/ObjectSIDManager.javagenerateSID()Ljava/lang/String;
isValidSID(Ljava/lang/String;)Z
SourceFile
0)8clover/com/atlassian/extras/core/jira/DefaultJiraLicense6clover/com/atlassian/extras/core/DefaultProductLicense
0clover/com/atlassian/extras/api/jira/JiraLicenseDefaultJiraLicense.javalicenseEdition0Lclover/com/atlassian/extras/
api/LicenseEdition;<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/Lice
nseProperties;)V

LicenseEdition9clover/com/atlassian/extras/common/util/LicensePropertiesgetProperty&(Ljava/lang/String;)Ljava/l
ang/String;@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolvergetLicenseEditionD(Ljava/lang/
String;)Lclover/com/atlassian/extras/api/LicenseEdition;
this:Lclover/com/atlassian/extras/core/jira/DefaultJiraLicense;product)Lclover/com/atlassian/extras/api/Product;lice
nseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;2()Lclover/com/atlassian/extras/api/Lice
nseEdition;CodeLocalVariableTableLineNumberTable
SourceFile
%*+,
,& !"#'$%/&'(
0!Eclover/com/atlassian/extras/core/DefaultProductLicense$DefaultContactjava/lang/Object'clover/com/atlassian/ex
tras/api/ContactDefaultProductLicense.java6clover/com/atlassian/extras/core/DefaultProductLicenseDefaultContact
nameLjava/lang/String;email<init>'(Ljava/lang/String;Ljava/lang/String;)V()V

thisGLclover/com/atlassian/extras/core/DefaultProductLicense$DefaultContact;getName()Ljava/lang/String;getEma
ilCodeLocalVariableTableLineNumberTable
SourceFileInnerClasses0
Y**+*,
)* +,/*0/*5
0)Fclover/com/atlassian/extras/core/greenhopper/DefaultGreenHopperLicense=clover/com/atlassian/extras/core/plu
gins/DefaultPluginLicense>clover/com/atlassian/extras/api/greenhopper/GreenHopperLicenseDefaultGreenHopperL
icense.javalicenseEdition0Lclover/com/atlassian/extras/api/LicenseEdition;<init>g(Lclover/com/atlassian/extras/api
/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

LicenseEdition9clover/com/atlassian/extras/common/util/LicensePropertiesgetProperty&(Ljava/lang/String;)Ljava/l
ang/String;@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolvergetLicenseEditionD(Ljava/lang/
String;)Lclover/com/atlassian/extras/api/LicenseEdition;
thisHLclover/com/atlassian/extras/core/greenhopper/DefaultGreenHopperLicense;product)Lclover/com/atlassian/ext
ras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;2()Lclover/com/atlas
sian/extras/api/LicenseEdition;CodeLocalVariableTableLineNumberTable

SourceFile

%*+,

,& !"#'$%/&'(

0Jclover/com/atlassian/extras/core/DefaultProductLicense$DefaultOrganisationjava/lang/Object,clover/com/atlassian/extras/api/OrganisationDefaultProductLicense.java6clover/com/atlassian/extras/core/DefaultProductLicenseDefaultOrganisationnameLjava/lang/String;<init>(Ljava/lang/String;)V()V

thisLLclover/com/atlassian/extras/core/DefaultProductLicense$DefaultOrganisation;getName()Ljava/lang/String;CodeLocalVariableTableLineNumberTable

SourceFileInnerClasses0

F

***+

 /*

0@clover/com/atlassian/extras/core/crucible/DefaultCrucibleLicense6clover/com/atlassian/extras/core/DefaultProductLicense8clover/com/atlassian/extras/api/crucible/CrucibleLicenseDefaultCrucibleLicense.java<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

thisBLclover/com/atlassian/extras/core/crucible/DefaultCrucibleLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;CodeLocalVariableTableLineNumberTable

SourceFile I*+,

06clover/com/atlassian/extras/core/DefaultProductLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicenseDefaultProductLicense.javaEclover/com/atlassian/extras/core/DefaultProductLicense$DefaultContactDefaultContactJclover/com/atlassian/extras/core/DefaultProductLicense$DefaultOrganisationDefaultOrganisationEclover/com/atlassian/extras/core/DefaultProductLicense$DefaultPartnerDefaultPartnerMILLIS_IN_A_DAYJ&\licenseVersionIdescriptionLjava/lang/String;product)Lclover/com/atlassian/extras/api/Product;serverIdpartner)Lclover/com/atlassian/extras/api/Partner;organisation.Lclover/com/atlassian/extras/api/Organisation;contactsLjava/util/Collection;ALjava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;creationDateLjava/util/Date;purchaseDatemaximumNumberOfUsers

expiryDategracePeriodEndDatemaintenanceExpiryDatesupportEntitlementNumber

evaluationZsubscriptionlicenseType-Lclover/com/atlassian/extras/api/LicenseType;

properties;Lclover/com/atlassian/extras/common/util/LicenseProperties;<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V()V24

5java/lang/String8valueOf(I)Ljava/lang/String;;;

9<9clover/com/atlassian/extras/common/util/LicenseProperties>getProperty8(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;@A?Bjava/lang/IntegerD'(Ljava/lang/String;)Ljava/lang/Integer;::F

EGintValue()IIJ

EK MDescriptionO&(Ljava/lang/String;)Ljava/lang/String;@Q?R T V

EvaluationXjava/lang/BooleanZ'(Ljava/lang/String;)Ljava/lang/Boolean;:\

[]booleanValue()Z_`

[a+, cSubscriptione-, gServerIDi k

getPartnerf(Lclover/com/atlassian/extras/common/util/LicenseProperties;)Lclover/com/atlassian/extras/api/Partner;mn

o qOrganisations(Ljava/lang/String;)V2u

v xgetContactsS(Lclover/com/atlassian/extras/common/util/LicenseProperties;)Ljava/util/Collection;z{

| !

~CreationDate=clover/com/atlassian/extras/common/LicensePropertiesConstantsDEFAULT_CREATION_DATE$getDate4(Ljava/lang/String;Ljava/util/Date;)Ljava/util/Date;?#$ PurchaseDate%$

LicenseExpiryDateDEFAULT_EXPIRY_DATE$ '$
getGracePeriodEndDate](Lclover/com/atlassian/extras/common/util/LicenseProperties;Ljava/util/Date;)Ljava/util/Date;
($ MaintenanceExpiryDate)$ SEN*
NumberOfUsersgetInt(Ljava/lang/String;I)I?&
LicenseTypeName@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolvergetLicenseTypeA(Ljava/lang/String;)Lclover/com/atlassian/extras/api/LicenseType;
./ 01 this8Lclover/com/atlassian/extras/core/DefaultProductLicense;GracePeriodjava/util/DategetTime()J
(J)V2
gracePeriod
getProduct+()Lclover/com/atlassian/extras/api/Product;getServerId()Ljava/lang/String;+()Lclover/com/atlassian/extras/api/Partner;getOrganisation0()Lclover/com/atlassian/extras/api/Organisation;()Ljava/util/Collection;getCreationDate()Ljava/util/Date;getPurchaseDate
getExpiryDategetNumberOfDaysBeforeExpirygetDaysBeforeDate(Ljava/util/Date;)I
 isExpired
5 compareTo
&getNumberOfDaysBeforeGracePeriodExpiryisWithinGracePeriod`
isGracePeriodExpired`
getMaintenanceExpiryDate&getNumberOfDaysBeforeMaintenanceExpiryisMaintenanceExpiredgetSupportEntitlementNumbergetMaximumNumberOfUsersisUnlimitedNumberOfUsersisEvaluationisSubscriptionnamejava/lang/SystemcurrentTimeMillis
datePartnerName
vpartnerNameContactEMailContactName'(Ljava/lang/String;Ljava/lang/String;)V2
 java/util/Collections
singletonList$(Ljava/lang/Object;)Ljava/util/List;

EMPTY_LISTLjava/util/List;
contactEmailcontactNamegetLicenseVersiongetDescription/()Lclover/com/atlassian/extras/api/LicenseType;
ConstantValue
SignatureCodeLocalVariableTableLineNumberTableC()Ljava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;~(Lclover/com/atlassian/extras/common/util/LicenseProperties;)Ljava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;
SourceFileInnerClasses! !"#$%$&'$($)$*+,-
,./0123*6*,7=CHLN*,PSU*+W*,YS^bd*,fS^bh*,jSl*,pr*Y,tSwy*,}*,*,**,**,**,*,S*,*,S*,
01RBCD&E+F=GOH[IcJvK~LMOPRSTUVWw#,+>Y,ia*##01#'$[]_`/*We/*ljm/*ro/*ytz/*y9Y*~9Y*D*Y*JE***
`H**YD*Y*JE***
`>**`H**YD*Y*JE***
`H**Y/*J/*`8*`/*d`/*h@Q?*+SB+em$
mnR*SL+Y+01

z{y+*SL*SM+, Y,+ +01 "
 'J/*N;
/*U@/*E

0:clover/com/atlassian/extras/core/crowd/DefaultCrowdLicense6clover/com/atlassian/extras/core/DefaultProductLicense2clover/com/atlassian/extras/api/crowd/CrowdLicenseDefaultCrowdLicense.java<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

this<Lclover/com/atlassian/extras/core/crowd/DefaultCrowdLicense;product)Lclover/com/atlassian/extras/api/Product;
properties;Lclover/com/atlassian/extras/common/util/LicenseProperties;CodeLocalVariableTableLineNumberTable
SourceFile I*+,

0<clover/com/atlassian/extras/core/clover/DefaultCloverLicense6clover/com/atlassian/extras/core/DefaultProductLicense4clover/com/atlassian/extras/api/clover/CloverLicenseDefaultCloverLicense.java<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

this>Lclover/com/atlassian/extras/core/clover/DefaultCloverLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;CodeLocalVariableTableLineNumberTable
SourceFile I*+,

0&Dclover/com/atlassian/extras/core/confluence/DefaultConfluenceLicense6clover/com/atlassian/extras/core/DefaultProductLicense<clover/com/atlassian/extras/api/confluence/ConfluenceLicenseDefaultConfluenceLicense.javamaximumNumberClusterNodesI<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

NumberOfClusterNodes9clover/com/atlassian/extras/common/util/LicensePropertiesgetInt(Ljava/lang/String;I)I
thisFLclover/com/atlassian/extras/core/confluence/DefaultConfluenceLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;
maxClustNodesgetMaximumNumberOfClusterNodes()ICodeLocalVariableTableLineNumberTable
SourceFile
"t *+,
,>*#* $!"/*#$%
0>clover/com/atlassian/extras/core/fisheye/DefaultFisheyeLicense6clover/com/atlassian/extras/core/DefaultProductLicense6clover/com/atlassian/extras/api/fisheye/FisheyeLicenseDefaultFisheyeLicense.java<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

this@Lclover/com/atlassian/extras/core/fisheye/DefaultFisheyeLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;CodeLocalVariableTableLineNumberTable
SourceFile I*+,

0Eclover/com/atlassian/extras/core/DefaultProductLicense$DefaultPartnerjava/lang/Object'clover/com/atlassian/extras/api/PartnerDefaultProductLicense.java6clover/com/atlassian/extras/core/DefaultProductLicenseDefaultPartnernameLjava/lang/String;<init>(Ljava/lang/String;)V()V
thisGLclover/com/atlassian/extras/core/DefaultProductLicense$DefaultPartner;getName()Ljava/lang/String;CodeLocalVariableTableLineNumberTable
SourceFileInnerClasses0
F
**+

/*
0<clover/com/atlassian/extras/core/bamboo/DefaultBambooLicense6clover/com/atlassian/extras/core/DefaultProductLicense4clover/com/atlassian/extras/api/bamboo/BambooLicenseDefaultBambooLicense.javaMAX_REMOTE_AGENTS_NONEIMAX_REMOTE_AGENTS_STANDARDMAX_REMOTE_AGENTS_PROFESSIONALMAX_REMOTE_AGENTS_ENTERPRISEMAX_REMOTE_AGENTS_UNLIMITEDdMAX_LOCAL_AGENTS_BASICMAX_LOCAL_AGENTS_UNLIMITEDMAX_PLANS_STARTERMAX_PLANS_UNLIMITEDmaximumNumberOfRemoteAgentsmaximumNumberOfLocalAgentsmaximumNumberOfPlanslicenseEdition0Lclover/com/

atlassian/extras/api/LicenseEdition;<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

LicenseEdition!9clover/com/atlassian/extras/common/util/LicenseProperties#getProperty&(Ljava/lang/String;)Ljava/lang/String;%&$'@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolver)getLicenseEditionD(Ljava/lang/String;)Lclover/com/atlassian/extras/api/LicenseEdition;+,

*- /calculateRemoteAgents>(Lclover/com/atlassian/extras/common/util/LicenseProperties;)I12

3 5calculateLocalAgents72

8 :calculatePlans<2

=

?this>Lclover/com/atlassian/extras/core/bamboo/DefaultBambooLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;2()Lclover/com/atlassian/extras/api/LicenseEdition;getMaximumNumberOfRemoteAgents()IgetMaximumNumberOfLocalAgentsgetMaximumNumberOfPlansisUnlimitedRemoteAgents()ZisUnlimitedLocalAgentsisUnlimitedPlansjava/lang/NumberFormatExceptionPNumberOfBambooRemoteAgentsRjava/lang/StringTlengthVI

UWjava/lang/IntegerYparseInt(Ljava/lang/String;)I[\

Z]+clover/com/atlassian/extras/api/LicenseType_STARTER-Lclover/com/atlassian/extras/api/LicenseType;ab

`cgetLicenseType/()Lclover/com/atlassian/extras/api/LicenseType;ef

gequals(Ljava/lang/Object;)Zij

`k.clover/com/atlassian/extras/api/LicenseEditionmSTANDARDo np

nkPROFESSIONALs nt

ENTERPRISEv nw UNLIMITEDy

nze!Ljava/lang/NumberFormatException;maxRemoteStringLjava/lang/String;NumberOfBambooLocalAgentsBASIC nmaxLocalStringNumberOfBambooPlans

maxPlanString

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile

     1*+, *,"(.0**,46**,9;**,>@
1AB1CD1EF$%'(')0*+G/*0AB.HI/*6AB3JI/*;AB8KI/*@AB=LM8*6ABBNM8*;ABGOM8*@ABL12j+S(M,,X ,^Nd*hlq*0ru*0r

x*0r{*0rdQ*|}jABjEF a~>Q RVX^`b6d8fEhHjUlXnepht72:+(M,,X ,^Nd*hl*0rQ*|}:AB:EF 1& z {')68<2,+(M,,X ,^Nd*hl

Q*|},AB,EF # '*

0_8clover/com/atlassian/extras/core/DefaultAtlassianLicensejava/lang/Object0clover/com/atlassian/extras/api/AtlassianLicenseDefaultAtlassianLicense.javaproductLicenseMapLjava/util/Map;jLjava/util/Map<Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/api/ProductLicense;>;<init>(Ljava/util/Collection;)V()V

java/util/HashMapjava/util/Collectionsize()I(I)V

iterator()Ljava/util/Iterator;java/util/Iterator!hasNext()Z#$"%next()Ljava/lang/Object;'(").clover/com/atlassian/extras/api/ProductLicense+

getProduct+()Lclover/com/atlassian/extras/api/Product;-.,/

java/util/Map1put8(Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object;3425license0Lclover/com/atlassian/extras/api/ProductLicense;i$Ljava/util/Iterator;this:Lclover/com/atlassian/extras/core/DefaultAtlassianLicense;productLicensesHLjava/util/Collection<Lclover/com/atlassian/extras/api/ProductLicense;>;Ljava/util/Collection;getProducLicenses()Ljava/util/Collection;getProductLicensesBA

CvaluesEA2Fjava/util/CollectionsHunmodifiableCollection.(Ljava/util/Collection;)Ljava/util/Collection;JK

ILgetProductLicense[(Lclover/com/atlassian/extras/api/Product;)Lclover/com/atlassian/extras/api/ProductLicense;get&(Ljava/lang/Object;)Ljava/lang/Object;PQ2Rproduct)Lclover/com/atlassian/extras/api/Product;

SignatureCodeLocalVariableTableLocalVariableTypeTableLineNumberTableK(Ljava/util/Collection<Lclover/com/atlassian/extras/api/ProductLicense;>;)V

DeprecatedJ()Ljava/util/Collection<Lclover/com/atlassian/extras/api/ProductLicense;>;

SourceFile V

WD**Y++ M,&!,*,N*-0-6WX*/78'9:D;<D=?YD=>Z/CV[@AW/*DX;<Z"\V]BAW7

*GMX

;<Z+V]NOWB*+S,X;<TUZ0^

0=clover/com/atlassian/extras/core/plugins/DefaultPluginLicense6clover/com/atlassian/extras/core/DefaultProductLicense4clover/com/atlassian/extras/api/plugin/PluginLicenseDefaultPluginLicense.java<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

this?Lclover/com/atlassian/extras/core/plugins/DefaultPluginLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;CodeLocalVariableTableLineNumberTable

SourceFile! I*+,

00clover/com/atlassian/extras/api/jira/JiraLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicense3clover/com/atlassian/extras/api/LicenseEditionAwareJiraLicense.java

SourceFile

0>clover/com/atlassian/extras/api/greenhopper/GreenHopperLicensejava/lang/Object4clover/com/atlassian/extras/api/plugin/PluginLicense3clover/com/atlassian/extras/api/LicenseEditionAwareGreenHopperLicense.java

SourceFile

00clover/com/atlassian/extras/api/AtlassianLicensejava/lang/ObjectAtlassianLicense.javagetProducLicenses()Ljava/util/Collection;getProductLicensesgetProductLicense[(Lclover/com/atlassian/extras/api/Product;)Lclover/com/atlassian/extras/api/ProductLicense;

Deprecated SignatureJ()Ljava/util/Collection<Lclover/com/atlassian/extras/api/ProductLicense;>;

SourceFile

0

4clover/com/atlassian/extras/api/plugin/PluginLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicensePluginLicense.java

SourceFile

0

8clover/com/atlassian/extras/api/crucible/CrucibleLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicenseCrucibleLicense.java

SourceFile

0

2clover/com/atlassian/extras/api/crowd/CrowdLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicenseCrowdLicense.java

SourceFile

0

4clover/com/atlassian/extras/api/clover/CloverLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicenseCloverLicense.java

SourceFile

0<clover/com/atlassian/extras/api/confluence/ConfluenceLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicenseConfluenceLicense.javagetMaximumNumberOfClusterNodes()I

SourceFile

0

6clover/com/atlassian/extras/api/fisheye/FisheyeLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicenseFisheyeLicense.java

SourceFile

04clover/com/atlassian/extras/api/bamboo/BambooLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicense3clover/com/atlassian/extras/api/LicenseEditionAwareBambooLicense.javagetMaximumNumberOfRemoteAgents()IgetMaximumNumberOfLocalAgentsgetMaximumNumberOfPlansisUnlimitedRemoteAgents()ZisUnlimitedLocalAgentsisUnlimitedPlans

SourceFile

0-

.clover/com/atlassian/extras/api/ProductLicensejava/lang/ObjectProductLicense.javagetLicenseVersion()IgetDescription()Ljava/lang/String;

getProduct+()Lclover/com/atlassian/extras/api/Product;getServerId

getPartner+()Lclover/com/atlassian/extras/api/Partner;getOrganisation0()Lclover/com/atlassian/extras/api/Organisation;getContacts()Ljava/util/Collection;getCreationDate()Ljava/util/Date;getPurchaseDate

getExpiryDategetNumberOfDaysBeforeExpiry

isExpired()ZgetGracePeriodEndDate&getNumberOfDaysBeforeGracePeriodExpiryisWithinGracePeriodisGracePeriodExpiredgetSupportEntitlementNumbergetMaintenanceExpiryDate&getNumberOfDaysBeforeMaintenanceExpiryisMaintenanceExpiredgetMaximumNumberOfUsersisUnlimitedNumberOfUsersisEvaluationisSubscriptiongetLicenseType/()Lclover/com/atlassian/extras/api/LicenseType;getProperty&(Ljava/lang/String;)Ljava/lang/String;

SignatureC()Ljava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;

SourceFile

*+ !"#$%&'(),

0com/cenqua/clover/CloverLicensejava/lang/ObjectCloverLicense.java1clover/com/atlassian/extras/common/log/Logger$Log-clover/com/atlassian/extras/common/log/LoggerLog

GENERIC_ERRORLjava/lang/String;Invalid license data

CLOVER_EDITION_PROPERTYclover.license.editionCLOVER_EDITION_PROPERTY_DESKTOPdesktopONE_DAYJ&\ PERMS_ALLPERMS_HIST_PDF@PERMS_HIST_HTML

PERMS_TEST_OPTPERMS_CURR_JSONPERMS_CURR_PDFPERMS_CURR_HTMLPERMS_CURR_XML

PERMS_HIST`

PERMS_CURR

PERMS_DESKTOPproductNamelicenseName

softExpiry

hardExpiryorganisationNameownerStatementpreExpiryStatementpostExpiryStatementcontactInfoStatementterminationStatementsupportEntitlementNumberZallowedPkgPrefixesLjava/util/HashSet;maintExpirysupportedFeaturesreadFrom)(Ljava/io/InputStream;)Ljava/lang/String;(com/atlassian/clover/api/CloverExceptionLjava/io/IOExceptionNjava/lang/StringBufferP<init>()VRS

QTjava/io/LineNumberReaderVjava/io/InputStreamReaderXUTF-8Z*(Ljava/io/InputStream;Ljava/lang/String;)VR\Y](Ljava/io/Reader;)VR_

W`readLine()Ljava/lang/String;bc

Wdappend,(Ljava/lang/String;)Ljava/lang/StringBuffer;fg

Qh

jtoStringlc

QmError reading license. o

getMessageqc

Or*(Ljava/lang/String;Ljava/lang/Throwable;)VRt

MulinelicenseCertLjava/lang/StringBuffer;linLjava/io/LineNumberReader;eLjava/io/IOException;

licenseInLjava/io/InputStream;(Ljava/lang/String;)Vjava/lang/NullPointerException

TconfigureLoggingForExtrasS

Product: Cloverjava/lang/StringindexOf(Ljava/lang/String;)I

Certificate: FCenqua licenses are no longer compatible with this version of Clover:

#com_cenqua_clover/CloverVersionInfoformatVersionInfoc

.

Please visit %http://www.atlassian.com/clover/renew to obtain a new clover.license.R

M6clover/com/atlassian/extras/core/LicenseManagerFactorygetLicenseManager2()Lclover/com/atlassian/extras/api/

LicenseManager;

.clover/com/atlassian/extras/api/LicenseManager

getLicenseF(Ljava/lang/String;)Lclover/com/atlassian/extras/api/AtlassianLicense;'clover/com/atlassian/extras/api/P

roductCLOVER)Lclover/com/atlassian/extras/api/Product;

0clover/com/atlassian/extras/api/AtlassianLicensegetProductLicense[(Lclover/com/atlassian/extras/api/Product;)Lcl

over/com/atlassian/extras/api/ProductLicense;4clover/com/atlassian/extras/api/clover/CloverLicenseNot a Clover

license. getProductLicenses()Ljava/util/Collection;,(Ljava/lang/Object;)Ljava/lang/StringBuffer;f

QisEvaluation()Z

getProduct+()Lclover/com/atlassian/extras/api/Product;getNamec

: getDescriptionc;

getOrganisation0()Lclover/com/atlassian/extras/api/Organisation;,clover/com/atlassian/extras/api/Organisation>

getSupportEntitlementNumbercD getMaintenanceExpiryDate()Ljava/util/Date;java/util/DategetTime()J

H

getExpiryDate= getCreationDate~< License registered to .? 6You have $daysleft day(s) before your license

expires.@ Your license has expired.A C RPlease visit http://www.atlassian.com/ex/GenerateLicense.jspa to obtain a

license.

B allowedpkgprefixesgetProperty&(Ljava/lang/String;)Ljava/lang/String;trimc

length()I

java/util/HashSet

TFG !java/util/StringTokenizer#, %'(Ljava/lang/String;Ljava/lang/String;)VR'

$(

hasMoreTokens*

$+ nextToken-c

$.add(Ljava/lang/Object;)Z01

2equalsIgnoreCase(Ljava/lang/String;)Z45

6E 8I :Invalid license data [E1300]. <

rprefixesLjava/util/StringTokenizer;expiresallowedPkgPrefixesStredition

Ljava/lang/NullPointerException;this!Lcom/cenqua/clover/CloverLicense;manager0Lclover/com/atlassian/extras/ap

i/LicenseManager;atlLlicense2Lclover/com/atlassian/extras/api/AtlassianLicense;license6Lclover/com/atlassian/extr

as/api/clover/CloverLicense;com/cenqua/clover/LicenseLoggerMcom/cenqua/clover/LoggerOgetInstance()Lcom/ce

nqua/clover/Logger;QR

PS(Lcom/cenqua/clover/Logger;)VRU

NVsetInstance6(Lclover/com/atlassian/extras/common/log/Logger$Log;)VXY

Zr(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;JJJ)

VClover](Clover Evaluation License registered to _nameowner preExpiry

postExpiryterminationcontactInfosoftExpiryDatehardExpiryDategetProductNamegetLicenseName

getSoftExpiry

getHardExpirygetMaintExpiry isDesktop isExpiredjava/lang/SystempcurrentTimeMillisr

qs(J)Zou

vA

xtimeisTerminated{u

|

terminates~

isMaintenanceExpiredu

maintenanceExpires

getOwnerStatementgetPreExpiryStatementgetPostExpiryStatementgetContactInfoStatementgetTerminationStatemen

tisFeatureSupportedfeaturegetFeaturesSupportedgetAllowedPkgPrefixes()Ljava/util/Set;java/util/Collectionsunmodi

fiableSet (Ljava/util/Set;)Ljava/util/Set;

getDaysTillExpiry(J)J

 countDays

nowdays

aMilliseconds

ConstantValueCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses1 "#%&()+,./124578:;<=>?@ABCDEFGHI JK[QYULWYYY*[^aM,eN-+-

ikiW,eN+nLMYQYUpi+sin+v:;O4w3xy!z{<|}[~"CDE#F.E6H;I<JMR2 **+9+/MYQYUiiiiinM,+N-:#MYQYUi-

n6******

*a *QYUiiiin* *

* *:B7*Y "$Y&):,*"/3W*":*79**9 8;':MYQYU=i>invf

?@fAEsBC"|DEFxLGHTIJbKL#RSUWH]L^T`bbgcgijklnop s:vLw^xpywz|}~MS<NYTW[EF

R\+

k**^*+*,** *QYU`i,iin*-*

***:*"**9f

kEFkakbkckdkekfkgkh

kIB!>CIOU[`ejic/*EFjc/*EFc/*EFc/*EFk/*EFl/*EFm/*EFn/*9EFo2*twEFouJ*y*EFz{2*t}EF{uJ**EFz2*tEFuJ**

EFzA9* EF~9* EF9* EFc/*EFc/*EFc/*

EFc/*EFc/*EFuE*; EF/*;EF=*"

*"EF2*tEFY**eB! ! EF

:mEF

BSD License

A copy is also included in the downloadable source code package
containing JNA, in file "AL2.0", under the same directory
as this file.
   GNU LESSER GENERAL PUBLIC LICENSE
       Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
   59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

     Preamble

 The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

 This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

 When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

 To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

 For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them

with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free

library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

 In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

 Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

 The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, whereas the latter must
be combined with the library in order to run.

  GNU LESSER GENERAL PUBLIC LICENSE
 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

 A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

 The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

 "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

  (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

 3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

 Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

 This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

 4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

 If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to

distribute the source code, even though third parties are not compelled to copy the source along with the object code.

 5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

 However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

 When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

 If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

 Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

 6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

 You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding
machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

 For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

 It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you

may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

 12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

 13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

 14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is

copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

   NO WARRANTY

 15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

 16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

   END OF TERMS AND CONDITIONS
onoMnApeShTpQtDJbcUgJTIFONPQeUndIgfQWWNNddIwBl
mi2Kp5RjfhIJdGCSo<bOTNof2KNxm9KCi5lxEyKI9BJW3p
qOPQUXpopOopMMPqnPnXXQPNOPNRnqQNQqStwVxuQSTtVW
UrwSUSSSTVwxWSXNmrrpnmqmUUnpsvpntsmmmmmUUnpsvp
ntsmmmmmUUFmbkWJlroZbW4bsbilmjbkqUUnmmmm
Java Native Access project (JNA) is dual-licensed under 2
alternative Open Source/Free licenses: LGPL 2.1 or later and
Apache License 2.0. (starting with JNA version 4.0.0).

You can freely decide which license you want to apply to
the project.

You may obtain a copy of the LGPL License at:

http://www.gnu.org/licenses/licenses.html

A copy is also included in the downloadable source code package
containing JNA, in file "LGPL2.1", under the same directory

as this file.

You may obtain a copy of the Apache License at:

http://www.apache.org/licenses/

A copy is also included in the downloadable source code package
containing JNA, in file "AL2.0", under the same directory
as this file.
1~
V
W
XYZ[
V\
]
^_
`

a
bcd e f g h ij
V kl
V m nocpqrsnameLjava/lang/String;emailurlorganizationorganizationUrlrolesLjava/util/List;
Signature$Ljava/util/List<Ljava/lang/String;>;timezone
propertiesLjava/util/Properties;<init>()VCodeLineNumberTableLocalVariableTablethis$Lorg/apache/maven/model/
Contributor;addProperty'(Ljava/lang/String;Ljava/lang/String;)VkeyvalueaddRole(Ljava/lang/String;)VstringgetEm
ail()Ljava/lang/String;getNamegetOrganizationgetOrganizationUrl
getProperties()Ljava/util/Properties;getRoles()Ljava/util/List;&()Ljava/util/List<Ljava/lang/String;>;getTimezonege
tUrl
removeRolesetEmailsetNamesetOrganizationsetOrganizationUrl
setProperties(Ljava/util/Properties;)VsetRoles(Ljava/util/List;)VLocalVariableTypeTable'(Ljava/util/List<Ljava/lan
g/String;>;)VsetTimezonesetUrl
SourceFileContributor.java-
.@Atujava/lang/Stringjava/lang/ClassCastExceptionjava/lang/StringBuilder:Contributor.addRoles(string) parameter
must be instanceof vwx=<y<-9BCz{|"!
!$!%!+,java/util/Properties&'java/util/ArrayList*!#!=Contributor.removeRoles(string) parameter must be instanceof
}|"org/apache/maven/model/Contributorjava/lang/Objectjava/io/Serializableput8(Ljava/lang/Object;Ljava/lang/Obje
ct;)Ljava/lang/Object;append-
(Ljava/lang/String;)Ljava/lang/StringBuilder;java/lang/ClasstoStringjava/util/Listadd(Ljava/lang/Object;)Zremove!
!"!#!$!%!&'()*!+,-.//*123045/M*+,W1 236!7!0
P
Q89/r2+"YY
*
+W12232:!0Z\&^1_;<//*1230h=<//*1230r></*1230|?<//*1230@A/I**Y*1230BC/I**Y*1230(DE</*1230F</*12
30G9/r2+"YY
*
+W12232:!0&1H9/>*+123"!0
I9/>*+123 !0
J9/>*+123$!0
K9/>*+123%!0

LM/>*+123+,0
NO/P*+123&'P&)0
 (QR9/>*+123*!0
S9/>*+123#!0
TU
1)
! " # $ %&'(nameLjava/lang/String;urldistributioncomments<init>()VCodeLineNumberTableLocalVariableTablethis
Lorg/apache/maven/model/License;getComments()Ljava/lang/String;getDistributiongetNamegetUrlsetComments(Lj
ava/lang/String;)VsetDistributionsetNamesetUrl
SourceFileLicense.java

org/apache/maven/model/Licensejava/lang/Objectjava/io/Serializable!

 /*/*J/*]/*g/*q>*+

{|>*+

>*+

>*+

1
'U
VW &X &Y &Z
V[
&\ &]
'^
_`
Vab
Vc
d`
Ve
Vf
dghihjk
Ulm
n
o
pqhrs
Ut
uvw
xy
z|} classNameLjava/lang/String;nameactionStringactionsLjava/util/Set;
Signature#Ljava/util/Set<Ljava/lang/String;>;<init>()VCodeLineNumberTableLocalVariableTablethis

PermissionInnerClasses3Lorg/apache/tools/ant/types/Permissions$Permission;setClass(Ljava/lang/String;)VaClassgetClassName()Ljava/lang/String;setNameaNamegetName

setActions

getActionsmatches(Ljava/security/Permission;)ZassizeIpermLjava/security/Permission;LocalVariableTypeTableparseActions#(Ljava/lang/String;)Ljava/util/Set;itemresulttkLjava/util/StringTokenizer;7(Ljava/lang/String;)Ljava/util/Set<Ljava/lang/String;>;toString

SourceFilePermissions.java01~=()*)+)KL,-@=*B=Fjava/util/HashSetjava/util/StringTokenizer,0=java/lang/StringBuilderPermission: ("", "")R=1org/apache/tools/ant/types/Permissions$Permissionjava/lang/Objectjava/lang/Stringtrimlength()IgetClass()Ljava/lang/Class;java/lang/Classequals(Ljava/lang/Object;)ZendsWith(Ljava/lang/String;)Zjava/security/Permission substring(II)Ljava/lang/String;

startsWith

java/util/Set removeAll(Ljava/util/Collection;)Z'(Ljava/lang/String;Ljava/lang/String;)V hasMoreTokens()Z nextTokenaddappend-(Ljava/lang/String;)Ljava/lang/StringBuilder;-(Ljava/lang/Object;)Ljava/lang/StringBuilder;&org/apache/tools/ant/types/Permissions!&'()*)+),-./ 012/*34589:2A *+3

4 58 ;)<=2/*3458>:2A *+3

4 58 ?)@=2/*3458A:2V*++**+3 "458,)B=2/*3)458CD2*+

<

+**d*+***+M,>,*W,3>1245&6A7C:Q;S?Z@cAjBuCEH4*cE-jFG58HIJcE/KL2;YMY+N-"-:,W,3"PQRS#T-U6W9X44#M);58;,)3N-(OPJ3N/.QR=2^4Y * ! * " *#$ %3a4458ST7

&{6

/*--

$Id: LICENSE.txt,v 1.11 2004/02/06 09:32:57 jhunter Exp $

In addition, we request (but do not require) that you include in the
end-user documentation provided with the redistribution and/or in the
software itself an acknowledgement equivalent to the following:
    "This product includes software developed by the
    JDOM Project (http://www.jdom.org/)."
Alternatively, the acknowledgment may be graphical using the logos
available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

This software consists of voluntary contributions made by many
individuals on behalf of the JDOM Project and was originally
created by Jason Hunter <jhunter_AT_jdom_DOT_org> and
Brett McLaughlin <brett_AT_jdom_DOT_org>. For more information
on the JDOM Project, please see <http://www.jdom.org/>.

*/

                        Apache License
                  Version 2.0, January 2004
                http://www.apache.org/licenses/

  TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
    and distribution as defined by Sections 1 through 9 of this document.

    "Licensor" shall mean the copyright owner or entity authorized by
    the copyright owner that is granting the License.

    "Legal Entity" shall mean the union of the acting entity and all
    other entities that control, are controlled by, or are under common
    control with that entity. For the purposes of this definition,
    "control" means (i) the power, direct or indirect, to cause the

direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and

wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor

has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==
=========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from 'com.keypoint/org.jfree' to 'clover.com.keypoint/clover.org.jfree'. This was
necessary to avoid potential name conflicts during instrumentation of a code using the original library when
using Clover. No source code of the original library was modified.
==

=========================

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free

software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

  GNU LESSER GENERAL PUBLIC LICENSE
 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not

covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

  (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If

identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

 3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

 Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

 This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

 4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

 If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

 However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

 When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

 If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

 Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

 6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

 You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

  a) Accompany the work with the complete corresponding

machine-readable source code for the Library including whatever
changes were used in the work (which must be distributed under
Sections 1 and 2 above); and, if the work is an executable linked
with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

 For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

 It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

 7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library

facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

  a) Accompany the combined library with a copy of the same work
  based on the Library, uncombined with any other library
  facilities. This must be distributed under the terms of the
  Sections above.

  b) Give prominent notice with the combined library of the fact
  that part of it is a work based on the Library, and explaining
  where to find the accompanying uncombined form of the same work.

 8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

 9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

 10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

 11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by

all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

 12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

 13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

 14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our

decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either

version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
/*
Copyright (c) 2000, Derek Petillo
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

Neither the name of Praxis Software nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/
==
=========================

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package names from the 'org.apache.commons' to the 'clover.org.apache.commons'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.

==
=========================

                        Apache License
                    Version 2.0, January 2004
                 http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
==
==========================
Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed
their package name from 'com.google.json' to 'clover.com.google.json'. This was necessary to
avoid potential name conflicts during instrumentation of a code using the original library when using Clover.
No source code of the original library was modified.
==
==========================

                    Apache License
                Version 2.0, January 2004
              http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

     "License" shall mean the terms and conditions for use, reproduction,

and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,

whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
==========================

Modified by Atlassian

The binary file of the original library has been modified by Atlassian in such way that classes have changed their package name from 'org.apache.commons' to 'clover.org.apache.commons'. This was necessary to avoid potential name conflicts during instrumentation of a code using the original library when using Clover. No source code of the original library was modified.

==
==========================

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

  "License" shall mean the terms and conditions for use, reproduction,
  and distribution as defined by Sections 1 through 9 of this document.

  "Licensor" shall mean the copyright owner or entity authorized by
  the copyright owner that is granting the License.

  "Legal Entity" shall mean the union of the acting entity and all
  other entities that control, are controlled by, or are under common
  control with that entity. For the purposes of this definition,
  "control" means (i) the power, direct or indirect, to cause the
  direction or management of such entity, whether by contract or
  otherwise, or (ii) ownership of fifty percent (50%) or more of the
  outstanding shares, or (iii) beneficial ownership of such entity.

  "You" (or "Your") shall mean an individual or Legal Entity
  exercising permissions granted by this License.

  "Source" form shall mean the preferred form for making modifications,
  including but not limited to software source code, documentation
  source, and configuration files.

  "Object" form shall mean any form resulting from mechanical
  transformation or translation of a Source form, including but
  not limited to compiled object code, generated documentation,
  and conversions to other media types.

  "Work" shall mean the work of authorship, whether in Source or
  Object form, made available under the License, as indicated by a
  copyright notice that is included in or attached to the work

(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses

granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"

/*
File: Core.js

Description:

Provides common utility functions and the Class object used internally by the library.

Also provides the <TreeUtil> object for manipulating JSON tree structures

Some of the Basic utility functions and the Class system are based in the MooTools Framework
<http://mootools.net>. Copyright (c) 2006-2009 Valerio Proietti, <http://mad4milk.net/>. MIT license
<http://mootools.net/license.txt>.

Author:

Nicolas Garcia Belmonte

Copyright:

Copyright 2008-2009 by Nicolas Garcia Belmonte.

Homepage:

<http://thejit.org>

Version:

1.1.2

License:

BSD License

AN ACTION OF CONTRACT, TORT OR

OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR

OTHER DEALINGS IN THE SOFTWARE.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

1. Redistributions of source code must retain the above copyright

   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

   notice, this list of conditions and the following disclaimer in the

   documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its

   contributors may be used to endorse or promote products derived from

   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE

LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF

THE POSSIBILITY OF SUCH DAMAGE.

.!"clover/org/jfree/ui/about/Licencesjava/lang/Object

Licences.javaGPLLjava/lang/String;DGNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA. Everyone is

permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU

General Public License is intended to guarantee your freedom to share and change free software--to make sure the

software is free for all its users. This General Public License applies to most of the Free Software Foundation's

software and to any other program whose authors commit to using it. (Some other Free Software Foundation

software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to

make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish),

that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new

free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public

License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.
LGPLeSGNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over

competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".
A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.
1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty

protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

* a) The modified work must itself be a software library.
* b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
* c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
* d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete

corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

*   a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
*   b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs

one, as long as the modified version is interface-compatible with the version that the work was made with.
*   c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
*   d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
*   e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

*   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
*   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a

consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY

COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!
 singleton$Lclover/org/jfree/ui/about/Licences;<init>()V

thisgetInstance&()Lclover/org/jfree/ui/about/Licences;

getGPL()Ljava/lang/String;getLGPL
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!

/*9 4
Y$%'- 1-;
.%clover/org/jfree/ui/about/Contributorjava/lang/ObjectContributor.javanameLjava/lang/String;email<init>'(Ljava/lang/String;Ljava/lang/String;)V()V
this'Lclover/org/jfree/ui/about/Contributor;getName()Ljava/lang/String;getEmailCodeLocalVariableTableLineNumberTable
SourceFile!
Y*
+, DE FG/*O/*X
    GNU LIBRARY GENERAL PUBLIC LICENSE
        Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.
        59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is
numbered 2 because it goes with version 2 of the ordinary GPL.]

        Preamble

 The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

 This license, the Library General Public License, applies to some
specially designated Free Software Foundation software, and to any
other libraries whose authors decide to use it. You can use it for
your libraries, too.

 When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

 To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.

These restrictions translate to certain responsibilities for you if
you distribute copies of the library, or if you modify it.

 For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link a program with the library, you must provide
complete object files to the recipients so that they can relink them
with the library, after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

 Our method of protecting your rights has two steps: (1) copyright
the library, and (2) offer you this license which gives you legal
permission to copy, distribute and/or modify the library.

 Also, for each distributor's protection, we want to make certain
that everyone understands that there is no warranty for this free
library. If the library is modified by someone else and passed on, we
want its recipients to know that what they have is not the original
version, so that any problems introduced by others will not reflect on
the original authors' reputations.
.
 Finally, any free program is threatened constantly by software
patents. We wish to avoid the danger that companies distributing free
software will individually obtain patent licenses, thus in effect
transforming the program into proprietary software. To prevent this,
we have made it clear that any patent must be licensed for everyone's
free use or not licensed at all.

 Most GNU software, including some libraries, is covered by the ordinary
GNU General Public License, which was designed for utility programs. This
license, the GNU Library General Public License, applies to certain
designated libraries. This license is quite different from the ordinary
one; be sure to read it in full, and don't assume that anything in it is
the same as in the ordinary license.

 The reason we have a separate public license for some libraries is that
they blur the distinction we usually make between modifying or adding to a
program and simply using it. Linking a program with a library, without
changing the library, is in some sense simply using the library, and is
analogous to running a utility program or application program. However, in
a textual and legal sense, the linked executable is a combined work, a
derivative of the original library, and the ordinary General Public License
treats it as such.

 Because of this blurred distinction, using the ordinary General
Public License for libraries did not effectively promote software
sharing, because most developers did not use the libraries. We

concluded that weaker conditions might promote sharing better.

 However, unrestricted linking of non-free programs would deprive the
users of those programs of all benefit from the free status of the
libraries themselves. This Library General Public License is intended to
permit developers of non-free programs to use free libraries, while
preserving your freedom as a user of such programs to change the free
libraries that are incorporated in them. (We have not seen how to achieve
this as regards changes in header files, but we have achieved it as regards
changes in the actual functions of the Library.) The hope is that this
will lead to faster development of free libraries.

 The precise terms and conditions for copying, distribution and
modification follow. Pay close attention to the difference between a
"work based on the library" and a "work that uses the library". The
former contains code derived from the library, while the latter only
works together with the library.

 Note that it is possible for a library to be covered by the ordinary
General Public License rather than by this special one.
 .
   GNU LIBRARY GENERAL PUBLIC LICENSE
 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library which
contains a notice placed by the copyright holder or other authorized
party saying it may be distributed under the terms of this Library
General Public License (also called "this License"). Each licensee is
addressed as "you".

 A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

 The "Library", below, refers to any such software library or work
which has been distributed under these terms. A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language. (Hereinafter, translation is
included without limitation in the term "modification".)

 "Source code" for a work means the preferred form of the work for
making modifications to it. For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

 1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

 You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.
.
 2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices
  stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no
  charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a
  table of data to be supplied by an application program that uses
  the facility, other than as an argument passed when the facility
  is invoked, then you must make a good faith effort to ensure that,
  in the event an application does not supply such function or
  table, the facility still operates, and performs whatever part of
  its purpose remains meaningful.

  (For example, a function in a library to compute square roots has
  a purpose that is entirely well-defined independent of the
  application. Therefore, Subsection 2d requires that any
  application-supplied function or table used by this function must
  be optional: if the application does not supply it, the square
  root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

 3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.
.
 Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

 This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

 4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

 If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not

compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.
.
6. As an exception to the Sections above, you may also compile or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

 For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

 It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.
.
 7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library

facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties to
this License.
.
11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

 12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

 13. The Free Software Foundation may publish revised and/or new
versions of the Library General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.
.
 14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

  NO WARRANTY

 15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO

WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

 16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS
MOZILLA PUBLIC LICENSE
Version 1.1

----

1. Definitions.

1.0.1. "Commercial Use" means distribution or otherwise making the Covered Code available to a third party.

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

    A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

    B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of

this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant.
Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have

made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d)   Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License.
The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

   (a) Third Party Claims.

   If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

   (b) Contributor APIs.

   If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

   (c)   Representations.

   Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor's Modifications are Contributor's original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.

You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to

charge a fee for, warranty, support, indemnity or liability
obligations to one or more recipients of Covered Code. However, You
may do so only on Your own behalf, and not on behalf of the Initial
Developer or any Contributor. You must make it absolutely clear than
any such warranty, support, indemnity or liability obligation is
offered by You alone, and You hereby agree to indemnify the Initial
Developer and every Contributor for any liability incurred by the
Initial Developer or such Contributor as a result of warranty,
support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the
requirements of Section 3.1-3.5 have been met for that Covered Code,
and if You include a notice stating that the Source Code version of
the Covered Code is available under the terms of this License,
including a description of how and where You have fulfilled the
obligations of Section 3.2. The notice must be conspicuously included
in any notice in an Executable version, related documentation or
collateral in which You describe recipients' rights relating to the
Covered Code. You may distribute the Executable version of Covered
Code or ownership rights under a license of Your choice, which may
contain terms different from this License, provided that You are in
compliance with the terms of this License and that the license for the
Executable version does not attempt to limit or alter the recipient's
rights in the Source Code version from the rights set forth in this
License. If You distribute the Executable version under a different
license You must make it absolutely clear that any terms which differ
from this License are offered by You alone, not by the Initial
Developer or any Contributor. You hereby agree to indemnify the
Initial Developer and every Contributor for any liability incurred by
the Initial Developer or such Contributor as a result of any such
terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code
not governed by the terms of this License and distribute the Larger
Work as a single product. In such a case, You must make sure the
requirements of this License are fulfilled for the Covered Code.

4. Inability to Comply Due to Statute or Regulation.

If it is impossible for You to comply with any of the terms of this
License with respect to some or all of the Covered Code due to
statute, judicial order, or regulation then You must: (a) comply with
the terms of this License to the maximum extent possible; and (b)
describe the limitations and the code they affect. Such description
must be included in the LEGAL file described in Section 3.4 and must
be included with all distributions of the Source Code. Except to the

extent prohibited by statute or regulation, such description must be
sufficiently detailed for a recipient of ordinary skill to be able to
understand it.

5. Application of this License.

This License applies to code to which the Initial Developer has
attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

6.1. New Versions.
Netscape Communications Corporation ("Netscape") may publish revised
and/or new versions of the License from time to time. Each version
will be given a distinguishing version number.

6.2. Effect of New Versions.
Once Covered Code has been published under a particular version of the
License, You may always continue to use it under the terms of that
version. You may also choose to use such Covered Code under the terms
of any subsequent version of the License published by Netscape. No one
other than Netscape has the right to modify the terms applicable to
Covered Code created under this License.

6.3. Derivative Works.
If You create or use a modified version of this License (which you may
only do in order to apply it to code which is not already Covered Code
governed by this License), You must (a) rename Your license so that
the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape",
"MPL", "NPL" or any confusingly similar phrase do not appear in your
license (except to note that your license differs from this License)
and (b) otherwise make it clear that Your version of the license
contains terms which differ from the Mozilla Public License and
Netscape Public License. (Filling in the name of the Initial
Developer, Original Code or Contributor in the notice described in
Exhibit A shall not of themselves be deemed to be modifications of
this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS,
WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING,
WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF
DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING.
THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE
IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT,
YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE
COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER

OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above,
all end user license agreements (excluding distributors and resellers)
which have been validly granted by You or any distributor hereunder
prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT
(INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL
DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE,
OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR
ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY
CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL,
WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER
COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN
INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF
LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY
RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW
PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE
EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO
THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in
48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer
software" and "commercial computer software documentation," as such
terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48
C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995),
all U.S. Government End Users acquire Covered Code with only those
rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject
matter hereof. If any provision of this License is held to be
unenforceable, such provision shall be reformed only to the extent
necessary to make it enforceable. This License shall be governed by
California law provisions (except to the extent applicable law, if
any, provides otherwise), excluding its conflict-of-law provisions.
With respect to disputes in which at least one party is a citizen of,
or an entity chartered or registered to do business in the United
States of America, any litigation relating to this License shall be
subject to the jurisdiction of the Federal Courts of the Northern
District of California, with venue lying in Santa Clara County,
California, with the losing party responsible for costs, including
without limitation, court costs and reasonable attorneys' fees and

expenses. The application of the United Nations Convention on
Contracts for the International Sale of Goods is expressly excluded.
Any law or regulation which provides that the language of a contract
shall be construed against the drafter shall not apply to this
License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is
responsible for claims and damages arising, directly or indirectly,
out of its utilization of rights under this License and You agree to
work with Initial Developer and Contributors to distribute such
responsibility on an equitable basis. Nothing herein is intended or
shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as
"Multiple-Licensed". "Multiple-Licensed" means that the Initial
Developer permits you to utilize portions of the Covered Code under
Your choice of the NPL or the alternative licenses, if any, specified
by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

``The contents of this file are subject to the Mozilla Public License
Version 1.1 (the "License"); you may not use this file except in
compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS"
basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
License for the specific language governing rights and limitations
under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____.
Portions created by _____ are Copyright (C) _____
_____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms
of the _____ license (the "[___] License"), in which case the
provisions of [_____] License are applicable instead of those
above. If you wish to allow use of your version of this file only
under the terms of the [____] License and not to allow others to use

your version of this file under the MPL, indicate your decision by

deleting the provisions above and replace them with the notice and

other provisions required by the [___] License. If you do not delete

the provisions above, a recipient may use your version of this file

under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of

the notices in the Source Code files of the Original Code. You should

use the text of this Exhibit A rather than the text found in the

Original Code Source Code for Your Modifications.]

/*

* The contents of this file are subject to the Mozilla Public License Version 1.1

* (the "License"); you may not use this file except in compliance with the License.

* You may obtain a copy of the License at http://www.mozilla.org/MPL/

*

* Software distributed under the License is distributed on an "AS IS" basis,

* WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License

* for the specific language governing rights and limitations under the License.

*

* The Original Code is 'iText, a free JAVA-PDF library'.

*

* The Initial Developer of the Original Code is Bruno Lowagie. Portions created by

* the Initial Developer are Copyright (C) 1999, 2000, 2001, 2002 by Bruno Lowagie.

* All Rights Reserved.

* Co-Developer of the code is Paulo Soares. Portions created by the Co-Developer

* are Copyright (C) 2000, 2001, 2002 by Paulo Soares. All Rights Reserved.

*

* Contributor(s): all the names of the contributors are added in the source code

* where applicable.

*

* Alternatively, the contents of this file may be used under the terms of the

* LGPL license (the "GNU LIBRARY GENERAL PUBLIC LICENSE"), in which case the

* provisions of LGPL are applicable instead of those above. If you wish to

* allow use of your version of this file only under the terms of the LGPL

* License and not to allow others to use your version of this file under

* the MPL, indicate your decision by deleting the provisions above and

* replace them with the notice and other provisions required by the LGPL.

* If you do not delete the provisions above, a recipient may use your version

* of this file under either the MPL or the GNU LIBRARY GENERAL PUBLIC LICENSE.

*

* This library is free software; you can redistribute it and/or modify it

* under the terms of the MPL as stated above or under the terms of the GNU

* Library General Public License as published by the Free Software Foundation;

* either version 2 of the License, or any later version.

*

* This library is distributed in the hope that it will be useful, but WITHOUT

* ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS

* FOR A PARTICULAR PURPOSE. See the GNU Library general Public License for more

* details.
*
* If you didn't download this code from the following link, you should check if
* you aren't using an obsolete version:
* http://www.lowagie.com/iText/
*
* This class is generated based on a grammar file privided by SUN, and updated
* by Carsten Hammer. SUN's license agreement can be found at this URL:
* http://java.sun.com/products/java-media/2D/samples/samples-license.html
* See also the file sun.txt in directory com.lowagie.text.pdf
*/
0(clover/com/atlassian/license/LicensePairjava/lang/ObjectLicensePair.javaNEW_LICENSE_PREFIX[Blicensehash

originalLicenseStringLjava/lang/String;isNGZ<init>([B[B)V-clover/com/atlassian/license/LicenseException()V

startsWith([B[B)Z

 =clover/com/atlassian/extras/decoder/v2/Version2LicenseDecoder!packLicense([B[B)Ljava/lang/String;#$
"%
packV1License'$
(
*this*Lclover/com/atlassian/license/LicensePair;([B[BLjava/lang/String;)VtextoriginalStringiItargetprefix'(Ljava/la
ng/String;Ljava/lang/String;)Vjava/lang/Exception6 License string or hash are null.8(Ljava/lang/String;)V:
;)clover/com/atlassian/license/LicenseUtils=getBytes(Ljava/lang/String;)[B?@
>Ajava/lang/StringBufferC
DException generating license: Fappend,(Ljava/lang/String;)Ljava/lang/StringBuffer;HI
DJ,(Ljava/lang/Object;)Ljava/lang/StringBuffer;HL
DMtoString()Ljava/lang/String;OP
DQeLjava/lang/Exception;contactLicense was
nullUsplitVersion2License>(Ljava/lang/String;)Lclover/com/atlassian/license/LicensePair;WX
Y=clover/com/atlassian/extras/decoder/v1/Version1LicenseDecoder[splitLicense]X
\^
concatLicensepair getString([B)Ljava/lang/String;bc
>djava/lang/Stringflength()Ihi
gj substring(II)Ljava/lang/String;lm
gn
p(I)Ljava/lang/String;lr
gssbLjava/lang/StringBuffer;
hashString
lineLength
licenseStrjava/io/IOExceptionz
" canDecode(Ljava/lang/String;)Z}~
"lastIndexOf(I)I
g()[B?
g-clover/org/apache/commons/codec/binary/Base64decodeBase64([B)[B
java/io/ByteArrayInputStream([B)V
java/io/DataInputStream(Ljava/io/InputStream;)V
readInti
read([B)I

availablei

.

(Ljava/lang/Throwable;)V
licenseContentdecodedBytesinLjava/io/ByteArrayInputStream;dInLjava/io/DataInputStream;
textLengthlicenseTextLjava/io/IOException;encodedLicensepos()Z
getLicensegetLicenseStringgetHash
getHashStringgetOriginalLicenseString<clinit>CodeLocalVariableTableLineNumberTable
Exceptions
SourceFile!

C**+*,*** ** **&***)+ C,-CC "$ %&'B(.#**+*,*** *-+*#,-#/# #0./ 012"3$,+>,+3,3*12$,-$3$478 9;<9">5z*+,
Y9<*+B*,B*** ** **&***)+NYDYEGK-NR<Z]7*^STz,-zz .FGINO&P5QZV]S^UyW:B*+
YV<*+ZM,+_M*,++*,*,*, B,-B`\*a-.`acfhj!m)n1o9pAq'$8DYEN,e:kl6k#-oKW-qKWt:-KW-qKW+e:k#-oKW-
qKWt:-KW-qKW-RH,- uvw{x2U=yFuwxz!|.}5~AHOU_lsWXH
z"Y|++X=+oN-:Y:Y:6:W: WY +NY-op{pR'I2>=3D,2J&[q Sz,-zc2>'2=DJR[cpq/* ,-/*,-P2*e,-/*,-P2*e,-P/*+,-
OP/*+,-8 Y
TYTYTY
TYT
0!2clover/com/atlassian/license/MemoryLicenseRegistryjava/lang/Object,clover/com/atlassian/license/LicenseRegis
tryMemoryLicenseRegistry.javaLICENSELjava/lang/String;HASH<init>()V

this4Lclover/com/atlassian/license/MemoryLicenseRegistry;setLicenseMessage(Ljava/lang/String;)V
licenseMessagesetLicenseHash
 licenseHashgetLicenseMessage()Ljava/lang/String;getLicenseHashCodeLocalVariableTableLineNumberTable
SourceFile!

 /*=+

=+
..
03clover/com/atlassian/license/decoder/LicenseAdaptorjava/lang/Object$clover/com/atlassian/license/LicenseLicen
seAdaptor.java_1_YEARJ\licenseType*Lclover/com/atlassian/license/LicenseType;creationDateLjava/util/Date;pur
chaseDate
expiryDatemaintenanceExpiryDate
evaluationZsupportEntitlementNumberLjava/lang/String;permittedClusterNodesIorganisationpartnermaximumNum
berOfUsers<init>h(Lclover/com/atlassian/extras/common/util/LicenseProperties;Lclover/com/atlassian/license/Lice
nseType;)V()V

!CreationDate#9clover/com/atlassian/extras/common/util/LicenseProperties%getProperty&(Ljava/lang/String;)Ljav
a/lang/String;'(&)-clover/com/atlassian/extras/common/DateEditor+getDate$(Ljava/lang/String;)Ljava/util/Date;-.
,/ 1LicenseExpiryDate3 5
Evaluation7java/lang/Boolean9valueOf'(Ljava/lang/String;)Ljava/lang/Boolean;;<
:=booleanValue()Z?@
:A CPurchaseDateE GMaintenanceExpiryDateI KSENM
ONumberOfClusterNodesQgetInt(Ljava/lang/String;I)IST&U WOrganisationY [PartnerName] _
NumberOfUsersa
cthis5Lclover/com/atlassian/license/decoder/LicenseAdaptor;licenseProperties;Lclover/com/atlassian/extras/commo

n/util/LicenseProperties;maxUsersgetDateCreated()Ljava/util/Date;java/util/DatelgetTime()Jno

mp(J)Vr

msupdatedCreationDategetDatePurchased

getExpiryDate(clover/com/atlassian/license/LicenseTypexexpiresz@y{getLicenseDurationgetLicenseId()Ljava/lang/String;getLicenseType,()Lclover/com/atlassian/license/LicenseType;getOrganisationgetPartnerNamegetPermittedClusteredNodes()IgetUsersrequiresUserLimit@y isExpiredjava/lang/SystemcurrentTimeMilliso

isLicenseLevel(Ljava/util/Collection;)Z

getDescriptionyjava/lang/StringtoLowerCase

java/util/Collectioniterator()Ljava/util/Iterator;java/util/IteratorhasNext@next()Ljava/lang/Object;indexOf(Ljava/lang/String;)I

levelLjava/util/Iterator;levelsLjava/util/Collection;descriptiongetSupportEntitlementNumber

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile!

* *,"*+$*02*+4*06*+8*>BD*+F*0H*+J*0L*+N*P*+RVX*+Z**+^*`+bV>* 'd*efgh

i: !'"9#H$W%c&p'|()*+jkx.*LmY*Lq

etLmY*2qtL+ u.ef,u026,8vk/*Hef=wkF*"|

*D

*6efB}o, efG~/*PefL/*"efQ/*\efV/*`ef[/*Xef`E*"*defegk@E*6*6qefq<*M+N- -:,4(&<ef<

/v

wy(z5|7~:/*Pef

03clover/com/atlassian/license/decoder/LicenseDecoderjava/lang/ObjectLicenseDecoder.java1clover/com/atlassian/extras/common/log/Logger$Log-

clover/com/atlassian/extras/common/log/LoggerLoglog3Lclover/com/atlassian/extras/common/log/Logger$Log;DURATION_PREFIXLjava/lang/String; Duration:JIRA_APPLICATION_NAMEJIRACONF_APPLICATION_NAMECONFLUENCE9class$clover$com$atlassian$license$decoder$LicenseDecoderLjava/lang/Class;<init>()V

this5Lclover/com/atlassian/license/decoder/LicenseDecoder;

getLicensed(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)Lclover/com/atlassian/license/License;-clover/com/atlassian/license/LicenseException!getPublicKey-(Ljava/lang/String;)Ljava/security/PublicKey;#$%loadLicense}(Lclover/com/atlassian/license/LicensePair;Ljava/security/PublicKey;Ljava/lang/String;)Lclover/com/atlassian/license/License;'(

)java/lang/RuntimeException+

getMessage()Ljava/lang/String;-.

"/*(Ljava/lang/String;Ljava/lang/Throwable;)V1

,2e/Lclover/com/atlassian/license/LicenseException;pair*Lclover/com/atlassian/license/LicensePair;applicationNameisValid?(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)Z

;java/lang/Exception=getPublicKeyFilename&(Ljava/lang/String;)Ljava/lang/String;?@

ACjava/lang/StringEequals(Ljava/lang/Object;)ZGH

FI]The filename for the public key is null. This must be set before a public key can be

located.K(Ljava/lang/String;)VM

"NloadPublicKeyFromFileP$

Q Sjava/lang/StringBufferU

V!Exception looking up public key: Xappend,(Ljava/lang/String;)Ljava/lang/StringBuffer;Z[

V\

>/toString_.

V`error*(Ljava/lang/Object;Ljava/lang/Throwable;)VbcdfException getting verification from file - possible

classloader problem, or corrupt JIRA installation

fLjava/lang/Exception;publicKeyFileNamejava/io/IOExceptionj&java/security/NoSuchAlgorithmExceptionl*java/s

ecurity/spec/InvalidKeySpecExceptionnjava/lang/Threadp

currentThread()Ljava/lang/Thread;rs

qtgetContextClassLoader()Ljava/lang/ClassLoader;vw

qxjava/lang/ClassLoaderzgetResourceAsStream)(Ljava/lang/String;)Ljava/io/InputStream;|}

{~ getClass()Ljava/lang/Class;

java/lang/ClassgetComponentType

getClassLoaderw

)clover/com/atlassian/license/LicenseUtilsreadKey(Ljava/io/InputStream;)[B

java/io/InputStreamclose

%java/security/spec/X509EncodedKeySpec([B)V

DSAjava/security/KeyFactorygetInstance.(Ljava/lang/String;)Ljava/security/KeyFactory;

generatePublic7(Ljava/security/spec/KeySpec;)Ljava/security/PublicKey;

keyfisLjava/io/InputStream;

contextLoaderLjava/lang/ClassLoader;encKey[B

pubKeySpec'Ljava/security/spec/X509EncodedKeySpec;

keyFactoryLjava/security/KeyFactory;(clover/com/atlassian/license/LicensePairisNG()Z

parseNewLicense

parseOldLicense(

(Ljava/lang/Object;)Vb publicKeyLjava/security/PublicKey;!java/security/InvalidKeyException

java/security/SignatureExceptionSHA1withDSAjava/security/Signature-

(Ljava/lang/String;)Ljava/security/Signature;

initVerify(Ljava/security/PublicKey;)V

()[B

update

getHash

verify([B)Z

getDecodedMessage([B)Ljava/lang/String;

java/util/StringTokenizer^'(Ljava/lang/String;Ljava/lang/String;)V

hasMoreTokens

 nextToken.

java/lang/IntegerparseInt(Ljava/lang/String;)I

+clover/com/atlassian/license/LicenseManager/()Lclover/com/atlassian/license/LicenseManager;

getLicenseType?(Ljava/lang/String;I)Lclover/com/atlassian/license/LicenseType;

-clover/com/atlassian/extras/common/DateEditorgetDate$(Ljava/lang/String;)Ljava/util/Date;

(clover/com/atlassian/license/LicenseTypeexpires

getOriginalLicenseString.

getLicenseIdFromLicenseString@

requiresUserLimit\|split'(Ljava/lang/String;)[Ljava/lang/String;

F%License contained invalid user

limit:+clover/com/atlassian/license/DefaultLicense(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String

;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;ILjava/lang/String;)V

"Signature did not verify

properly."warn$%usersAndClustersLimitsparts[Ljava/lang/String;licenseTypeCodeIlicenseType*Lclover/com/atlas

sian/license/LicenseType;dateCreatedLjava/util/Date;

datePurchaseddateExpiresorganisation licenseIdusersclusterCountpartnerName

---

messageString tokenizerLjava/util/StringTokenizer;

signatureLjava/security/Signature;=clover/com/atlassian/extras/decoder/v2/Version2LicenseDecoder<

= canDecode(Ljava/lang/String;)Z?@

=A Failed to decode as V2 license:

C,(Ljava/lang/Object;)Ljava/lang/StringBuffer;ZE

VFdecode*(Ljava/lang/String;)Ljava/util/Properties;HI

=J

lookupProduct=(Ljava/lang/String;)Lclover/com/atlassian/extras/api/Product;LM

N@clover/com/atlassian/extras/common/util/ProductLicensePropertiesPB(Lclover/com/atlassian/extras/api/Product;

Ljava/util/Properties;)VR

QSLicenseEditionUgetPropertyW@

QXLicenseTypeZP(Ljava/lang/String;Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;\

]LicenseTypeName_

Evaluationajava/lang/BooleancvalueOf'(Ljava/lang/String;)Ljava/lang/Boolean;ef

dgbooleanValuei

dj@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolverlgetLicenseEditionD(Ljava/lang/String;)

Lclover/com/atlassian/extras/api/LicenseEdition;no

mp>clover/com/atlassian/extras/legacy/util/OldLicenseTypeResolverr(Lclover/com/atlassian/extras/api/Product;Lja

va/lang/String;ZLclover/com/atlassian/extras/api/LicenseEdition;)Lclover/com/atlassian/license/LicenseType;t

su3clover/com/atlassian/license/decoder/LicenseAdaptorwh(Lclover/com/atlassian/extras/common/util/LicenseProp

erties;Lclover/com/atlassian/license/LicenseType;)Vy

xzlicenseDecoder?Lclover/com/atlassian/extras/decoder/v2/Version2LicenseDecoder;propLjava/util/Properties;prod

uct)Lclover/com/atlassian/extras/api/Product;productPropertiesBLclover/com/atlassian/extras/common/util/Product

LicenseProperties;editionNamelicenseTypeString'clover/com/atlassian/extras/api/Productvalues,()[Lclover/com/atla

ssian/extras/api/Product;

getName.

equalsIgnoreCase@

FgetNamespace.

replace(CC)Ljava/lang/String;

F"java/lang/IllegalArgumentException Could not find product for key <>

Narr$*[Lclover/com/atlassian/extras/api/Product;len$i$keylength()I

F

-clover/com/atlassian/license/util/StringUtils

replaceAllJ(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;

substring(II)Ljava/lang/String;

FtoUpperCase.

ForiginalLicenseString$java/io/UnsupportedEncodingExceptionUTF-8([BLjava/lang/String;)V

F(Ljava/lang/Throwable;)V

,&Ljava/io/UnsupportedEncodingException;messageX(Lclover/com/atlassian/license/LicensePair;Ljava/security/Pu

blicKey;Ljava/lang/String;)ZgetLicenseTypeStoreC(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseTypeS

tore;

-clover/com/atlassian/license/LicenseTypeStoregetPublicKeyFileName.

<clinit>F(Ljava/lang/Class;)Lclover/com/atlassian/extras/common/log/Logger$Log;

ConstantValue SyntheticCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses!

/*# f*+&+*M,Y,0,3
"
45678.
02 9:B*+<6788
#$K*BL+D+J
"YLO+RMTVYWY],^]a,e"YgO !> ")4hK8Fi=?@D!F"HAI" P$WLuyM, ,*L+!YWY*L+N+Y-::>WiU N:HO.OP R
SVW5Z:[>]H^O`kmo
'(* *,*+,NT->
>*4h 67 8gk
ortu (-N-+-*-**:Y:6,::: :

:
:*:6
6X::' "YVYW]]aO26
26:
:Y

!T#&N'D()F*+Q,-[./e0/ h1/
234+
5+*6'7489-67--8':;!{|}'4<FQ[ehr| +
m"
X
=Y>M,*B"YVYWD]*GaO,*KN+O:QY-T:VY:[Y:+^:.`YbYhkqv:xY{f
n,-678|}8p~>jJ^TT,-^J2/8>JT^cq"
LM]L+=>0+2:** _YVYW]*]]a4"74+ 2+]25;
@f4*0*F'*DDD;ED4
1
ZFY*L,Y+
 4
 9M*+,* 678
?@5*85YWYT%

0Cclover/com/atlassian/license/applications/jira/JiraLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreJiraLicenseTypeStore.javaAPPLICATION_NAMELjava/lang/String;
JIRAJIRA_STANDARD_ACADEMIC*Lclover/com/atlassian/license/LicenseType;JIRA_STANDARD_EVALUA
TIONJIRA_STANDARD_NON_PROFITJIRA_STANDARD_FULL_LICENSEJIRA_STANDARD_COMMUNI
TYJIRA_STANDARD_OPEN_SOURCEJIRA_STANDARD_DEVELOPERJIRA_STANDARD_DEMONSTRAT
IONJIRA_STANDARD_PERSONALJIRA_STANDARD_STARTERJIRA_PROFESSIONAL_ACADEMICJIRA
_PROFESSIONAL_EVALUATIONJIRA_PROFESSIONAL_NON_PROFITJIRA_PROFESSIONAL_FULL_LIC
ENSEJIRA_PROFESSIONAL_COMMUNITYJIRA_PROFESSIONAL_OPEN_SOURCEJIRA_PROFESSIONAL
_DEVELOPERJIRA_PROFESSIONAL_DEMONSTRATIONJIRA_PROFESSIONAL_PERSONALJIRA_PROFE
SSIONAL_STARTERJIRA_ENTERPRISE_ACADEMICJIRA_ENTERPRISE_EVALUATIONJIRA_ENTERPRI
SE_NON_PROFITJIRA_ENTERPRISE_FULL_LICENSEJIRA_ENTERPRISE_HOSTEDJIRA_ENTERPRISE_C
OMMUNITYJIRA_ENTERPRISE_OPEN_SOURCEJIRA_ENTERPRISE_DEVELOPERJIRA_ENTERPRISE_D
EMONSTRATIONJIRA_ENTERPRISE_TESTINGJIRA_ENTERPRISE_PERSONALJIRA_ENTERPRISE_STA
RTERpublicKeyFileNameprivateKeyFileName<init>()V-.
/applicationLicenseTypesLjava/util/ArrayList;12 3

5java/util/ArrayList7add(Ljava/lang/Object;)Z9:

8; =

? A C E G I K M O Q S U W Y [ ] _ a c e! g" i# k$ m% o& q' s(u) w*

ythisELclover/com/atlassian/license/applications/jira/JiraLicenseTypeStore;getAllLicenses()Ljava/util/Collection;getPublicKeyFileName()Ljava/lang/String;+ getPrivateKeyFileName,

<clinit>/clover/com/atlassian/license/DefaultLicenseTypeJIRA Standard:

Academic+clover/com/atlassian/extras/api/LicenseTypeACADEMIC-

Lclover/com/atlassian/extras/api/LicenseType; name

.clover/com/atlassian/extras/api/LicenseEditionSTANDARD0Lclover/com/atlassian/extras/api/LicenseEdition;

Z(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V-

JIRA Standard: Evaluation

COMMERCIAL 'JIRA Standard: Non-Profit / Open Source

NON_PROFIT JIRA Standard: Commercial ServerJIRA Standard: Community COMMUNITY JIRA Standard: Open

SourceOPEN_SOURCE JIRA Standard: Developer DEVELOPER JIRA Standard: Demonstration

DEMONSTRATION JIRA Standard: PersonalPERSONAL JIRA Standard: StarterSTARTER JIRA Professional:

AcademicPROFESSIONAL JIRA Professional: Evaluation+JIRA Professional: Non-Profit / Open Source$JIRA

Professional: Commercial ServerJIRA Professional: CommunityJIRA Professional: Open SourceJIRA Professional:

Developer JIRA Professional: DemonstrationJIRA Professional: PersonalJIRA Professional: StarterJIRA Enterprise:

Academic

ENTERPRISE JIRA Enterprise: Evaluation)JIRA Enterprise: Non-Profit / Open Source"JIRA Enterprise:

Commercial ServerJIRA Enterprise: HostedHOSTED JIRA Enterprise: CommunityJIRA Enterprise: Open

SourceJIRA Enterprise: DeveloperJIRA Enterprise: DemonstrationJIRA Enterprise: TestingTESTING

[(ILjava/lang/String;ZZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V-

JIRA Enterprise: PersonalJIRA Enterprise: Starter"clover/com/atlassian/jira/leaf.keyjira/jira.byte

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile!#

        ! " # $ % & ' ( ) * + ,-

.e*0*46<W*4><W*4@<W*4B<W*4D<W*4F<W*4H<W*4J<W*4L<W*4N<W*4P<W*4R<W*4T<W*4V<W*4

X<W*4Z<W*4\<W*4^<W*4`<W*4b<W*4d<W*4f<W*4h<W*4j<W*4l<W*4n<W*4p<W*4r<W*4t<W*4v<W*

4x<W*4z<We{| "6789%:0;;<F=Q>\?g@rB}CDEFGHIJKMNOPQR"S-T8UCVNWYXdY}~/*4{|].{| b.{| g.

BY6Ym>Y@YBYkDYzFYHYJYLYNY[PY/RYLTYWVY*XY'ZYR\Yc^YY`YZbY

dYfYhYjYlYnYpYrYtYvY!xY"z "3Mg3Le~ !"$%/&I'c(})*+,-./52;3

0nQclover/com/atlassian/license/applications/greenhopper/GreenHopperLicenseTypeStore-

clover/com/atlassian/license/LicenseTypeStore

GreenHopperLicenseTypeStore.javaNAMELjava/lang/String;GreenHopper!GREENHOPPER_STANDARD_FULL

_LICENSE*Lclover/com/atlassian/license/LicenseType;%GREENHOPPER_PROFESSIONAL_FULL_LICENSE#

GREENHOPPER_ENTERPRISE_FULL_LICENSE!GREENHOPPER_ENTERPRISE_EVALUATIONGREENH

OPPER_ENTERPRISE_ACADEMIC"GREENHOPPER_ENTERPRISE_OPEN_SOURCEGREENHOPPER_ENT

ERPRISE_PERSONAL<init>()V

applicationLicenseTypesLjava/util/ArrayList;

 java/util/ArrayListadd(Ljava/lang/Object;)Z

 "

 $ & (*

,thisSLclover/com/atlassian/license/applications/greenhopper/GreenHopperLicenseTypeStore;getPublicKeyFileName()Ljava/lang/String;)clover/com/atlassian/greenhopper/leaf.key2getPrivateKeyFileNamegreenhopper/greenhopper.

byte5<clinit>/clover/com/atlassian/license/DefaultLicenseType8'GreenHopper Standard: Commercial

Server:+clover/com/atlassian/extras/api/LicenseType<

COMMERCIAL-Lclover/com/atlassian/extras/api/LicenseType;>? =@nameB1
=C.clover/com/atlassian/extras/api/LicenseEditionESTANDARD0Lclover/com/atlassian/extras/api/LicenseEdition;
GH FIZ(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)VK
9L+GreenHopper Professional: Commercial ServerNPROFESSIONALPH FQ)GreenHopper Enterprise:
Commercial ServerS
ENTERPRISEUH FV"GreenHopper Enterprise: EvaluationX GreenHopper Enterprise: AcademicZACADEMIC\?
=]#GreenHopper Enterprise: Open Source_OPEN_SOURCEa? =b GreenHopper Enterprise:
PersonaldPERSONALf? =g
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!i

jR**!W*#!W*%!W*'!W*)!W*+!W*-!WkR./l& %0; F!Q"01j-3k./l&41j-
6k./l+7j9Yx;ADJM9YyOADRM#9YzTADWM%9Y{YADWM'9Y|[^DWM)9Y}`cDWM+9Y~ehDWM-l4Nhm
0tKclover/com/atlassian/license/applications/crucible/CrucibleLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreCrucibleLicenseTypeStore.javaCRUCIBLE_ACADEMIC*Lclover/c
om/atlassian/license/LicenseType;CRUCIBLE_COMMERCIALCRUCIBLE_COMMUNITYCRUCIBLE_EVALU
ATIONCRUCIBLE_OPEN_SOURCECRUCIBLE_DEVELOPERCRUCIBLE_STARTERCRUCIBLE_DEMONS
TRATIONpublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
" $
& (*
,
.thisMLclover/com/atlassian/license/applications/crucible/CrucibleLicenseTypeStore;getAllLicenses()Ljava/util/Col
lection;getPublicKeyFileName()Ljava/lang/String; 6getPrivateKeyFileName
9<clinit>/clover/com/atlassian/license/DefaultLicenseType<Crucible:
Academic>+clover/com/atlassian/extras/api/LicenseType@ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;BC ADnameF5
AG*(ILjava/lang/String;ZZLjava/lang/String;)VI
=JCrucible: CommercialL
COMMERCIALNC AOCrucible: CommunityQ COMMUNITYSC ATCrucible: EvaluationVCrucible: Open
SourceXOPEN_SOURCEZC A[Crucible: Developer] DEVELOPER_C A`Crucible: StarterbSTARTERdC
AeCrucible: Demonstrationg
DEMONSTRATIONiC
Aj&clover/com/atlassian/crucible/leaf.keylcrucible/crucible.bytenCodeLocalVariableTableLineNumberTable
SourceFile!

  p]**!W*#!W*%!W*'!W*)!W*+!W*-!W*/!Wq]01r*
%0;
F!Q"\#23p/*q01r'45p.7q01r,85p.:q01r1;p=YL?EHK=YVMPHK#=Y`RUHK%=YjWPHK'=YtY\HK)=Y~^aHK+=Y
cfHK-=YhkHK/m7o:r*

.E\ss
0yUclover/com/atlassian/license/applications/sharepoint/SharePointPluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStore%SharePointPluginLicenseTypeStore.javaSP_PLUGIN_APPNAME
Ljava/lang/String;SharePoint
PluginAPPLICATION_NAMESHAREPOINT_ACADEMIC*Lclover/com/atlassian/license/LicenseType;SHAREP
OINT_EVALUATIONSHAREPOINT_DEMONSTRATIONSHAREPOINT_NON_PROFITSHAREPOINT_COM

MUNITYSHAREPOINT_DEVELOPERSHAREPOINT_OPEN_SOURCESHAREPOINT_FULL_LICENSEpublic
KeyFileNameprivateKeyFileName<init>()V

applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayList add(Ljava/lang/Object;)Z"#

!$

 & (* , . 0

2thisWLclover/com/atlassian/license/applications/sharepoint/SharePointPluginLicenseTypeStore;getAllLicenses()Lj
ava/util/Collection;getPublicKeyFileName()Ljava/lang/String; :getPrivateKeyFileName

=<clinit>/clover/com/atlassian/license/DefaultLicenseType@SharePoint:
AcademicB+clover/com/atlassian/extras/api/LicenseTypeDACADEMIC-

Lclover/com/atlassian/extras/api/LicenseType;FG EHnameJ9

EK*(ILjava/lang/String;ZZLjava/lang/String;)VM

ANSharePoint: EvaluationP

COMMERCIALRG ESSharePoint: DemonstrationU

DEMONSTRATIONWG EX$SharePoint: Non-Profit / Open SourceZ

NON_PROFIT\G E]SharePoint: Community_ COMMUNITYaG EbSharePoint: Developerd DEVELOPERfG

EgSharePoint: Open SourceiOPEN_SOURCEkG ElSharePoint: Commercialn-
clover/com/atlassian/confluence/page/Page.keypconfluence/confluence.byter

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile!t

t

    u]**%W*'%W*)%W*+%W*-%W*/%W*1%W*3%Wv]45w*

%'()%W*0+;,F-Q.\/67u/*v45w389u.;v45w8<9u.>v45w=?uAYCILOAYQTLO'AY(VYLO/AY2[^LO)AY<`cLO-
AYFehLO1AYPjmLO3AYZoTLO+q;s>w*

.E\s!"x

0tMclover/com/atlassian/license/applications/vssplugin/VSSPluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreVSSPluginLicenseTypeStore.javaVSS_ACADEMIC*Lclover/com/at
lassian/license/LicenseType;VSS_EVALUATIONVSS_NON_PROFITVSS_FULL_LICENSE

VSS_COMMUNITY

VSS_DEVELOPERVSS_DEMONSTRATIONVSS_OPEN_SOURCEpublicKeyFileNameLjava/lang/String;private
KeyFileName<init>()V

applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z

" $

& (*

,

.thisOLclover/com/atlassian/license/applications/vssplugin/VSSPluginLicenseTypeStore;getAllLicenses()Ljava/util/
Collection;getPublicKeyFileName()Ljava/lang/String; 6getPrivateKeyFileName

9<clinit>/clover/com/atlassian/license/DefaultLicenseType<

VSS: Academic>+clover/com/atlassian/extras/api/LicenseType@ACADEMIC-

Lclover/com/atlassian/extras/api/LicenseType;BC ADnameF5

AG*(ILjava/lang/String;ZZLjava/lang/String;)VI

=JVSS: EvaluationL

COMMERCIALNC AOVSS: Non-Profit / Open SourceQ

NON_PROFITSC ATVSS: CommercialVVSS: CommunityX COMMUNITYZC A[VSS: Developer]

DEVELOPER_C A`VSS: Demonstrationb

DEMONSTRATIONdC AeVSS: Open SourcegOPEN_SOURCEiC

AjCclover/com/atlassian/license/applications/jira/JiraLicenseTypeStorel m6

m9CodeLocalVariableTableLineNumberTable

SourceFile!

p]**!W*#!W*%!W*'!W*)!W*+!W*-!W*/!Wq]01r*

!"#%$0%;&F'Q(\)23p/*q01r-

45p.7q01r285p.:q01r7;p=Y@?EHK=YAMPHK#=YBRUHK%=YCWPHK'=YDY\HK)=YE^aHK+=YFcfHK-
=YGhkHK/n7o:r*

.E\ss

0lEclover/com/atlassian/license/applications/crowd/CrowdLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreCrowdLicenseTypeStore.javaCROWD_ACADEMIC*Lclover/com/a
tlassian/license/LicenseType;CROWD_COMMERCIALCROWD_COMMUNITYCROWD_EVALUATIONCRO
WD_OPEN_SOURCECROWD_DEVELOPERCROWD_DEMONSTRATIONpublicKeyFileNameLjava/lang/Strin
g;privateKeyFileName<init>()V

applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z

! #

% ')

+thisGLclover/com/atlassian/license/applications/crowd/CrowdLicenseTypeStore;getAllLicenses()Ljava/util/Collect
ion;getPublicKeyFileName()Ljava/lang/String; 3getPrivateKeyFileName

6<clinit>/clover/com/atlassian/license/DefaultLicenseType9Crowd:

Academic;+clover/com/atlassian/extras/api/LicenseType=ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;?@ >AnameC2

>D*(ILjava/lang/String;ZZLjava/lang/String;)VF

:GCrowd: CommercialI

COMMERCIALK@ >LCrowd: CommunityN COMMUNITYP@ >QCrowd: EvaluationSCrowd: Open

SourceUOPEN_SOURCEW@ >XCrowd: DeveloperZ DEVELOPER\@ >]Crowd: Demonstration_

DEMONSTRATIONa@

>b#clover/com/atlassian/crowd/leaf.keydcrowd/crowd.bytefCodeLocalVariableTableLineNumberTable

SourceFile!

hR** W*" W*$ W*& W*(W** W*, WiR-.j& %0; F!Q"/0h/*i-.j&12h.4i-.j+52h.7i-
.j08h:YY<BEH:YaJMEH":YiOREH$:YqTMEH&:YyVYEH(:Y[^EH*:Y`cEH,e4g7j&

.E\sk

0tWclover/com/atlassian/license/applications/perforceplugin/PerforcePluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStore#PerforcePluginLicenseTypeStore.javaPERFORCE_ACADEMIC*L
clover/com/atlassian/license/LicenseType;PERFORCE_EVALUATIONPERFORCE_DEMONSTRATIONPERFO
RCE_NON_PROFITPERFORCE_COMMUNITYPERFORCE_DEVELOPERPERFORCE_OPEN_SOURCEPERF
ORCE_FULL_LICENSEpublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z

"

$ & (* ,

.thisYLclover/com/atlassian/license/applications/perforceplugin/PerforcePluginLicenseTypeStore;getAllLicenses()L
java/util/Collection;getPublicKeyFileName()Ljava/lang/String; 6getPrivateKeyFileName

9<clinit>/clover/com/atlassian/license/DefaultLicenseType<Perforce:

Academic>+clover/com/atlassian/extras/api/LicenseType@ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;BC ADnameF5

AG*(ILjava/lang/String;ZZLjava/lang/String;)VI

=JPerforce: EvaluationL

COMMERCIALNC AOPerforce: DemonstrationQ

DEMONSTRATIONSC AT"Perforce: Non-Profit / Open SourceV

NON_PROFITXC AYPerforce: Community[COMMUNITY]C A^Perforce: Developer` DEVELOPERbC AcPerforce: Open SourceeOPEN_SOURCEgC AhPerforce: CommercialjCclover/com/atlassian/license/applications/jira/JiraLicenseTypeStorel m6 m9CodeLocalVariableTableLineNumberTable
SourceFile!

   p]**!W*#!W*%!W*'!W*)!W*+!W*-!W*/!Wq]01r*
!"%#0$;%F&Q'\(23p/*q01r,45p.7q01r185p.:q01r6;p=Y3?EHK=Y7MPHK#=Y8RUHK+=Y9WZHK%=Y:_HK)=Y
;adHK-=Y<fiHK/=Y=kPHK'n7o:r*
.E\ss
0lGclover/com/atlassian/license/applications/clover/CloverLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreCloverLicenseTypeStore.javaCLOVER_ACADEMIC*Lclover/com/
atlassian/license/LicenseType;CLOVER_COMMERCIALCLOVER_COMMUNITYCLOVER_EVALUATIONCL
OVER_OPEN_SOURCECLOVER_DEVELOPERCLOVER_DEMONSTRATIONpublicKeyFileNameLjava/lang/
String;privateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
 ! #
 % ')
+thisILclover/com/atlassian/license/applications/clover/CloverLicenseTypeStore;getAllLicenses()Ljava/util/Collecti
on;getPublicKeyFileName()Ljava/lang/String; 3getPrivateKeyFileName
6<clinit>/clover/com/atlassian/license/DefaultLicenseType9Clover:
Academic;+clover/com/atlassian/extras/api/LicenseType=ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;?@ >AnameC2
>D*(ILjava/lang/String;ZZLjava/lang/String;)VF
:GClover: CommercialI
COMMERCIALK@ >LClover: CommunityN COMMUNITYP@ >QClover: EvaluationSClover: Open
SourceUOPEN_SOURCEW@ >XClover: DeveloperZ DEVELOPER\@ >]Clover: Demonstration_
DEMONSTRATIONa@
>b$clover/com/atlassian/clover/leaf.keydclover/clover.bytefCodeLocalVariableTableLineNumberTable
SourceFile!

 hR** W*" W*$ W*& W*(W** W*, WiR-.j& %0; F!Q"/0h/*i-.j&12h.4i-.j+52h.7i-
.j08h:Y<BEH:YJMEH":YOREH$:YTMEH&:YVYEH(:Y[^EH*:Y$`cEH,e4g7j&
.E\sk
0Oclover/com/atlassian/license/applications/confluence/ConfluenceLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreConfluenceLicenseTypeStore.javaACADEMIC*Lclover/com/atlassia
n/license/LicenseType;
EVALUATIONTESTINGHOSTED_EVALUATION
NON_PROFITFULL_LICENSEPERSONALSTARTERHOSTED COMMUNITYOPEN_SOURCE DEVELOPER
DEMONSTRATIONpublicKeyFileNameLjava/lang/String;-
clover/com/atlassian/confluence/page/Page.keyprivateKeyFileNameconfluence/confluence.byteAPPLICATION_N
AMECONF<init>()V
applicationLicenseTypesLjava/util/ArrayList;"# $ &java/util/ArrayList(add(Ljava/lang/Object;)Z*+
),. 0
 2 4 6
 8 : < > @ B
DthisQLclover/com/atlassian/license/applications/confluence/ConfluenceLicenseTypeStore;getAllLicenses()Ljava/u

til/Collection;getPublicKeyFileName()Ljava/lang/String;getPrivateKeyFileName<clinit>/clover/com/atlassian/licen
se/DefaultLicenseTypeNConfluence: AcademicP+clover/com/atlassian/extras/api/LicenseTypeR-
Lclover/com/atlassian/extras/api/LicenseType;T SUnameWK
SX*(ILjava/lang/String;ZZLjava/lang/String;)VZ
O[Confluence: Evaluation]
COMMERCIAL_T S`Confluence: Testingb T Sd+(ILjava/lang/String;ZZZLjava/lang/String;)Vf
OgConfluence: Hosted EvaluationiT Sk$Confluence: Non-Profit / Open SourcemT SoConfluence: Commercial
ServerqConfluence: Personal Servers
T SuConfluence: StarterwT SyConfluence: Commercial Hosted{Confluence: Community}T SConfluence: Open
SourceT SConfluence: DeveloperT SConfluence: DemonstrationT S
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!

    !%'-W*%/-W*%1-W*%3-W*%5-W*%7-W*%9-W*%;-W*%=-W*%?-W*%A-W*%C-W*%E-WFG>
!"#%\$0%;&F'Q(\)g*r+},-.HI/*%FG2JK-FG7LK-FG<Mo'OYQVY\'OY
^aY\/OY0ceYh1OY@jlY\3OYNnpY\5OYUraY\7OY`tvY\9OY0xzY\COY|lYh;OY~Y\=OY
Y\?OYY\AOY+Y\E6
,
CYo
0Iclover/com/atlassian/license/applications/fisheye/FishEyeLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreFishEyeLicenseTypeStore.javaFISHEYE_ACADEMIC*Lclover/co
m/atlassian/license/LicenseType;FISHEYE_COMMERCIALFISHEYE_COMMUNITYFISHEYE_EVALUATION
FISHEYE_OPEN_SOURCEFISHEYE_DEVELOPERFISHEYE_STARTERFISHEYE_DEMONSTRATIONFISH
EYE_TESTINGpublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
!# %
 ') +
 - /
1thisKLclover/com/atlassian/license/applications/fisheye/FishEyeLicenseTypeStore;getAllLicenses()Ljava/util/Coll
ection;getPublicKeyFileName()Ljava/lang/String; 9getPrivateKeyFileName
<<clinit>/clover/com/atlassian/license/DefaultLicenseType?FishEye:
AcademicA+clover/com/atlassian/extras/api/LicenseTypeCACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;EF DGnameI8
DJ*(ILjava/lang/String;ZZLjava/lang/String;)VL
@MFishEye: CommercialO
COMMERCIALQF DRFishEye: CommunityT COMMUNITYVF DWFishEye: EvaluationYFishEye: Open
Source[OPEN_SOURCE]F D^FishEye: Developer` DEVELOPERbF DcFishEye: StartereSTARTERgF DhFishEye:
Demonstrationj
DEMONSTRATIONlF DmFishEye: TestingoTESTINGqF Dr+(ILjava/lang/String;ZZZLjava/lang/String;)Vt
@u%clover/com/atlassian/fisheye/leaf.keywfisheye/fisheye.byteyCodeLocalVariableTableLineNumberTable
SourceFile!

    {h**"W*\$"W*&"W*("W**"W*,"W*."W*0"W*2"W|h34}.%0
;!F"Q#\\$g%56{/*|34})78{.:|34}.;8{.=|34}3>{@YBHKN@YPSKN\$@YUXKN&@YZSKN(@Y_KN*@YadKN,
@YfiKN.@YknKN0@YpsKv2x:z=}.
.E\s~
0QWclover/com/atlassian/license/applications/editliveplugin/EditlivePluginLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStore#EditlivePluginLicenseTypeStore.javaEDITLIVE_ACADEMIC*Lcl

over/com/atlassian/license/LicenseType;EDITLIVE_EVALUATIONEDITLIVE_NON_PROFITEDITLIVE_FULL
_LICENSEpublicKeyFileNameLjava/lang/String;,clover/com/atlassian/editlive/publickey.byte
privateKeyFileNameeditlive/editlive.byte<init>()V
applicationLicenseTypesLjava/util/ArrayList; java/util/ArrayListadd(Ljava/lang/Object;)Z
  " $
&thisYLclover/com/atlassian/license/applications/editliveplugin/EditlivePluginLicenseTypeStore;getAllLicenses()L
java/util/Collection;getPublicKeyFileName()Ljava/lang/String;getPrivateKeyFileName<clinit>/clover/com/atlassian
/license/DefaultLicenseType0EditLive!: Academic2+clover/com/atlassian/extras/api/LicenseType4ACADEMIC-
Lclover/com/atlassian/extras/api/LicenseType;67 58name:-
5;*(ILjava/lang/String;ZZLjava/lang/String;)V=
1>EditLive!: Evaluation@
COMMERCIALB7 5C#EditLive!: Non-Profit / Open SourceE
NON_PROFITG7 5HEditLive!: CommercialJ
ConstantValueCodeLocalVariableTableLineNumberTable
SourceFile!
LLMo1**!W*#!W*%!W*'!WN1()O%0*+M/*N()O,-M-N()O$.-M-
N()O)/M]1Y39<?1YAD<?#1YFI<?%1YKD<?'O
.EP
0Gclover/com/atlassian/license/applications/bamboo/BambooLicenseTypeStore-
clover/com/atlassian/license/LicenseTypeStoreBambooLicenseTypeStore.javaBAMBOO_BASIC_EVALUATION*
Lclover/com/atlassian/license/LicenseType;BAMBOO_BASIC_ACADEMICBAMBOO_BASIC_DEMONSTRATI
ONBAMBOO_BASIC_DEVELOPERBAMBOO_BASIC_COMMUNITYBAMBOO_BASIC_OPEN_SOURCEB
AMBOO_BASIC_COMMERCIAL_SERVERBAMBOO_EVALUATIONBAMBOO_ACADEMICBAMBOO_DE
MONSTRATIONBAMBOO_DEVELOPERBAMBOO_COMMUNITYBAMBOO_OPEN_SOURCEBAMBOO_C
OMMERCIAL_SERVERBAMBOO_PROFESSIONAL_EVALUATIONBAMBOO_PROFESSIONAL_ACADEM
IC!BAMBOO_PROFESSIONAL_DEMONSTRATIONBAMBOO_PROFESSIONAL_DEVELOPERBAMBOO_P
ROFESSIONAL_COMMUNITYBAMBOO_PROFESSIONAL_OPEN_SOURCE%BAMBOO_PROFESSIONAL
_COMMERCIAL_SERVERBAMBOO_ENTERPRISE_EVALUATIONBAMBOO_ENTERPRISE_ACADEMICB
AMBOO_ENTERPRISE_DEMONSTRATIONBAMBOO_ENTERPRISE_DEVELOPERBAMBOO_ENTERPRIS
E_COMMUNITYBAMBOO_ENTERPRISE_OPEN_SOURCE#BAMBOO_ENTERPRISE_COMMERCIAL_SER
VERBAMBOO_TEMP_2_0_BETApublicKeyFileNameLjava/lang/String;privateKeyFileName<init>()V'(
)applicationLicenseTypesLjava/util/ArrayList;+, - /java/util/ArrayList1add(Ljava/lang/Object;)Z34
25 7 9 ; = ? A C E G
 I K M
 O Q S U W Y [ ] _ a c e g! i" k#
mthisILclover/com/atlassian/license/applications/bamboo/BambooLicenseTypeStore;getAllLicenses()Ljava/util/Col
lection;getPublicKeyFileName()Ljava/lang/String;$% ugetPrivateKeyFileName&%
x<clinit>/clover/com/atlassian/license/DefaultLicenseType{Bamboo Basic:
Evaluation}+clover/com/atlassian/extras/api/LicenseType
COMMERCIAL-Lclover/com/atlassian/extras/api/LicenseType; namet
.clover/com/atlassian/extras/api/LicenseEditionBASIC0Lclover/com/atlassian/extras/api/LicenseEdition;
Z(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V'
|Bamboo Basic: AcademicACADEMIC Bamboo Basic: Demonstration
DEMONSTRATION Bamboo Basic: Developer DEVELOPER Bamboo Basic: Community COMMUNITY Bamboo
Basic: Open SourceOPEN_SOURCE Bamboo Basic: Commercial ServerBamboo Standard: EvaluationSTANDARD
Bamboo Standard: AcademicBamboo Standard: DemonstrationBamboo Standard: DeveloperBamboo Standard:
CommunityBamboo Standard: Open Source"Bamboo Standard: Commercial ServerBamboo Professional:
EvaluationPROFESSIONAL Bamboo Professional: Academic"Bamboo Professional: DemonstrationBamboo

Professional: DeveloperBamboo Professional: Community Bamboo Professional: Open Source&Bamboo Professional: Commercial ServerBamboo Enterprise: Evaluation
ENTERPRISE Bamboo Enterprise: Academic Bamboo Enterprise: DemonstrationBamboo Enterprise: DeveloperBamboo Enterprise: CommunityBamboo Enterprise: Open Source$Bamboo Enterprise: Commercial ServerBamboo: 2.0 betaTESTING
$clover/com/atlassian/bamboo/leaf.keybamboo/bamboo.byteCodeLocalVariableTableLineNumberTable SourceFile!

! " # $%
&%'(D***.06W*.86W*.:6W*.<6W*.>6W*.@6W*.B6W*.D6W*.F6W*.H6W*.J6W*.L6W*.N6W*.P6W*.R6W*.
T6W*.V6W*.X6W*.Z6W*.\6W*.^6W*.`6W*.b6W*.d6W*.f6W*.h6W*.j6W*.l6W*.n6WDop~79:;%<0=;>F?QA\
BgCrD}EFGIJKLMNOQRSTU"V-
W8YCZqr/*.op^st.vopcwt.yophz(|Y~F|YD|YH|YJ|YL|YN|YP|Y8|Y0|Y:|Y<|Y>|Y@|YB|YT|YR|YV|YX|YZ|Y\|Y^|
Yb|Y`|Yd|Yf|Yh|Yj|Yl|Ynvy~
4Nh8Rl !"#$%"(<)V*p+,-.134
0+clover/com/atlassian/license/LicenseManagerjava/lang/ObjectLicenseManager.java1clover/com/atlassian/extras/c
ommon/log/Logger$Log-
clover/com/atlassian/extras/common/log/LoggerLoglog3Lclover/com/atlassian/extras/common/log/Logger$Log;lice
nseListLjava/util/Map;licenseConfigurationslicenseManager-
Lclover/com/atlassian/license/LicenseManager;1class$clover$com$atlassian$license$LicenseManagerLjava/lang/Cl
ass;<init>()V
java/util/HashMap

 thisgetInstance/()Lclover/com/atlassian/license/LicenseManager; "
addLicenseConfigurationr(Ljava/lang/String;Lclover/com/atlassian/license/LicenseTypeStore;Lclover/com/atlassian
/license/LicenseRegistry;)V1clover/com/atlassian/license/LicenseConfiguration`(Lclover/com/atlassian/license/Lice
nseRegistry;Lclover/com/atlassian/license/LicenseTypeStore;)V)
(*
java/util/Map,put8(Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object;./-
0applicationNameLjava/lang/String;licenseTypeStore/Lclover/com/atlassian/license/LicenseTypeStore;licenseRegis
try.Lclover/com/atlassian/license/LicenseRegistry;licenseConfiguration3Lclover/com/atlassian/license/LicenseConfi
guration;getLicenseRegistryB(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseRegistry;getLicenseConfigu
rationG(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseConfiguration;<=
>0()Lclover/com/atlassian/license/LicenseRegistry;:@
(AgetLicenseTypeStoreC(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseTypeStore;1()Lclover/com/atlass
ian/license/LicenseTypeStore;CE
(Fget&(Ljava/lang/Object;)Ljava/lang/Object;HI-Jjava/lang/RuntimeExceptionLjava/lang/StringBufferN
O&No LicenseConfiguration found for key Qappend,(Ljava/lang/String;)Ljava/lang/StringBuffer;ST
OUtoString()Ljava/lang/String;WX
OY(Ljava/lang/String;)V[
M\lookupLicenseTypeStorehasValidLicense(Ljava/lang/String;)Z
getLicense:(Ljava/lang/String;)Lclover/com/atlassian/license/License;ab
c$clover/com/atlassian/license/Licensee isExpired()Zghfi
licenseKey-clover/com/atlassian/license/LicenseExceptionljava/lang/ExceptionnisEmptyph-
qcontainsKey(Ljava/lang/Object;)Zst-u w>There is no License Configuration defined for the application
y.{error(Ljava/lang/Object;)V}~,clover/com/atlassian/license/LicenseRegistrygetLicenseMessageXgetLicenseHash
X?There is no license string or hash defined for the application
info~(clover/com/atlassian/license/LicensePair'(Ljava/lang/String;Ljava/lang/String;)V

Could not build a license

pair*(Ljava/lang/Object;Ljava/lang/Throwable;)V}3clover/com/atlassian/license/decoder/LicenseDecoderd(Lclover /com/atlassian/license/LicensePair;Ljava/lang/String;)Lclover/com/atlassian/license/License;a

Exception getting license: ,(Ljava/lang/Object;)Ljava/lang/StringBuffer;S

Oe/Lclover/com/atlassian/license/LicenseException;license&Lclover/com/atlassian/license/License;

licenseStrhashpair*Lclover/com/atlassian/license/LicensePair;Ljava/lang/Exception;

setLicenseL(Ljava/lang/String;Ljava/lang/String;)Lclover/com/atlassian/license/License;

\isValid?(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)Z

?(Lclover/com/atlassian/license/LicensePair;Ljava/lang/String;)V

qAttempt to set invalid license. Ensure that you are calling setLicense(license, appName) - not (appName, license)warnupdatedLicenseremoveI-()[Ba

)clover/com/atlassian/license/LicenseUtils getString([B)Ljava/lang/String;

setLicenseMessage[getHash

setLicenseHash[getLicensePair>(Ljava/lang/String;)Lclover/com/atlassian/license/LicensePair; Couldn't get the LicensePair ...getLicenseTypeP(Ljava/lang/String;Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;- clover/com/atlassian/license/LicenseTypeStore>(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType; licenseTypeString?(Ljava/lang/String;I)Lclover/com/atlassian/license/LicenseType;- (I)Lclover/com/atlassian/license/LicenseType;

licenseTypeCodeIresetclear-clearLicenseConfigurations

removeLicense<clinit> getClass()Ljava/lang/Class;

java/lang/ClassgetComponentType

F(Ljava/lang/Class;)Lclover/com/atlassian/extras/common/log/Logger$Log;

  SyntheticCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses!

Q**Y*Y !"# !4#

Y$##+-0%&s(Y-,+:*+1W423456789567:;= *+?B 23;CD= *+?G 23@<=y/*+K(M,MYOYPRV+VZ], //23!89EFH- J^Dc*+K(M,,G 2389TUW[_`O*+d*+djk3bab

r+vM*+K(N-%xOYPzV+V|VZ-B:::%xOYPV+V|VZ:Y::x+M*+,1W MxOYPV,Z,*+KfmNoOooof

)89Ut67^k3gb3323bgkl)m-oMpOrUs^tgvqxy|3NY+N-,:-, *-,:x"#o>%3333231& #%1@+>*,W*,K(N- B:++4"89#67@@@23#1?m5*+K(M,BNY--Mx,&'m48967(

5523'(3c*+K(N-G,*233 89

mc*+K(N-G*23 89

mM**# 8

*

 [D*+W23

5YWYx

0X.clover/com/atlassian/license/DefaultSIDManagerjava/lang/Object'clover/com/atlassian/license/SIDManagerDefa ultSIDManager.javaCHARACTER_POOLLjava/lang/String;$ABCDEFGHIJKLMNOPQRSTUVWXYZ01234567 89

 BAD_WORDS[Ljava/lang/String;

KEY_LENGTHICURRENT_VERSION_INITAL_CHARCBPREVIOUS_VERSIONS_INITIAL_CHARSASEPA RATOR_CHAR-randomLjava/security/SecureRandom;<init>()V([B)V

this0Lclover/com/atlassian/license/DefaultSIDManager;

"java/lang/System$currentTimeMillis()J&'

%(java/lang/String*valueOf(J)Ljava/lang/String;,-

+.java/lang/StringBuffer0

1"append,(Ljava/lang/String;)Ljava/lang/StringBuffer;34

15:7identityHashCode(Ljava/lang/Object;)I9:

%;(I)Ljava/lang/StringBuffer;3=

1>toString()Ljava/lang/String;@A

1B

getProperties()Ljava/util/Properties;DE

%Fjava/util/PropertiesH

IBgetBytes()[BKL

+Mjava/security/SecureRandomO

P RseedStrseed[BgenerateSID(C)Ljava/lang/StringBuffer;3X

1Y

nextDouble()D[\

P]length()I_`

+acharAt(I)Ccd

+egetCharacterForCRC([B)Cgh

i

isKeyClean(Ljava/lang/String;)Zkl

mindexi charCountresLjava/lang/StringBuffer;

isValidSIDvalidateStringSyntaxul

v substring(II)Ljava/lang/String;xy

+zkeyStrcrcChar checkCharsidvalidZindexOf(I)I

+java/util/zip/CRC32

"update

getValue'

bytescrc32Ljava/util/zip/CRC32;crcValueJstripDashesAndNumbers&(Ljava/lang/String;)Ljava/lang/String;

 (Ljava/lang/String;)I

+badWordkeycharKey--clover/com/atlassian/license/util/StringUtils

replaceAllJ(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;

123E45S6G7T8B9P0O<clinit>FUCKSHITCOCKDICKCUNTTWATBITCHBASTARDJIZJISMFARTCRAPASS

PORNPISSPUSSYBALLSTITSBOOBSCOOCHCUMCHOADDILDODOUCHECLITMUFFNOBPECKERPRIC

KPOONTANG QUEEFSNATCH

TWOTDYKECOONNIGFAGWANKERGOOKFUDGEPACKERQUEERRAGHEAD!SKANK#SPIC%GOD'DA

MN)FICK+SCHEISSE-SCHWANZ/FOTZE1HURE3

SCHWUCHTEL5SCHWUL7TITTEN9ARSCH;IDIOT=SAU?ASSHATATURDBURGLARCDIRTYSANCHEZE

FELCHGBLASENIWICKSERKFEUCHTMMOESEOMILCHTUETENQ

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile! S

SS SS T4*U !V

TY*#+G)/M1Y2,686,<?CM1Y2,686GJ6CM,NL*PY+QSU =T Y !YUVV"&')**+G-L0X1WATp1Y2L+BZW=>:+-

ZW=#*S^bk6+fZW+*+CNjZW*+Cn+CU4;o=pOqp !hrsV:8;<=@B%C*H;IGJJ=PO`QkStlT3*+w=*+{N*-

Nj6+f6=U>| }$

~3 !3 -VYZ

]_a$c1fulTZ=+=P+b=B+fB+f=%+f-+ f-+f-=U Z !Z XV*

pqsuwy1{6}VXghT%YM,+,B!bq6fU4% !%VoV

klT+*+M>2:,U4 !p+ !+ % V!#)Te+M,M,M,M,M,M,M,M,M,M,M,U e !e \r V2 $-

---

6?HQZcTB+YSYSYSYSYSYSYSYSYSY SY
SYSYSY
SYSYSYSYSYSYSYSYSYSYSYSYSYSYSYSY
SYSYSY SY!SY"SY#SY$SY%SY&SY'SY(SY)"SY*$SY+&SY,(SY-
*SY.,SY/.SY00SY12SY24SY36SY48SY5:SY6<SY7>SY8@SY9BSY:DSY;FSY<HSY=JSY>LSY?NSY@PSYA
RSVW
0-
clover/com/atlassian/license/LicenseExceptionjava/lang/ExceptionLicenseException.java<init>(Ljava/lang/String;)
V
this/Lclover/com/atlassian/license/LicenseException;messageLjava/lang/String;(Ljava/lang/Throwable;)V
causeLjava/lang/Throwable;*(Ljava/lang/String;Ljava/lang/Throwable;)V
CodeLocalVariableTableLineNumberTable
SourceFile!>*+

>*+

I*+,
0$clover/com/atlassian/license/Licensejava/lang/ObjectLicense.javagetDateCreated()Ljava/util/Date;getDatePurchas
edgetOrganisation()Ljava/lang/String;getLicenseType,()Lclover/com/atlassian/license/LicenseType; isExpired()Z
getExpiryDatetoStringisLicenseLevel(Ljava/util/Collection;)ZgetUsers()IgetPartnerNamegetLicenseIdgetPermitted
ClusteredNodesgetLicenseDuration()JgetSupportEntitlementNumber
SourceFile
0Z4clover/com/atlassian/license/AbstractLicenseRegistryjava/lang/Object,clover/com/atlassian/license/LicenseRegi
stryAbstractLicenseRegistry.java<init>()V

this6Lclover/com/atlassian/license/AbstractLicenseRegistry;getAllLicenseTypes()Ljava/util/Collection;getLicenseT
ype>(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;-clover/com/atlassian/license/LicenseException
java/util/Collectioniterator()Ljava/util/Iterator;java/util/IteratorhasNext()Z
next()Ljava/lang/Object;"#$(clover/com/atlassian/license/LicenseType&getDescription()Ljava/lang/String;()'*java/l
ang/String,toLowerCase.)
-/indexOf(Ljava/lang/String;)I12
-3java/lang/StringBuffer5
6
The license type (8append,(Ljava/lang/String;)Ljava/lang/StringBuffer;:;
6<) specified is invalid.>toString@)
6A(Ljava/lang/String;)VC
DlicenseType*Lclover/com/atlassian/license/LicenseType;licenseTypeDescLjava/lang/String;Ljava/util/Iterator;typ
e-(I)Lclover/com/atlassian/license/LicenseType;getType()IMN'O(I)Ljava/lang/StringBuffer;:Q
6RICodeLocalVariableTableLineNumberTable
Exceptions
SourceFile! U/*V
W
UZ*M,!*,%'N-+0:+04-Y6Y79=+=?=BEV4FG(HI
0JZ
ZKIW(57:XLUL*M,!,%'N-P-Y6Y79=S?=BEV*FG
"JL
LKTW#%"()),*XY

0L/clover/com/atlassian/license/DefaultLicenseTypejava/lang/Object(clover/com/atlassian/license/LicenseTypeDefa
ultLicenseType.javatypeIdescriptionLjava/lang/String;isEvaluationZrequiresUserLimitexpiresnewLicenseTypeNam
eedition0Lclover/com/atlassian/extras/api/LicenseEdition;<init>*(ILjava/lang/String;ZZLjava/lang/String;)V[(ILjav
a/lang/String;ZZZLjava/lang/String;Lclover/com/atlassian/extras/api/LicenseEdition;)V
this1Lclover/com/atlassian/license/DefaultLicenseType;Z(ILjava/lang/String;ZZLjava/lang/String;Lclover/com/atla
ssian/extras/api/LicenseEdition;)V+(ILjava/lang/String;ZZZLjava/lang/String;)V()V

!
#
%
') +
getEdition2()Lclover/com/atlassian/extras/api/LicenseEdition;getNewLicenseTypeName()Ljava/lang/String;hashCo
de()Ijava/lang/String312
45getTypetoStringgetDescription90
:isEvaluationLicenseType()ZgetNiceNameequals(Ljava/lang/Object;)Z72
AAlicense*Lclover/com/atlassian/license/LicenseType;oLjava/lang/Object;CodeLocalVariableTableLineNumberTa
ble
SourceFile!

Hn*,I>

J

Hy*,IH

J
Hy*,IH

J
!H ,** *,"*$*&*(***,IR,, ,
,
,
,
,,J& $% &'()*%++,-.H/*,IJ0/0H/**IJ512H7
* *"6`I
J:72H/* IJ?80H/*;IJD90H/*"IJI<=H/*$IJN=H/*(IJS=H/*&IJX>0H/*"IJ`?@Hk++M*B,CI DEFGJeghjmK
0)clover/com/atlassian/license/LicenseUtilsjava/lang/ObjectLicenseUtils.java1clover/com/atlassian/extras/common/l
og/Logger$Log-
clover/com/atlassian/extras/common/log/LoggerLoglog3Lclover/com/atlassian/extras/common/log/Logger$Log;PO
ST_LICENSE_EVAL_PERIODJ~UPDATE_ALLOWED_PERIOD\ALMOST_EXPIRED_PERIODKPARTNER_
NOT_MATCHING_BUILDLjava/lang/String;'partner not matching build partner
nameLICENSE_NO_PARTNER'License does not contain a partner

name/class$clover$com$atlassian$license$LicenseUtilsLjava/lang/Class;<init>()V !

"this+Lclover/com/atlassian/license/LicenseUtils; getString([B)Ljava/lang/String;rndChar(I)C()

*getCharInRange,)

- /java/lang/StringBuffer1

2"Invalid Char in stream 4append,(Ljava/lang/String;)Ljava/lang/StringBuffer;67

28(I)Ljava/lang/StringBuffer;6:

2;toString()Ljava/lang/String;=>

2?debug(Ljava/lang/Object;)VABCjava/lang/StringE([C)V G

FHaByteBiI byteArray[BcharByte[Cstrjava/lang/MathSrandom()DUV

TW@@cuZInvalid int in stream `c1getByteInRange(C)BIncorrect character in stream

e(C)Ljava/lang/StringBuffer;6g

2hCgetBytes(Ljava/lang/String;)[BtoCharArray()[Cmn

Fojava/lang/CharacterqtoLowerCase(C)Cst

rucd

w!Invalid character in byte stream ystring

charArraybytesgetSupportPeriodEnd)(Lclover/com/atlassian/license/License;)J$clover/com/atlassian/license/Licens

egetDateCreated()Ljava/util/Date;java/util/DategetTime()J

license&Lclover/com/atlassian/license/License;isLicenseTooOldForBuild9(Lclover/com/atlassian/license/License;L

java/util/Date;)Z~

 buildDateLjava/util/Date;confirmExtendLicenseExpired(Ljava/util/Date;)Z

"#getNewBuildWithOldLicenseExpiryDate(Ljava/util/Date;)J

dateConfirmed(Ljava/lang/String;)Zjava/lang/NumberFormatExceptionjava/lang/Long

parseLong(Ljava/lang/String;)J

(J)V

!getSupportPeriodAlmostExpiredDateisPartnerDetailsValidL(Lclover/com/atlassian/license/License;Ljava/lang/Stri

ng;)Ljava/lang/String;getPartnerName>equals(Ljava/lang/Object;)Z

FbuildPartnerNamelicensePartnerNamereadKey(Ljava/io/InputStream;)[Bjava/io/IOExceptionjava/io/ByteArrayOut

putStream

"java/io/InputStreamread([B)I

write([BII)V

toByteArray()[B

isLjava/io/InputStream;boutLjava/io/ByteArrayOutputStream;len<clinit> getClass()Ljava/lang/Class;

java/lang/ClassgetComponentType

getInstanceF(Ljava/lang/Class;)Lclover/com/atlassian/extras/common/log/Logger$Log;

ConstantValue SyntheticCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses!

!3*#$%

 &'a*hL=**3>!++U+*``.U++U+*`@`.Ua@++U+*`.UB@!++U+*`@d.U02Y359<@DPFY+IM,4JK

LMNOPQRN#$&')&*7,A.H/X1c3j4w689=$@A

()u/hXYk`<X[k=aA` /LM!]M ^_FG H

,)j 0`

#

dA`$=$da`><?>02Y3a9<@DjbM2PRTV&X2Z;\A^D`JbMehf

cdk09 0dAZAd

`azad$`<>>?02Y3f9i@Dkbj2nprt'v3x<zB|E~KNi

kl**pL+lM>,+4vg,+,`4xdTv+4vm,+,`4x@dTX+4vs,+,`4xT>+4vy,+,`4x@`T 02Y3z9+4i@De,*LM{|Q}O:3>Q\kv

~8*a F*+ BY* 9Y*

3 *a 9Y* 3 *e ?*M,,,+,,++ ??89< *YL=N*-Y=

+-+**"

M}O

%!5YWY0

0?-clover/com/atlassian/license/util/StringUtilsjava/lang/ObjectStringUtils.java<init>()V

this/Lclover/com/atlassian/license/util/StringUtils;

replaceAllJ(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;java/lang/Stringequals(Ljava/lang/Object;)Z

java/lang/StringBufferlength()I

(I)V

indexOf(Ljava/lang/String;)I

! substring(II)Ljava/lang/String;#$

%append,(Ljava/lang/String;)Ljava/lang/StringBuffer;'(

)(I)Ljava/lang/String;#+

,toString()Ljava/lang/String;./

0iIstrLjava/lang/String;

oldPattern

newPattern remainderbufLjava/lang/StringBuffer;CodeLocalVariableTableLineNumberTable

SourceFile!;/* <

=

;i*++**NY*h:-+"6&-&*W,*W-+`-N

-*W1<>-323i45i65i75R85&C9:=6

&-3@GV]` c!>

0(clover/com/atlassian/license/LicenseTypejava/lang/ObjectLicenseType.javaequals(Ljava/lang/Object;)ZhashCode

()IgetTypetoString()Ljava/lang/String;getDescriptiongetNiceNameisEvaluationLicenseType()ZrequiresUserLimitexpiresgetNewLicenseTypeName

getEdition2()Lclover/com/atlassian/extras/api/LicenseEdition;

SourceFile

0m-

clover/com/atlassian/license/LicenseTypeStorejava/lang/ObjectLicenseTypeStore.javaapplicationLicenseTypesLjava/util/ArrayList;<init>()V

java/util/ArrayList

this/Lclover/com/atlassian/license/LicenseTypeStore;getPublicKeyFileName()Ljava/lang/String;getPrivateKeyFileNamegetLicenseType>(Ljava/lang/String;)Lclover/com/atlassian/license/LicenseType;-

clover/com/atlassian/license/LicenseExceptionjava/lang/Stringequals(Ljava/lang/Object;)Z

java/lang/StringBuffer"

#

1License description must be specified; you used [%append,(Ljava/lang/String;)Ljava/lang/StringBuffer;'(

#)]+toString-

#.(Ljava/lang/String;)V0

1iterator()Ljava/util/Iterator;34

5java/util/Iterator7hasNext()Z9:8;next()Ljava/lang/Object;=>8?(clover/com/atlassian/license/LicenseTypeAgetDesc
riptionCBD:License type added with an invalid description; you used [FtoLowerCaseH

IindexOf(Ljava/lang/String;)IKL

M(LicenseType not found with description

[OlicenseType*Lclover/com/atlassian/license/LicenseType;licenseTypeDescLjava/lang/String;Ljava/util/Iterator;lic
enseTypeString-(I)Lclover/com/atlassian/license/LicenseType;getType()IXYBZThe license type

(\(I)Ljava/lang/StringBuffer;'^

#_) specified is

invalid.alicenseCodeIlookupLicenseTypegetAllLicenses()Ljava/util/Collection;CodeLocalVariableTableLineNumbe
rTable

Exceptions

SourceFile! h>**

Yij

 h;++!#Y#Y$&*+*,*/2*6M,<f,@BN-E-E!(Y#Y$G*-E*,*/2-EJ:+JN-Y#Y$P*+*,*/2i4HVQRST5l3UVTj.
->H_!$&(*kWhJ*6M,<,@BN-[-Y#Y$]*`b*/2i*QR"3UJJcdj/13%4'5*6keWh,*6M,<,@BN-[-
i*QR"3U,,cdj@BD%E'F*Gfgh/*ijNl

0

,clover/com/atlassian/license/LicenseRegistryjava/lang/ObjectLicenseRegistry.javasetLicenseMessage(Ljava/lang/St
ring;)VsetLicenseHashgetLicenseMessage()Ljava/lang/String;getLicenseHash

SourceFile

0!1clover/com/atlassian/license/LicenseConfigurationjava/lang/ObjectLicenseConfiguration.javalicenseRegistry.Lcl
over/com/atlassian/license/LicenseRegistry;licenseTypeStore/Lclover/com/atlassian/license/LicenseTypeStore;<init
>`(Lclover/com/atlassian/license/LicenseRegistry;Lclover/com/atlassian/license/LicenseTypeStore;)V()V

this3Lclover/com/atlassian/license/LicenseConfiguration;getLicenseRegistry0()Lclover/com/atlassian/license/Licen
seRegistry;getLicenseTypeStore1()Lclover/com/atlassian/license/LicenseTypeStore;setLicenseRegistry1(Lclover/co
m/atlassian/license/LicenseRegistry;)VsetLicenseTypeStore2(Lclover/com/atlassian/license/LicenseTypeStore;)VC
odeLocalVariableTableLineNumberTable

SourceFile!

Y**+*, /*/*>*+

!>*+

%&

0+clover/com/atlassian/license/DefaultLicensejava/lang/Object$clover/com/atlassian/license/LicenseDefaultLicense
.javaEVALUATION_PERIODJdateCreatedLjava/util/Date;

datePurchaseddateExpiredorganisationLjava/lang/String;licenseType*Lclover/com/atlassian/license/LicenseType;us
ersIpartnerName

licenseIdpermittedClusteredNodesdurationsen<init>(Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/
atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;)V(Ljava/util/Date;Ljava/util/Date;Ljava/lang/St
ring;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;ILjava/lang/String;)V

this-

Lclover/com/atlassian/license/DefaultLicense;(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lcl
over/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;)V(Ljava/util/Date;Ljava/util/Date;Lja
va/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;Ljava/lang/String;ILja
va/lang/String;)V#

$expiresorganisationName(Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/License Type;ILjava/lang/String;Ljava/lang/String;I)V(

)

+(Ljava/util/Date;Ljava/util/Date;JLjava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String; Ljava/lang/String;ILjava/lang/String;)V .

0r(Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String;) V(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/LicenseType;ILja va/lang/String;)V(Ljava/util/Date;Ljava/util/Date;Ljava/util/Date;Ljava/lang/String;Lclover/com/atlassian/license/Li censeType;ILjava/lang/String;Ljava/lang/String;I)V4

5()V7

8

< > @ B D F H

J(Ljava/util/Date;Ljava/util/Date;JLjava/lang/String;Lclover/com/atlassian/license/LicenseType;ILjava/lang/String; Ljava/lang/String;I)VgetDateCreated()Ljava/util/Date;getDatePurchasedgetOrganisation()Ljava/lang/String;getLice nseType,()Lclover/com/atlassian/license/LicenseType;toStringjava/lang/StringBufferU

V8(clover/com/atlassian/license/LicenseTypeXgetNiceNameZQY[append,(Ljava/lang/String;)Ljava/lang/StringBuff er;]^

V_

licensed to aTQ

Vc isExpired()Z

getExpiryDategN

hjava/util/DatejgetTime()Jlm

knjava/lang/SystempcurrentTimeMillisrm

qsexpiryisEvaluationLicenseTypevfYw y(J)V{

k|getPartnerNameisLicenseLevel(Ljava/util/Collection;)Zjava/util/Collectioniterator()Ljava/util/Iterator;java/util/Iter atorhasNextfnext()Ljava/lang/Object;java/lang/StringRS

getDescriptionQYtoLowerCaseQ

indexOf(Ljava/lang/String;)I

levelLjava/util/Iterator;levelsLjava/util/Collection;getUsers()IrequiresUserLimitfYgetLicenseIdgetPermittedClustere dNodesgetLicenseDurationgetSupportEntitlementNumber<clinit>$CodeLocalVariableTableLineNumberTable SourceFile!

 *+,-R !

"
*+,-%\ !
&'
$% *+,-** ,f
!
)*+-

*+,
*!/p !

/01#
*+,
*-1p !

5672{ *+,-*H !

?@3
 *+,-6R !
&'
DE(>*9*:/*,*+=*,?*-A*C*E*G*I*K\ > !>
>>>>>>>2HIJKL%M+N1O7P=QL *+,
**!/f
!

UVW4
*+, **-1f
!

 [\]MN/*= !aON/*? !fPQ/*A !kRS/*C !pTQM#VYW*C\`b`*A`d# !uef*iL++ot !uz{ |~gNn**1L+"*CxkY*?oza}L+*
!%u (~Q/*G !8+M,),N*-*/8 !8136E*C*E !Q/*I !/*K !m/*/ !Q/*, !7z
0)8clover/com/atlassian/extras/core/jira/DefaultJiraLicense6clover/com/atlassian/extras/core/DefaultProductLicense
0clover/com/atlassian/extras/api/jira/JiraLicenseDefaultJiraLicense.javalicenseEdition0Lclover/com/atlassian/extras/
api/LicenseEdition;<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/Lice
nseProperties;)V

LicenseEdition9clover/com/atlassian/extras/common/util/LicensePropertiesgetProperty&(Ljava/lang/String;)Ljava/l
ang/String;@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolvergetLicenseEditionD(Ljava/lang/
String;)Lclover/com/atlassian/extras/api/LicenseEdition;
this:Lclover/com/atlassian/extras/core/jira/DefaultJiraLicense;product)Lclover/com/atlassian/extras/api/Product;lice
nseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;2()Lclover/com/atlassian/extras/api/Lice
nseEdition;CodeLocalVariableTableLineNumberTable
SourceFile
%*+,
,& !"#'$%/&'(
0!Eclover/com/atlassian/extras/core/DefaultProductLicense$DefaultContactjava/lang/Object'clover/com/atlassian/ex
tras/api/ContactDefaultProductLicense.java6clover/com/atlassian/extras/core/DefaultProductLicenseDefaultContact
nameLjava/lang/String;email<init>'(Ljava/lang/String;Ljava/lang/String;)V()V

thisGLclover/com/atlassian/extras/core/DefaultProductLicense$DefaultContact;getName()Ljava/lang/String;getEma
ilCodeLocalVariableTableLineNumberTable
SourceFileInnerClasses0
Y**+*,
)* +,/*0/*5
0)Fclover/com/atlassian/extras/core/greenhopper/DefaultGreenHopperLicense=clover/com/atlassian/extras/core/plu
gins/DefaultPluginLicense>clover/com/atlassian/extras/api/greenhopper/GreenHopperLicenseDefaultGreenHopperL
icense.javalicenseEdition0Lclover/com/atlassian/extras/api/LicenseEdition;<init>g(Lclover/com/atlassian/extras/api
/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

LicenseEdition9clover/com/atlassian/extras/common/util/LicensePropertiesgetProperty&(Ljava/lang/String;)Ljava/l
ang/String;@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolvergetLicenseEditionD(Ljava/lang/

String;)Lclover/com/atlassian/extras/api/LicenseEdition;

thisHLclover/com/atlassian/extras/core/greenhopper/DefaultGreenHopperLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;2()Lclover/com/atlassian/extras/api/LicenseEdition;CodeLocalVariableTableLineNumberTable

SourceFile

%*+,

,& !"#'$%/&'(

0Jclover/com/atlassian/extras/core/DefaultProductLicense$DefaultOrganisationjava/lang/Object,clover/com/atlassian/extras/api/OrganisationDefaultProductLicense.java6clover/com/atlassian/extras/core/DefaultProductLicenseDefaultOrganisationnameLjava/lang/String;<init>(Ljava/lang/String;)V()V

thisLLclover/com/atlassian/extras/core/DefaultProductLicense$DefaultOrganisation;getName()Ljava/lang/String;CodeLocalVariableTableLineNumberTable

SourceFileInnerClasses0

F

**+

 /*

06clover/com/atlassian/extras/core/DefaultProductLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicenseDefaultProductLicense.javaEclover/com/atlassian/extras/core/DefaultProductLicense$DefaultContactDefaultContactJclover/com/atlassian/extras/core/DefaultProductLicense$DefaultOrganisationDefaultOrganisationEclover/com/atlassian/extras/core/DefaultProductLicense$DefaultPartnerDefaultPartnerMILLIS_IN_A_DAYJ&\licenseVersionIdescriptionLjava/lang/String;product)Lclover/com/atlassian/extras/api/Product;serverIdpartner)Lclover/com/atlassian/extras/api/Partner;organisation.Lclover/com/atlassian/extras/api/Organisation;contactsLjava/util/Collection;ALjava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;creationDateLjava/util/Date;purchaseDatemaximumNumberOfUsers

expiryDategracePeriodEndDatemaintenanceExpiryDatesupportEntitlementNumber

evaluationZsubscriptionlicenseType-Lclover/com/atlassian/extras/api/LicenseType;

properties;Lclover/com/atlassian/extras/common/util/LicenseProperties;<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V()V24

5java/lang/String8valueOf(I)Ljava/lang/String;;;

9<9clover/com/atlassian/extras/common/util/LicenseProperties>getProperty8(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;@A?Bjava/lang/IntegerD'(Ljava/lang/String;)Ljava/lang/Integer;::F

EGintValue()IIJ

EK MDescriptionO&(Ljava/lang/String;)Ljava/lang/String;@Q?R T V

EvaluationXjava/lang/BooleanZ'(Ljava/lang/String;)Ljava/lang/Boolean;:\

[]booleanValue()Z_`

[a+, cSubscriptione-, gServerIDi k

getPartnerf(Lclover/com/atlassian/extras/common/util/LicenseProperties;)Lclover/com/atlassian/extras/api/Partner;

mn

o qOrganisations(Ljava/lang/String;)V2u

v xgetContactsS(Lclover/com/atlassian/extras/common/util/LicenseProperties;)Ljava/util/Collection;z{

| !

~CreationDate=clover/com/atlassian/extras/common/LicensePropertiesConstantsDEFAULT_CREATION_DATE$

getDate4(Ljava/lang/String;Ljava/util/Date;)Ljava/util/Date;?#$ PurchaseDate%$

LicenseExpiryDateDEFAULT_EXPIRY_DATE$ '$

getGracePeriodEndDate](Lclover/com/atlassian/extras/common/util/LicenseProperties;Ljava/util/Date;)Ljava/util/Date;

($ MaintenanceExpiryDate)$ SEN*

NumberOfUsersgetInt(Ljava/lang/String;I)I?&

LicenseTypeName@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolvergetLicenseTypeA(Ljava/lang/String;)Lclover/com/atlassian/extras/api/LicenseType;

./ 01 this8Lclover/com/atlassian/extras/core/DefaultProductLicense;GracePeriodjava/util/DategetTime()J(J)V2

gracePeriod

getProduct+()Lclover/com/atlassian/extras/api/Product;getServerId()Ljava/lang/String;+()Lclover/com/atlassian/extras/api/Partner;getOrganisation0()Lclover/com/atlassian/extras/api/Organisation;()Ljava/util/Collection;getCreationDate()Ljava/util/Date;getPurchaseDate

getExpiryDategetNumberOfDaysBeforeExpirygetDaysBeforeDate(Ljava/util/Date;)I

 isExpired

5 compareTo

&getNumberOfDaysBeforeGracePeriodExpiryisWithinGracePeriod`

isGracePeriodExpired`

getMaintenanceExpiryDate&getNumberOfDaysBeforeMaintenanceExpiryisMaintenanceExpiredgetSupportEntitlementNumbergetMaximumNumberOfUsersisUnlimitedNumberOfUsersisEvaluationisSubscriptionnamejava/lang/SystemcurrentTimeMillis

datePartnerName

vpartnerNameContactEMailContactName'(Ljava/lang/String;Ljava/lang/String;)V2

 java/util/Collections

singletonList$(Ljava/lang/Object;)Ljava/util/List;

EMPTY_LISTLjava/util/List;

contactEmailcontactNamegetLicenseVersiongetDescription/()Lclover/com/atlassian/extras/api/LicenseType;

ConstantValue

SignatureCodeLocalVariableTableLineNumberTableC()Ljava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;~(Lclover/com/atlassian/extras/common/util/LicenseProperties;)Ljava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;

SourceFileInnerClasses! !"#$%$&'$($)$*+,-

,./0123*6*,7=CHLN*,PSU*+W*,YS^bd*,fS^bh*,jSl*,pr*Y,tSwy*,}*,*,**,**,**,*,S*,*,S*,

01RBCD&E+F=GOH[IcJvK~LMOPRSTUVWw#,+>Y,ia*##01#'$[]_`/*We/*ljm/*ro/*ytz/*y9Y*~9Y*D*Y*JE***

`H**YD*Y*JE***

`>**`H**YD*Y*JE***

`H**Y/*J/*`8*`/*d`/*h@Q?*+SB+em$

mnR*SL+Y+01

z{y+*SL*SM+, Y,+ +01 "

 'J/*N;

/*U@/*E

0&Dclover/com/atlassian/extras/core/confluence/DefaultConfluenceLicense6clover/com/atlassian/extras/core/DefaultProductLicense<clover/com/atlassian/extras/api/confluence/ConfluenceLicenseDefaultConfluenceLicense.javamaximumNumberClusterNodesI<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

NumberOfClusterNodes9clover/com/atlassian/extras/common/util/LicensePropertiesgetInt(Ljava/lang/String;I)I

thisFLclover/com/atlassian/extras/core/confluence/DefaultConfluenceLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;

maxClustNodesgetMaximumNumberOfClusterNodes()ICodeLocalVariableTableLineNumberTable

SourceFile

"t *+,

,>*#* $!"/*#$%

0Eclover/com/atlassian/extras/core/DefaultProductLicense$DefaultPartnerjava/lang/Object'clover/com/atlassian/extras/api/PartnerDefaultProductLicense.java6clover/com/atlassian/extras/core/DefaultProductLicenseDefaultPartnernameLjava/lang/String;<init>(Ljava/lang/String;)V()V

thisGLclover/com/atlassian/extras/core/DefaultProductLicense$DefaultPartner;getName()Ljava/lang/String;CodeLocalVariableTableLineNumberTable

SourceFileInnerClasses0

F

**+

 /*

0<clover/com/atlassian/extras/core/bamboo/DefaultBambooLicense6clover/com/atlassian/extras/core/DefaultProductLicense4clover/com/atlassian/extras/api/bamboo/BambooLicenseDefaultBambooLicense.javaMAX_REMOTE_AGENTS_NONEIMAX_REMOTE_AGENTS_STANDARDMAX_REMOTE_AGENTS_PROFESSIONAL
MAX_REMOTE_AGENTS_ENTERPRISEMAX_REMOTE_AGENTS_UNLIMITEDdMAX_LOCAL_AGENTS_BASICMAX_LOCAL_AGENTS_UNLIMITEDMAX_PLANS_STARTERMAX_PLANS_UNLIMITEDmaximumNumberOfRemoteAgentsmaximumNumberOfLocalAgentsmaximumNumberOfPlanslicenseEdition0Lclover/com/atlassian/extras/api/LicenseEdition;<init>g(Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/common/util/LicenseProperties;)V

LicenseEdition!9clover/com/atlassian/extras/common/util/LicenseProperties#getProperty&(Ljava/lang/String;)Ljava/lang/String;%&$'@clover/com/atlassian/extras/common/LicenseTypeAndEditionResolver)getLicenseEditionD(Ljava/lang/String;)Lclover/com/atlassian/extras/api/LicenseEdition;+,

*- /calculateRemoteAgents>(Lclover/com/atlassian/extras/common/util/LicenseProperties;)I12

3 5calculateLocalAgents72

8 :calculatePlans<2

=

?this>Lclover/com/atlassian/extras/core/bamboo/DefaultBambooLicense;product)Lclover/com/atlassian/extras/api/Product;licenseProperties;Lclover/com/atlassian/extras/common/util/LicenseProperties;2()Lclover/com/atlassian/extras/api/LicenseEdition;getMaximumNumberOfRemoteAgents()IgetMaximumNumberOfLocalAgentsgetMaximumNumberOfPlansisUnlimitedRemoteAgents()ZisUnlimitedLocalAgentsisUnlimitedPlansjava/lang/NumberFormatExceptionPNumberOfBambooRemoteAgentsRjava/lang/StringTlengthVI

UWjava/lang/IntegerYparseInt(Ljava/lang/String;)I[\

Z]+clover/com/atlassian/extras/api/LicenseType_STARTER-Lclover/com/atlassian/extras/api/LicenseType;ab

`cgetLicenseType/()Lclover/com/atlassian/extras/api/LicenseType;ef

gequals(Ljava/lang/Object;)Zij

`k.clover/com/atlassian/extras/api/LicenseEditionmSTANDARDo np

nkPROFESSIONALs nt

ENTERPRISEv nw UNLIMITEDy

nze!Ljava/lang/NumberFormatException;maxRemoteStringLjava/lang/String;NumberOfBambooLocalAgentsBASIC nmaxLocalStringNumberOfBambooPlans

maxPlanString

ConstantValueCodeLocalVariableTableLineNumberTable

SourceFile

1*+, *,"(.0**,46**,9;**,>@

1AB1CD1EF$%'(')0*+G/*0AB.HI/*6AB3JI/*;AB8KI/*@AB=LM8*6ABBNM8*;ABGOM8*@ABL12j+S(M,,X ,^Nd*hlq*0ru*0r

x*0r{*0rdQ*|}jABjEF a~>Q RVX^`)b6d8fEhHjUlXnepht72:+(M,,X ,^Nd*hl*0rQ*|}:AB:EF 1& z {')68<2,+(M,,X ,^Nd*hl

Q*|},AB,EF # '*

0_8clover/com/atlassian/extras/core/DefaultAtlassianLicensejava/lang/Object0clover/com/atlassian/extras/api/AtlassianLicenseDefaultAtlassianLicense.javaproductLicenseMapLjava/util/Map;jLjava/util/Map<Lclover/com/atlassian/extras/api/Product;Lclover/com/atlassian/extras/api/ProductLicense;>;<init>(Ljava/util/Collection;)V()V

java/util/HashMapjava/util/Collectionsize()I(I)V

iterator()Ljava/util/Iterator;java/util/Iterator!hasNext()Z#$"%next()Ljava/lang/Object;'(").clover/com/atlassian/extras/api/ProductLicense+

getProduct+()Lclover/com/atlassian/extras/api/Product;-.,/

java/util/Map1put8(Ljava/lang/Object;Ljava/lang/Object;)Ljava/lang/Object;3425license0Lclover/com/atlassian/extras/api/ProductLicense;i$Ljava/util/Iterator;this:Lclover/com/atlassian/extras/core/DefaultAtlassianLicense;productLicensesHLjava/util/Collection<Lclover/com/atlassian/extras/api/ProductLicense;>;Ljava/util/Collection;getProducLicenses()Ljava/util/Collection;getProductLicensesBA

CvaluesEA2Fjava/util/CollectionsHunmodifiableCollection.(Ljava/util/Collection;)Ljava/util/Collection;JK

ILgetProductLicense[(Lclover/com/atlassian/extras/api/Product;)Lclover/com/atlassian/extras/api/ProductLicense;get&(Ljava/lang/Object;)Ljava/lang/Object;PQ2Rproduct)Lclover/com/atlassian/extras/api/Product;

SignatureCodeLocalVariableTableLocalVariableTypeTableLineNumberTableK(Ljava/util/Collection<Lclover/com/atlassian/extras/api/ProductLicense;>;)V

DeprecatedJ()Ljava/util/Collection<Lclover/com/atlassian/extras/api/ProductLicense;>;

SourceFile V

WD**Y++ M,&!,*,N*-0-6WX*/78'9:D;<D=?YD=>Z/CV[@AW/*DX;<Z"\V]BAW7

*GMX

;<Z+V]NOWB*+S,X;<TUZ0^

04clover/com/atlassian/extras/api/bamboo/BambooLicensejava/lang/Object.clover/com/atlassian/extras/api/ProductLicense3clover/com/atlassian/extras/api/LicenseEditionAwareBambooLicense.javagetMaximumNumberOfRemoteAgents()IgetMaximumNumberOfLocalAgentsgetMaximumNumberOfPlansisUnlimitedRemoteAgents()ZisUnlimitedLocalAgentsisUnlimitedPlans

SourceFile

0-

.clover/com/atlassian/extras/api/ProductLicensejava/lang/ObjectProductLicense.javagetLicenseVersion()IgetDescription()Ljava/lang/String;

getProduct+()Lclover/com/atlassian/extras/api/Product;getServerId

getPartner+()Lclover/com/atlassian/extras/api/Partner;getOrganisation0()Lclover/com/atlassian/extras/api/Organisation;getContacts()Ljava/util/Collection;getCreationDate()Ljava/util/Date;getPurchaseDate

getExpiryDategetNumberOfDaysBeforeExpiry

isExpired()ZgetGracePeriodEndDate&getNumberOfDaysBeforeGracePeriodExpiryisWithinGracePeriodisGracePeriodExpiredgetSupportEntitlementNumbergetMaintenanceExpiryDate&getNumberOfDaysBeforeMaintenanceExpiryisMaintenanceExpiredgetMaximumNumberOfUsersisUnlimitedNumberOfUsersisEvaluationisSubscriptiongetLicenseType/()Lclover/com/atlassian/extras/api/LicenseType;getProperty&(Ljava/lang/String;)Ljava/lang/String;

SignatureC()Ljava/util/Collection<Lclover/com/atlassian/extras/api/Contact;>;

SourceFile

*+ !"#$%&'(),

0com/cenqua/clover/CloverLicensejava/lang/ObjectCloverLicense.java1clover/com/atlassian/extras/common/log/Logger$Log-clover/com/atlassian/extras/common/log/LoggerLog

GENERIC_ERRORLjava/lang/String;Invalid license data

CLOVER_EDITION_PROPERTYclover.license.editionCLOVER_EDITION_PROPERTY_DESKTOPdesktopONE_DAYJ&\ PERMS_ALLPERMS_HIST_PDF@PERMS_HIST_HTML

PERMS_TEST_OPTPERMS_CURR_JSONPERMS_CURR_PDFPERMS_CURR_HTMLPERMS_CURR_XML

PERMS_HIST`

PERMS_CURR

PERMS_DESKTOPproductNamelicenseName

softExpiry

hardExpiryorganisationNameownerStatementpreExpiryStatementpostExpiryStatementcontactInfoStatementterminationStatementsupportEntitlementNumberZallowedPkgPrefixesLjava/util/HashSet;maintExpirysupportedFeaturesreadFrom)(Ljava/io/InputStream;)Ljava/lang/String;(com/atlassian/clover/api/CloverExceptionLjava/io/IOExceptionNjava/lang/StringBufferP<init>()VRS

QTjava/io/LineNumberReaderVjava/io/InputStreamReaderXUTF-8Z*(Ljava/io/InputStream;Ljava/lang/String;)VR\Y](Ljava/io/Reader;)VR_

W`readLine()Ljava/lang/String;bc

Wdappend,(Ljava/lang/String;)Ljava/lang/String;StringBuffer;fg

Qh

jtoStringlc

QmError reading license. o

getMessageqc

Or*(Ljava/lang/String;Ljava/lang/Throwable;)VRt

MulinelicenseCertLjava/lang/StringBuffer;linLjava/io/LineNumberReader;eLjava/io/IOException;

licenseInLjava/io/InputStream;(Ljava/lang/String;)Vjava/lang/NullPointerException

TconfigureLoggingForExtrasS

Product: Cloverjava/lang/StringindexOf(Ljava/lang/String;)I

Certificate: FCenqua licenses are no longer compatible with this version of Clover:

#com_cenqua_clover/CloverVersionInfoformatVersionInfoc

.

Please visit %http://www.atlassian.com/clover/renew to obtain a new clover.license.R

M6clover/com/atlassian/extras/core/LicenseManagerFactorygetLicenseManager2()Lclover/com/atlassian/extras/api/LicenseManager;

.clover/com/atlassian/extras/api/LicenseManager

getLicenseF(Ljava/lang/String;)Lclover/com/atlassian/extras/api/AtlassianLicense;'clover/com/atlassian/extras/api/ProductCLOVER)Lclover/com/atlassian/extras/api/Product;

0clover/com/atlassian/extras/api/AtlassianLicensegetProductLicense[(Lclover/com/atlassian/extras/api/Product;)Lclover/com/atlassian/extras/api/ProductLicense;4clover/com/atlassian/extras/api/clover/CloverLicenseNot a Clover license. getProductLicenses()Ljava/util/Collection;,(Ljava/lang/Object;)Ljava/lang/StringBuffer;f

QisEvaluation()Z

getProduct+()Lclover/com/atlassian/extras/api/Product;getNamec

: getDescriptionc;

getOrganisation0()Lclover/com/atlassian/extras/api/Organisation;,clover/com/atlassian/extras/api/Organisation>

getSupportEntitlementNumbercD getMaintenanceExpiryDate()Ljava/util/Date;java/util/DategetTime()J

H

getExpiryDate= getCreationDate~< License registered to .? 6You have $daysleft day(s) before your license

expires.@ Your license has expired.A C RPlease visit http://www.atlassian.com/ex/GenerateLicense.jspa to obtain a license.

B allowedpkgprefixesgetProperty&(Ljava/lang/String;)Ljava/lang/String;trimc

length()I

java/util/HashSet

TFG !java/util/StringTokenizer#, %'(Ljava/lang/String;Ljava/lang/String;)VR'

$(

hasMoreTokens*

$+ nextToken-c

$.add(Ljava/lang/Object;)Z01

2equalsIgnoreCase(Ljava/lang/String;)Z45

6E 8I :Invalid license data [E1300]. <

rprefixesLjava/util/StringTokenizer;expiresallowedPkgPrefixesStredition

Ljava/lang/NullPointerException;this!Lcom/cenqua/clover/CloverLicense;manager0Lclover/com/atlassian/extras/api/LicenseManager;atlLlicense2Lclover/com/atlassian/extras/api/AtlassianLicense;license6Lclover/com/atlassian/extras/api/clover/CloverLicense;com/cenqua/clover/LicenseLoggerMcom/cenqua/clover/LoggerOgetInstance()Lcom/cenqua/clover/Logger;QR

PS(Lcom/cenqua/clover/Logger;)VRU

NVsetInstance6(Lclover/com/atlassian/extras/common/log/Logger$Log;)VXY

Zr(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;JJJ)

VClover](Clover Evaluation License registered to _nameowner preExpiry

postExpiryterminationcontactInfosoftExpiryDatehardExpiryDategetProductNamegetLicenseName

getSoftExpiry

getHardExpirygetMaintExpiry isDesktop isExpiredjava/lang/SystempcurrentTimeMillisr

qs(J)Zou

vA

xtimeisTerminated{u

|

terminates~

isMaintenanceExpiredu

maintenanceExpires

getOwnerStatementgetPreExpiryStatementgetPostExpiryStatementgetContactInfoStatementgetTerminationStatemen

tisFeatureSupportedfeaturegetFeaturesSupportedgetAllowedPkgPrefixes()Ljava/util/Set;java/util/Collectionsunmodi

fiableSet (Ljava/util/Set;)Ljava/util/Set;

getDaysTillExpiry(J)J

 countDays

nowdays

aMilliseconds

ConstantValueCodeLocalVariableTableLineNumberTable

Exceptions

SourceFileInnerClasses1 "#%&()+,./124578:;<=>?@ABCDEFGHI JK[QYULWYYY*[^aM,eN-+-

ikiW,eN+nLMYQYUpi+sin+v:;O4w3xy!z{<|}[~"CDE#F.E6H;I<JMR2 **+9+/MYQYUiiiiinM,+N-:#MYQYUi-

n6******

*a *QYUiiiin* *

* *:B7*Y "$Y&):,*"/3W*":*79**9 8;':MYQYU=i>invf

?@fAEsBC"|DEFxLGHTIJbKL#RSUWH]L^T`bbgcgijklnop s:vLw^xpywz|}~MS<NYTW[EF

R\+

k**^*+*,** *QYU`i,iin*-*

***;*"**9f

kEFkakbkckdkekfkgkh

kIB!>CIOU[`ejic/*EFjc/*EFc/*EFc/*EFk/*EFl/*EFm/*EFn/*9EFo2*twEFouJ*y*EFz{2*t}EF{uJ**EFz2*tEFuJ**

EFzA9* EF~9* EF9* EFc/*EFc/*EFc/*

EFc/*EFc/*EFuE*; EF/*;EF=*"

*"EF2*tEFY**eB！！EF

:mEF

1.67 jol-core 0.9

1.67.1 Available under license :

The GNU General Public License (GPL)

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license
document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share
and change it. By contrast, the GNU General Public License is intended to
guarantee your freedom to share and change free software--to make sure the
software is free for all its users. This General Public License applies to
most of the Free Software Foundation's software and to any other program whose
authors commit to using it. (Some other Free Software Foundation software is
covered by the GNU Library General Public License instead.) You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our
General Public Licenses are designed to make sure that you have the freedom to
distribute copies of free software (and charge for this service if you wish),
that you receive source code or can get it if you want it, that you can change
the software or use pieces of it in new free programs; and that you know you
can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny
you these rights or to ask you to surrender the rights. These restrictions
translate to certain responsibilities for you if you distribute copies of the
software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for
a fee, you must give the recipients all the rights that you have. You must
make sure that they, too, receive or can get the source code. And you must
show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2)
offer you this license which gives you legal permission to copy, distribute
and/or modify the software.

Also, for each author's protection and ours, we want to make certain that
everyone understands that there is no warranty for this free software. If the
software is modified by someone else and passed on, we want its recipients to
know that what they have is not the original, so that any problems introduced

by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We
wish to avoid the danger that redistributors of a free program will
individually obtain patent licenses, in effect making the program proprietary.
To prevent this, we have made it clear that any patent must be licensed for
everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification
follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice
placed by the copyright holder saying it may be distributed under the terms of
this General Public License. The "Program", below, refers to any such program
or work, and a "work based on the Program" means either the Program or any
derivative work under copyright law: that is to say, a work containing the
Program or a portion of it, either verbatim or with modifications and/or
translated into another language. (Hereinafter, translation is included
without limitation in the term "modification".) Each licensee is addressed as
"you".

Activities other than copying, distribution and modification are not covered by
this License; they are outside its scope. The act of running the Program is
not restricted, and the output from the Program is covered only if its contents
constitute a work based on the Program (independent of having been made by
running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as
you receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice and
disclaimer of warranty; keep intact all the notices that refer to this License
and to the absence of any warranty; and give any other recipients of the
Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may
at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus
forming a work based on the Program, and copy and distribute such modifications
or work under the terms of Section 1 above, provided that you also meet all of
these conditions:

  a) You must cause the modified files to carry prominent notices stating
  that you changed the files and the date of any change.

  b) You must cause any work that you distribute or publish, that in whole or

in part contains or is derived from the Program or any part thereof, to be
licensed as a whole at no charge to all third parties under the terms of
this License.

c) If the modified program normally reads commands interactively when run,
you must cause it, when started running for such interactive use in the
most ordinary way, to print or display an announcement including an
appropriate copyright notice and a notice that there is no warranty (or
else, saying that you provide a warranty) and that users may redistribute
the program under these conditions, and telling the user how to view a copy
of this License. (Exception: if the Program itself is interactive but does
not normally print such an announcement, your work based on the Program is
not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable
sections of that work are not derived from the Program, and can be reasonably
considered independent and separate works in themselves, then this License, and
its terms, do not apply to those sections when you distribute them as separate
works. But when you distribute the same sections as part of a whole which is a
work based on the Program, the distribution of the whole must be on the terms
of this License, whose permissions for other licensees extend to the entire
whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your
rights to work written entirely by you; rather, the intent is to exercise the
right to control the distribution of derivative or collective works based on
the Program.

In addition, mere aggregation of another work not based on the Program with the
Program (or with a work based on the Program) on a volume of a storage or
distribution medium does not bring the other work under the scope of this
License.

3. You may copy and distribute the Program (or a work based on it, under
Section 2) in object code or executable form under the terms of Sections 1 and
2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source
code, which must be distributed under the terms of Sections 1 and 2 above
on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to
give any third party, for a charge no more than your cost of physically
performing source distribution, a complete machine-readable copy of the
corresponding source code, to be distributed under the terms of Sections 1
and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to

distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent

obligations, then as a consequence you may not distribute the Program at all.
For example, if a patent license would not permit royalty-free redistribution
of the Program by all those who receive copies directly or indirectly through
you, then the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply and
the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or
other property right claims or to contest validity of any such claims; this
section has the sole purpose of protecting the integrity of the free software
distribution system, which is implemented by public license practices. Many
people have made generous contributions to the wide range of software
distributed through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing to
distribute software through any other system and a licensee cannot impose that
choice.

This section is intended to make thoroughly clear what is believed to be a
consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain
countries either by patents or by copyrighted interfaces, the original
copyright holder who places the Program under this License may add an explicit
geographical distribution limitation excluding those countries, so that
distribution is permitted only in or among countries not thus excluded. In
such case, this License incorporates the limitation as if written in the body
of this License.

9. The Free Software Foundation may publish revised and/or new versions of the
General Public License from time to time. Such new versions will be similar in
spirit to the present version, but may differ in detail to address new problems
or concerns.

Each version is given a distinguishing version number. If the Program
specifies a version number of this License which applies to it and "any later
version", you have the option of following the terms and conditions either of
that version or of any later version published by the Free Software Foundation.
If the Program does not specify a version number of this License, you may
choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs
whose distribution conditions are different, write to the author to ask for
permission. For software which is copyrighted by the Free Software Foundation,
write to the Free Software Foundation; we sometimes make exceptions for this.
Our decision will be guided by the two goals of preserving the free status of

all derivatives of our free software and of promoting the sharing and reuse of
software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR
THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE
STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE
PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND
PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE,
YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL
ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE
THE
PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR
INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA
BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER
OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible
use to the public, the best way to achieve this is to make it free software
which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach
them to the start of each source file to most effectively convey the exclusion
of warranty; and each file should have at least the "copyright" line and a
pointer to where the full notice is found.

   One line to give the program's name and a brief idea of what it does.

   Copyright (C) <year> <name of author>

   This program is free software; you can redistribute it and/or modify it
   under the terms of the GNU General Public License as published by the Free
   Software Foundation; either version 2 of the License, or (at your option)
   any later version.

   This program is distributed in the hope that it will be useful, but WITHOUT
   ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or

FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for
more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it
starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes
with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free
software, and you are welcome to redistribute it under certain conditions;
type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may be
called something other than 'show w' and 'show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school,
if any, to sign a "copyright disclaimer" for the program, if necessary. Here
is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Library General Public
License instead of this License.

"CLASSPATH" EXCEPTION TO THE GPL

Certain source files distributed by Oracle America and/or its affiliates are
subject to the following clarification and special exception to the GPL, but
only where Oracle has expressly included in the particular source file's header
the words "Oracle designates this particular file as subject to the "Classpath"
exception as provided by Oracle in the LICENSE file that accompanied this code."

Linking this library statically or dynamically with other modules is making
a combined work based on this library. Thus, the terms and conditions of
the GNU General Public License cover the whole combination.

As a special exception, the copyright holders of this library give you
permission to link this library with independent modules to produce an
executable, regardless of the license terms of these independent modules,
and to copy and distribute the resulting executable under terms of your
choice, provided that you also meet, for each linked independent module,
the terms and conditions of the license of that module. An independent
module is a module which is not derived from or based on this library. If
you modify this library, you may extend this exception to your version of
the library, but you are not obligated to do so. If you do not wish to do
so, delete this exception statement from your version.

1.68 jsonpath 2.4.0
1.68.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 2017 Jayway

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.69 jsonpath 0.11.2

1.69.1 Available under license :

Copyright (c) 2014-2016 David Chester <david@fmail.co.uk>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.70 jsreport 1.10.0

1.70.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

 This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

 0. Additional Definitions.

 As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

 "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

 An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

 A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

 The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

 The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

  a) under this License, provided that you make a good faith effort to
  ensure that, in the event an Application does not supply the
  function or data, the facility still operates, and performs
  whatever part of its purpose remains meaningful, or

  b) under the GNU GPL, with none of the additional permissions of
  this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

  a) Give prominent notice with each copy of the object code that the
  Library is used in it and that the Library and its use are
  covered by this License.

  b) Accompany the object code with a copy of the GNU GPL and this license
  document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

  a) Give prominent notice with each copy of the Combined Work that
  the Library is used in it and that the Library and its use are
  covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities,

conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

1.71 jsreport-chrome-pdf 0.3.2
1.71.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser

General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

"The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

   a) under this License, provided that you make a good faith effort to
   ensure that, in the event an Application does not supply the
   function or data, the facility still operates, and performs
   whatever part of its purpose remains meaningful, or

   b) under the GNU GPL, with none of the additional permissions of
   this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

  a) Give prominent notice with each copy of the object code that the
  Library is used in it and that the Library and its use are
  covered by this License.

  b) Accompany the object code with a copy of the GNU GPL and this license
  document.

 4. Combined Works.

 You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

  a) Give prominent notice with each copy of the Combined Work that
  the Library is used in it and that the Library and its use are
  covered by this License.

  b) Accompany the Combined Work with a copy of the GNU GPL and this license
  document.

  c) For a Combined Work that displays copyright notices during
  execution, include the copyright notice for the Library among
  these notices, as well as a reference directing the user to the
  copies of the GNU GPL and this license document.

  d) Do one of the following:

    0) Convey the Minimal Corresponding Source under the terms of this
    License, and the Corresponding Application Code in a form
    suitable for, and under terms that permit, the user to
    recombine or relink the Application with a modified version of
    the Linked Version to produce a modified Combined Work, in the
    manner specified by section 6 of the GNU GPL for conveying
    Corresponding Source.

    1) Use a suitable shared library mechanism for linking with the
    Library. A suitable mechanism is one that (a) uses at run time
    a copy of the Library already present on the user's computer

system, and (b) will operate properly with a modified version
of the Library that is interface-compatible with the Linked
Version.

e) Provide Installation Information, but only if you would otherwise
be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based
on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

1.72 jsreport-pdf-utils 0.5.0

1.72.1 Available under license :

MIT License

Copyright (c) 2017 jsreport

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.73 less 2.7.3

1.73.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.74 less-loader 4.0.5

1.74.1 Available under license :

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.75 log4j-core 2.11.1

1.75.1 Available under license :

Apache Log4j Core
Copyright 1999-2012 Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

ResolverUtil.java
Copyright 2005-2006 Tim Fennell

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,

any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.76 log4j-slf4j-impl 2.11.1
1.76.1 Available under license :

Apache Log4j SLF4J Binding
Copyright 1999-2018 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

  "License" shall mean the terms and conditions for use, reproduction,
  and distribution as defined by Sections 1 through 9 of this document.

  "Licensor" shall mean the copyright owner or entity authorized by
  the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not

pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.77 logbackclassic 1.2.3

1.77.1 Available under license :

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
  <title>License</title>
  <link rel="stylesheet" type="text/css" href="css/common.css" />
  <link rel="stylesheet" type="text/css" href="css/screen.css" media="screen" />
  <link rel="stylesheet" type="text/css" href="css/_print.css" media="print" />

</head>
<body>
  <script type="text/javascript">prefix=';</script>

  <script src="templates/header.js" type="text/javascript"></script>
  <div id="left">
    <script src="templates/left.js" type="text/javascript"></script>
  </div>
  <div id="right">
    <script type="text/javascript" src="templates/right.js" ></script>
  </div>

  <div id="content">

  <div class="section">
    <h2>Logback License</h2>
  </div>

  <p>As of release 0.9.18, logback source code and binaries are
  dual-licensed under the EPL v1.0 and the LGPL 2.1, or more
  formally:</p>

  <p class="source">Logback: the reliable, generic, fast and flexible logging framework.
Copyright (C) 1999-2017, QOS.ch. All rights reserved.

This program and the accompanying materials are dual-licensed under
either the terms of the Eclipse Public License v1.0 as published by
the Eclipse Foundation

 or (per the licensee's choosing)

under the terms of the GNU Lesser General
Public License version 2.1
as published by the Free Software Foundation.</p>

  <!-- ======================================= -->

  <p>The EPL/LGPL dual-license serves several purposes. The LGPL
  license ensures continuity in terms of licensing of the
  logback project. Prior to version 0.9.18, logback was licensed
  (exclusively) under the LGPL v2.1. Moreover, since the EPL is
  deemed <a
  href="http://www.fsf.org/licensing/licenses/index_htm">incompatible
  by the Free Software Foundation, the LGPL will allow various
  licensees, in particular software distributors who may be already
  bound by the terms of the GPL or the LGPL, to distribute our
  software.
  </p>

  <p>On the other hand, the EPL license will placate organizations
  which refuse certain restrictions imposed by the LGPL.
  </p>

  <p>Please note that logback-classic is intended to be used behind
  the SLF4J API, which is licensed under the <a
  href="http://www.slf4j.org/license.html">MIT license.
  </p>

  <p>If you wish to make a significant contribution to the logback
  project, you are invited to file a Contributor
  License Agreement. The purpose of this agreement is to
  formalize the terms of your contribution and to protect the
  project in case of litigation.
  </p>

  <p>Upon request, we may exempt open-source projects from LGPL and
  EPL's reciprocity clauses so that the said projects can develop
  logback extensions under the license of their choice. Exemptions
  are granted on a case by case basis.</p>

<script src="templates/footer.js" type="text/javascript"></script>
</div>
</body>
</html>
Logback LICENSE

Logback: the reliable, generic, fast and flexible logging framework.
Copyright (C) 1999-2015, QOS.ch. All rights reserved.

This program and the accompanying materials are dual-licensed under

either the terms of the Eclipse Public License v1.0 as published by

the Eclipse Foundation

 or (per the licensee's choosing)

under the terms of the GNU Lesser General Public License version 2.1

as published by the Free Software Foundation.

1.78 method-override 2.3.9

1.78.1 Available under license :

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the

'Software'), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.79 mocha 5.0.0

1.79.1 Available under license :

#!/usr/bin/env node

/**

* This script updates the "contributors" property of the root `package.json`.

* It modifies `package.json` in place!

*

```
 * See `.mailmap` for username/email mappings.
 */

'use strict';

const exec = require('child_process').exec;
const path = require('path');
const fs = require('fs');

// list of authors/emails that should not appear in the contributors list, e.g. bots
const BLACKLIST = [
 'greenkeeperio-bot <support@greenkeeper.io>',
 'greenkeeper[bot] <greenkeeper[bot]@users.noreply.github.com>',
 'TJ Holowaychuk <tj@vision-media.ca>' // author
];

const ROOT = path.join(__dirname, '..');
const PKG_FILEPATH = path.join(ROOT, 'package.json');

const pkg = JSON.parse(fs.readFileSync(PKG_FILEPATH, 'utf8'));
const contributorCount = pkg.contributors.length;

// could use `| sort | uniq` here but didn't want to assume 'nix
// see `man git-log` for info about the format
exec('git log --format="%aN <%aE>"', {cwd: ROOT}, (err, gitOutput) => {
 if (err) {
   throw err;
 }

 // result will be many lines of contributors, one or more per commit.
 // we wrap it in a `Set` to get unique values, then attempt to get
 // a consistent sort.
 const contributors = Array.from(new Set(gitOutput.trim().split(/\r?\n/)))
   .filter(contributor => BLACKLIST.indexOf(contributor) < 0)
   .sort((a, b) => a.localeCompare(b, 'en', {sensitivity: 'accent'}));

 const newContributorCount = contributors.length;

 if (newContributorCount !== contributorCount) {
   pkg.contributors = contributors;
   fs.writeFileSync(PKG_FILEPATH, JSON.stringify(pkg, null, 2));

   console.log(
     newContributorCount < contributorCount
       ? `WARNING: Reducing contributor count by ${contributorCount -
         newContributorCount}! Hopefully it's because you updated .mailmap.`
       : `Wrote ${newContributorCount -
         contributorCount} new contributors to package.json.`
```

```
  );
} else {
  console.log('No new contributors; nothing to do.');
}
});
```
Attribution 4.0 International

===

Creative Commons Corporation ("Creative Commons") is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an "as-is" basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

    Considerations for licensors: Our public licenses are
    intended for use by those authorized to give the public
    permission to use material in ways otherwise restricted by
    copyright and certain other rights. Our licenses are
    irrevocable. Licensors should read and understand the terms
    and conditions of the license they choose before applying it.
    Licensors should also secure all rights necessary before
    applying our licenses so that the public can reuse the
    material as expected. Licensors should clearly mark any
    material not subject to the license. This includes other CC-
    licensed material, or material used under an exception or
    limitation to copyright. More considerations for licensors:
    wiki.creativecommons.org/Considerations_for_licensors

    Considerations for the public: By using one of our public
    licenses, a licensor grants the public permission to use the
    licensed material under specified terms and conditions. If
    the licensor's permission is not necessary for any reason--for
    example, because of any applicable exception or limitation to
    copyright--then that use is not regulated by the license. Our

licenses grant only permissions under copyright and certain
other rights that a licensor has authority to grant. Use of
the licensed material may still be restricted for other
reasons, including because others have copyright or other
rights in the material. A licensor may make special requests,
such as asking that all changes be marked or described.
Although not required by our licenses, you are encouraged to
respect those requests where reasonable. More_considerations
for the public:
 wiki.creativecommons.org/Considerations_for_licensees

===
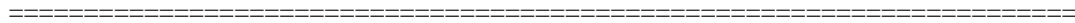
Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree
to be bound by the terms and conditions of this Creative Commons
Attribution 4.0 International Public License ("Public License"). To the
extent this Public License may be interpreted as a contract, You are
granted the Licensed Rights in consideration of Your acceptance of
these terms and conditions, and the Licensor grants You such rights in
consideration of benefits the Licensor receives from making the
Licensed Material available under these terms and conditions.

Section 1 -- Definitions.

 a. Adapted Material means material subject to Copyright and Similar
    Rights that is derived from or based upon the Licensed Material
    and in which the Licensed Material is translated, altered,
    arranged, transformed, or otherwise modified in a manner requiring
    permission under the Copyright and Similar Rights held by the
    Licensor. For purposes of this Public License, where the Licensed
    Material is a musical work, performance, or sound recording,
    Adapted Material is always produced where the Licensed Material is
    synched in timed relation with a moving image.

 b. Adapter's License means the license You apply to Your Copyright
    and Similar Rights in Your contributions to Adapted Material in
    accordance with the terms and conditions of this Public License.

 c. Copyright and Similar Rights means copyright and/or similar rights
    closely related to copyright including, without limitation,
    performance, broadcast, sound recording, and Sui Generis Database
    Rights, without regard to how the rights are labeled or
    categorized. For purposes of this Public License, the rights
    specified in Section 2(b)(1)-(2) are not Copyright and Similar
    Rights.

d. Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. Licensor means the individual(s) or entity(ies) granting rights under this Public License.

i. Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 -- Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to

exercise the Licensed Rights in the Licensed Material to:

   a. reproduce and Share the Licensed Material, in whole or
      in part; and

   b. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where
  Exceptions and Limitations apply to Your use, this Public
  License does not apply, and You do not need to comply with
  its terms and conditions.

3. Term. The term of this Public License is specified in Section
  6(a).

4. Media and formats; technical modifications allowed. The
  Licensor authorizes You to exercise the Licensed Rights in
  all media and formats whether now known or hereafter created,
  and to make technical modifications necessary to do so. The
  Licensor waives and/or agrees not to assert any right or
  authority to forbid You from making technical modifications
  necessary to exercise the Licensed Rights, including
  technical modifications necessary to circumvent Effective
  Technological Measures. For purposes of this Public License,
  simply making modifications authorized by this Section 2(a)
  (4) never produces Adapted Material.

5. Downstream recipients.

   a. Offer from the Licensor -- Licensed Material. Every
      recipient of the Licensed Material automatically
      receives an offer from the Licensor to exercise the
      Licensed Rights under the terms and conditions of this
      Public License.

   b. No downstream restrictions. You may not offer or impose
      any additional or different terms or conditions on, or
      apply any Effective Technological Measures to, the
      Licensed Material if doing so restricts exercise of the
      Licensed Rights by any recipient of the Licensed
      Material.

6. No endorsement. Nothing in this Public License constitutes or
  may be construed as permission to assert or imply that You
  are, or that Your use of the Licensed Material is, connected
  with, or sponsored, endorsed, or granted official status by,
  the Licensor or others designated to receive attribution as
  provided in Section 3(a)(1)(A)(i).

b. Other rights.

   1. Moral rights, such as the right of integrity, are not
      licensed under this Public License, nor are publicity,
      privacy, and/or other similar personality rights; however, to
      the extent possible, the Licensor waives and/or agrees not to
      assert any such rights held by the Licensor to the limited
      extent necessary to allow You to exercise the Licensed
      Rights, but not otherwise.

   2. Patent and trademark rights are not licensed under this
      Public License.

   3. To the extent possible, the Licensor waives any right to
      collect royalties from You for the exercise of the Licensed
      Rights, whether directly or through a collecting society
      under any voluntary or waivable statutory or compulsory
      licensing scheme. In all other cases the Licensor expressly
      reserves any right to collect such royalties.

Section 3 -- License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the
following conditions.

 a. Attribution.

   1. If You Share the Licensed Material (including in modified
      form), You must:

      a. retain the following if it is supplied by the Licensor
         with the Licensed Material:

         i. identification of the creator(s) of the Licensed
            Material and any others designated to receive
            attribution, in any reasonable manner requested by
            the Licensor (including by pseudonym if
            designated);

         ii. a copyright notice;

         iii. a notice that refers to this Public License;

         iv. a notice that refers to the disclaimer of
             warranties;

v. a URI or hyperlink to the Licensed Material to the
extent reasonably practicable;

b. indicate if You modified the Licensed Material and
retain an indication of any previous modifications; and

c. indicate the Licensed Material is licensed under this
Public License, and include the text of, or the URI or
hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any
reasonable manner based on the medium, means, and context in
which You Share the Licensed Material. For example, it may be
reasonable to satisfy the conditions by providing a URI or
hyperlink to a resource that includes the required
information.

3. If requested by the Licensor, You must remove any of the
information required by Section 3(a)(1)(A) to the extent
reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's
License You apply must not prevent recipients of the Adapted
Material from complying with this Public License.

Section 4 -- Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that
apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right
to extract, reuse, reproduce, and Share all or a substantial
portion of the contents of the database;

b. if You include all or a substantial portion of the database
contents in a database in which You have Sui Generis Database
Rights, then the database in which You have Sui Generis Database
Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share
all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not
replace Your obligations under this Public License where the Licensed
Rights include other Copyright and Similar Rights.

Section 5 -- Disclaimer of Warranties and Limitation of Liability.

 a. UNLESS OTHERWISE SEPARATELY UNDERTAKEN BY THE LICENSOR, TO THE
    EXTENT POSSIBLE, THE LICENSOR OFFERS THE LICENSED MATERIAL AS-IS
    AND AS-AVAILABLE, AND MAKES NO REPRESENTATIONS OR WARRANTIES OF
    ANY KIND CONCERNING THE LICENSED MATERIAL, WHETHER EXPRESS,
    IMPLIED, STATUTORY, OR OTHER. THIS INCLUDES, WITHOUT LIMITATION,
    WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR
    PURPOSE, NON-INFRINGEMENT, ABSENCE OF LATENT OR OTHER DEFECTS,
    ACCURACY, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT
    KNOWN OR DISCOVERABLE. WHERE DISCLAIMERS OF WARRANTIES ARE NOT
    ALLOWED IN FULL OR IN PART, THIS DISCLAIMER MAY NOT APPLY TO YOU.

 b. TO THE EXTENT POSSIBLE, IN NO EVENT WILL THE LICENSOR BE LIABLE
    TO YOU ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION,
    NEGLIGENCE) OR OTHERWISE FOR ANY DIRECT, SPECIAL, INDIRECT,
    INCIDENTAL, CONSEQUENTIAL, PUNITIVE, EXEMPLARY, OR OTHER LOSSES,
    COSTS, EXPENSES, OR DAMAGES ARISING OUT OF THIS PUBLIC LICENSE OR
    USE OF THE LICENSED MATERIAL, EVEN IF THE LICENSOR HAS BEEN
    ADVISED OF THE POSSIBILITY OF SUCH LOSSES, COSTS, EXPENSES, OR
    DAMAGES. WHERE A LIMITATION OF LIABILITY IS NOT ALLOWED IN FULL OR
    IN PART, THIS LIMITATION MAY NOT APPLY TO YOU.

 c. The disclaimer of warranties and limitation of liability provided
    above shall be interpreted in a manner that, to the extent
    possible, most closely approximates an absolute disclaimer and
    waiver of all liability.

Section 6 -- Term and Termination.

 a. This Public License applies for the term of the Copyright and
    Similar Rights licensed here. However, if You fail to comply with
    this Public License, then Your rights under this Public License
    terminate automatically.

 b. Where Your right to use the Licensed Material has terminated under
    Section 6(a), it reinstates:

     1. automatically as of the date the violation is cured, provided
        it is cured within 30 days of Your discovery of the
        violation; or

     2. upon express reinstatement by the Licensor.

    For the avoidance of doubt, this Section 6(b) does not affect any
    right the Licensor may have to seek remedies for Your violations
    of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the
Licensed Material under separate terms or conditions or stop
distributing the Licensed Material at any time; however, doing so
will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public
License.

Section 7 -- Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different
terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the
Licensed Material not stated herein are separate from and
independent of the terms and conditions of this Public License.

Section 8 -- Interpretation.

a. For the avoidance of doubt, this Public License does not, and
shall not be interpreted to, reduce, limit, restrict, or impose
conditions on any use of the Licensed Material that could lawfully
be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is
deemed unenforceable, it shall be automatically reformed to the
minimum extent necessary to make it enforceable. If the provision
cannot be reformed, it shall be severed from this Public License
without affecting the enforceability of the remaining terms and
conditions.

c. No term or condition of this Public License will be waived and no
failure to comply consented to unless expressly agreed to by the
Licensor.

d. Nothing in this Public License constitutes or may be interpreted
as a limitation upon, or waiver of, any privileges and immunities
that apply to the Licensor or You, including from the legal
processes of any jurisdiction or authority.

===

Creative Commons is not a party to its public
licenses. Notwithstanding, Creative Commons may elect to apply one of

its public licenses to material it publishes and in those instances

will be considered the Licensor. The text of the Creative Commons

public licenses is dedicated to the public domain under the CC0 Public

Domain Dedication. Except for the limited purpose of indicating that

material is shared under a Creative Commons public license or as

otherwise permitted by the Creative Commons policies published at

creativecommons.org/policies, Creative Commons does not authorize the

use of the trademark "Creative Commons" or any other trademark or logo

of Creative Commons without its prior written consent including,

without limitation, in connection with any unauthorized modifications

to any of its public licenses or any other arrangements,

understandings, or agreements concerning use of licensed material. For

the avoidance of doubt, this paragraph does not form part of the

public licenses.

Creative Commons may be contacted at creativecommons.org.
(The MIT License)

Copyright (c) 2011-2018 JS Foundation and contributors, https://js.foundation

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the

'Software'), to deal in the Software without restriction, including

without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.80 mock-local-storage 1.0.5

1.80.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 letsrock-today

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.81 model-mapper 1.1.0

1.81.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.82 moment 2.22.0

1.82.1 Available under license :

Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person

obtaining a copy of this software and associated documentation

files (the "Software"), to deal in the Software without

restriction, including without limitation the rights to use,

copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the

Software is furnished to do so, subject to the following

conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

1.83 moment-timezone 0.5.16

1.83.1 Available under license :

The MIT License (MIT)

Copyright (c) JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of

this software and associated documentation files (the "Software"), to deal in

the Software without restriction, including without limitation the rights to

use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of

the Software, and to permit persons to whom the Software is furnished to do so,

subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.84 mongodb-migrations 0.8.5

1.84.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014-15 Eugene Mirotin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.85 mongoose 4.11.12

1.86 multer 1.3.0

1.86.1 Available under license :

Copyright (c) 2014 Hage Yaapa <http://www.hacksparrow.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT
OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.87 nock 9.2.5

1.87.1 Available under license :

MIT License

Copyright (c) 2017 Pedro Teixeira and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.88 node-promise 0.5.12

1.89 node-sass 4.5.3
1.89.1 Available under license :

Copyright (C) 2012 by Hampton Catlin

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

The following files in the spec were taken from the original Ruby Sass project which
is copyright Hampton Catlin, Nathan Weizenbaum, and Chris Eppstein and under
the same license.

Copyright (C) 2012-2016 by the Sass Open Source Foundation

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following files in the spec were taken from the original Ruby Sass project which is copyright Hampton Catlin, Nathan Weizenbaum, and Chris Eppstein and under the same license.
Copyright (c) 2013-2016 Andrew Nesbitt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.90 node-sass-chokidar 0.0.03

1.90.1 Available under license :

Copyright (c) 2013-2016 Andrew Nesbitt

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
MIT License

Copyright (c) 2017 Michael Wayman

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.91 nodemon 1.17.3

1.91.1 Available under license :

(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE FOR ANY

DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

* * *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors
(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (C) 2011 Peter Zotov <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
// MIT License

Copyright (C) Roman Shtylman <shtylman@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to

the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(The MIT License)

Copyright (c) 2012 Federico Romero
Copyright (c) 2012-2014 Isaac Z. Schlueter

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be

included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.92 nodemon 1.12.1

1.92.1 Available under license :

(The MIT License)

Copyright (c) 2012-2014 TJ Holowaychuk <vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2014 Nathan LaFreniere and other contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright
    notice, this list of conditions and the following disclaimer in the
    documentation and/or other materials provided with the distribution.
  * The names of any contributors may not be used to endorse or promote
    products derived from this software without specific prior written
    permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS AND CONTRIBUTORS BE LIABLE
FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*   *   *

The complete list of contributors can be found at: https://github.com/hapijs/qs/graphs/contributors
(The MIT License)

Copyright (c) 2010 Sencha Inc.
Copyright (c) 2011 LearnBoost
Copyright (c) 2011 TJ Holowaychuk
Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (C) 2011 Peter Zotov <whitequark@whitequark.org>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
// MIT License

Copyright (C) Roman Shtylman <shtylman@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN
AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2013-2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2013 Jared Hanson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2010 Benjamin Thomas, Robert Kieffer

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(The MIT License)

Copyright (c) 2012 TJ Holowaychuk
Copyright (c) 2014 Douglas Christopher Wilson

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE

SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
The MIT License (MIT)

Copyright 2014 Alex Gorbatchev

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT License (MIT)

Copyright (c) 2014 Jonathan Ong me@jongleberry.com

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(The MIT License)

Copyright (c) 2013 Jonathan Ong <me@jongleberry.com>
Copyright (c) 2014 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
'Software'), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.93 npm-run-all 4.0.2

1.93.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Toru Nagashima

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.94 nyc 11.3.0

1.94.1 Available under license :

ISC License

Copyright (c) 2015, Contributors

Permission to use, copy, modify, and/or distribute this software
for any purpose with or without fee is hereby granted, provided
that the above copyright notice and this permission notice
appear in all copies.

1.95 okhttp 3.9.0

1.95.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

1.96 powermockapimockito 1.7.3

1.96.1 Available under license :

1.97 powermockmoduletestng 1.7.3

1.97.1 Available under license :

The JMockit Testing Toolkit

Copyright (c) 2006-2011 Rogrio Liesenfeld

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ASM: a very small and fast Java bytecode manipulation framework

Copyright (c) 2000-2011 INRIA, France Telecom

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The MIT License

Copyright (c) 2007 Mockito contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable

copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and

do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright 2007-2017 PowerMock Contributors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.98 prop-types 15.5.10

1.98.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.99 Quartz 2.2.1

1.99.1 Available under license :

Copyright Declaration:

Copyright 2003-2016 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or
its subsidiaries and/or its affiliates and/or their licensors.

Trademark and Patent declaration

The name Software AG and all Software AG product names are either trademarks or registered trademarks of
Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other
company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at
http://softwareag.com/licenses.

Third Party declaration

This software may include portions of third-party products. For third-party copyright notices, license terms,
additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party
Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available
under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of
Software AG Products". These documents are part of the product documentation, located at
http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Confidentiality Disclaimer:

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your
License Agreement with Software AG.

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works

that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.100 ratelimit4j-inmemory 0.4.0

1.101 ratelimit4j-redis 0.4.0

1.102 ratelimit4jcore 0.4.0

1.103 react 16.2.0

1.103.1 Available under license :

MIT License

Copyright (c) 2013-present, Facebook, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
MIT License

1.104 react-ace 5.2.2

1.104.1 Available under license :

1.105 react-datetime 2.11.0

1.105.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Javier Marquez

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.106 react-dom 16.2.0

1.106.1 Available under license :

```
/**
 * Copyright (c) 2013-present, Facebook, Inc.
 *
 * This source code is licensed under the MIT license found in the
 * LICENSE file in the root directory of this source tree.
 *
 * @flow
 */

'use strict';

module.exports = require('./server.node');
```

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.107 react-input-autosize 2.2.1

1.107.1 Available under license :

The MIT License (MIT)

Copyright (c) 2018 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
The MIT License (MIT)

Copyright (c) 2018 Jed Watson

1.108 react-intl 2.4.0

1.108.1 Available under license :

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.109 react-paginate 5.0.0

1.109.1 Available under license :

The MIT License (MIT)

Copyright (c) 2016 Adle Delamarche

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.110 react-redux 5.0.5

1.110.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.111 react-router 3.0.5

1.111.1 Available under license :

MIT License

Copyright (c) React Training 2016-2018

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.112 react-scroll 1.5.4

1.112.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015 Joachim Karlsson (fisshy)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.113 react-scrollspy 3.3.4

1.114 react-select 1.2.1

1.114.1 Available under license :

The MIT License (MIT)

Copyright (c) 2018 Jed Watson

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.115 react-tagsinput 3.18.0

1.115.1 Available under license :

The MIT License

Copyright (c) 2015 Ola Holmström <olaholmstrom+github@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.116 react-test-renderer 16.2.0

1.117 redux 3.5.2

1.117.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

The [Redux logo](./logo/) is dedicated to the public domain and licensed under
[CC0]([CC0](http://creativecommons.org/publicdomain/zero/1.0/)).
You can copy, modify, and distribute it, even for commercial purposes, all without asking permission.

[Read more about CC0.](http://creativecommons.org/publicdomain/zero/1.0/)
You can find its legal text below.

Creative Commons Zero v1.0 Universal

CC0 1.0 Universal

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer
exclusive Copyright and Related Rights (defined below) upon the creator and
subsequent owner(s) (each and all, an "owner") of an original work of
authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the
purpose of contributing to a commons of creative, cultural and scientific
works ("Commons") that the public can reliably and without fear of later
claims of infringement build upon, modify, incorporate in other works, reuse
and redistribute as freely as possible in any form whatsoever and for any
purposes, including without limitation commercial purposes. These owners may
contribute to the Commons to promote the ideal of a free culture and the
further production of creative, cultural and scientific works, or to gain
reputation or greater distribution for their Work in part through the use and
efforts of others.

For these and/or other purposes and motivations, and without any expectation
of additional consideration or compensation, the person associating CC0 with a
Work (the "Affirmer"), to the extent that he or she is an owner of Copyright
and Related Rights in the Work, voluntarily elects to apply CC0 to the Work
and publicly distribute the Work under its terms, with knowledge of his or her
Copyright and Related Rights in the Work and the meaning and intended legal
effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be
protected by copyright and related or neighboring rights ("Copyright and
Related Rights"). Copyright and Related Rights include, but are not limited
to, the following:

 i. the right to reproduce, adapt, distribute, perform, display, communicate,
 and translate a Work;

 ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of the License for any reason be judged legally invalid or ineffective under applicable law, such partial invalidity or ineffectiveness shall not

invalidate the remainder of the License, and in such case Affirmer hereby
affirms that he or she will not (i) exercise any of his or her remaining
Copyright and Related Rights in the Work or (ii) assert any associated claims
and causes of action with respect to the Work, in either case contrary to
Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

a. No trademark or patent rights held by Affirmer are waived, abandoned,
surrendered, licensed or otherwise affected by this document.

b. Affirmer offers the Work as-is and makes no representations or warranties
of any kind concerning the Work, express, implied, statutory or otherwise,
including without limitation warranties of title, merchantability, fitness
for a particular purpose, non infringement, or the absence of latent or
other defects, accuracy, or the present or absence of errors, whether or not
discoverable, all to the greatest extent permissible under applicable law.

c. Affirmer disclaims responsibility for clearing rights of other persons
that may apply to the Work or any use thereof, including without limitation
any person's Copyright and Related Rights in the Work. Further, Affirmer
disclaims responsibility for obtaining any necessary consents, permissions
or other rights required for any use of the Work.

d. Affirmer understands and acknowledges that Creative Commons is not a
party to this document and has no duty or obligation with respect to this
CC0 or use of the Work.

For more information, please see
<http://creativecommons.org/publicdomain/zero/1.0/>

1.118 redux-act 1.2.0
1.118.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.119 redux-mock-store 1.5.1

1.119.1 Available under license :

The MIT License (MIT)

Copyright (c) 2017 Arnaud Benard

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.120 redux-saga 0.16.0

1.120.1 Available under license :

The [Redux-Saga logo](logo/) is dedicated to the public domain and licensed under
[CC0]([CC0](http://creativecommons.org/publicdomain/zero/1.0/)).
You can copy, modify, and distribute it, even for commercial purposes, all without asking permission.

[Read more about CC0.](http://creativecommons.org/publicdomain/zero/1.0/)
You can find its legal text below.

Creative Commons Zero v1.0 Universal

CC0 1.0 Universal

Statement of Purpose

The laws of most jurisdictions throughout the world automatically confer
exclusive Copyright and Related Rights (defined below) upon the creator and
subsequent owner(s) (each and all, an "owner") of an original work of
authorship and/or a database (each, a "Work").

Certain owners wish to permanently relinquish those rights to a Work for the
purpose of contributing to a commons of creative, cultural and scientific
works ("Commons") that the public can reliably and without fear of later
claims of infringement build upon, modify, incorporate in other works, reuse
and redistribute as freely as possible in any form whatsoever and for any
purposes, including without limitation commercial purposes. These owners may
contribute to the Commons to promote the ideal of a free culture and the
further production of creative, cultural and scientific works, or to gain
reputation or greater distribution for their Work in part through the use and
efforts of others.

For these and/or other purposes and motivations, and without any expectation
of additional consideration or compensation, the person associating CC0 with a
Work (the "Affirmer"), to the extent that he or she is an owner of Copyright
and Related Rights in the Work, voluntarily elects to apply CC0 to the Work
and publicly distribute the Work under its terms, with knowledge of his or her
Copyright and Related Rights in the Work and the meaning and intended legal
effect of CC0 on those rights.

1. Copyright and Related Rights. A Work made available under CC0 may be
protected by copyright and related or neighboring rights ("Copyright and
Related Rights"). Copyright and Related Rights include, but are not limited
to, the following:

 i. the right to reproduce, adapt, distribute, perform, display, communicate,
 and translate a Work;

ii. moral rights retained by the original author(s) and/or performer(s);

iii. publicity and privacy rights pertaining to a person's image or likeness depicted in a Work;

iv. rights protecting against unfair competition in regards to a Work, subject to the limitations in paragraph 4(a), below;

v. rights protecting the extraction, dissemination, use and reuse of data in a Work;

vi. database rights (such as those arising under Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, and under any national implementation thereof, including any amended or successor version of such directive); and

vii. other similar, equivalent or corresponding rights throughout the world based on applicable law or treaty, and any national implementations thereof.

2. Waiver. To the greatest extent permitted by, but not in contravention of, applicable law, Affirmer hereby overtly, fully, permanently, irrevocably and unconditionally waives, abandons, and surrenders all of Affirmer's Copyright and Related Rights and associated claims and causes of action, whether now known or unknown (including existing as well as future claims and causes of action), in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "Waiver"). Affirmer makes the Waiver for the benefit of each member of the public at large and to the detriment of Affirmer's heirs and successors, fully intending that such Waiver shall not be subject to revocation, rescission, cancellation, termination, or any other legal or equitable action to disrupt the quiet enjoyment of the Work by the public as contemplated by Affirmer's express Statement of Purpose.

3. Public License Fallback. Should any part of the Waiver for any reason be judged legally invalid or ineffective under applicable law, then the Waiver shall be preserved to the maximum extent permitted taking into account Affirmer's express Statement of Purpose. In addition, to the extent the Waiver is so judged Affirmer hereby grants to each affected person a royalty-free, non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights in the Work (i) in all territories worldwide, (ii) for the maximum duration provided by applicable law or treaty (including future time extensions), (iii) in any current or future medium and for any number of copies, and (iv) for any purpose whatsoever, including without limitation commercial, advertising or promotional purposes (the "License"). The License shall be deemed effective as

of the date CC0 was applied by Affirmer to the Work. Should any part of the
License for any reason be judged legally invalid or ineffective under
applicable law, such partial invalidity or ineffectiveness shall not
invalidate the remainder of the License, and in such case Affirmer hereby
affirms that he or she will not (i) exercise any of his or her remaining
Copyright and Related Rights in the Work or (ii) assert any associated claims
and causes of action with respect to the Work, in either case contrary to
Affirmer's express Statement of Purpose.

4. Limitations and Disclaimers.

 a. No trademark or patent rights held by Affirmer are waived, abandoned,
 surrendered, licensed or otherwise affected by this document.

 b. Affirmer offers the Work as-is and makes no representations or warranties
 of any kind concerning the Work, express, implied, statutory or otherwise,
 including without limitation warranties of title, merchantability, fitness
 for a particular purpose, non infringement, or the absence of latent or
 other defects, accuracy, or the present or absence of errors, whether or not
 discoverable, all to the greatest extent permissible under applicable law.

 c. Affirmer disclaims responsibility for clearing rights of other persons
 that may apply to the Work or any use thereof, including without limitation
 any person's Copyright and Related Rights in the Work. Further, Affirmer
 disclaims responsibility for obtaining any necessary consents, permissions
 or other rights required for any use of the Work.

 d. Affirmer understands and acknowledges that Creative Commons is not a
 party to this document and has no duty or obligation with respect to this
 CC0 or use of the Work.

For more information, please see
<http://creativecommons.org/publicdomain/zero/1.0/>
The MIT License (MIT)

Copyright (c) 2015 Yassine Elouafi

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.121 redux-thunk 2.1.0

1.121.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-present Dan Abramov

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.122 request 2.83.0

1.122.1 Available under license :

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1

through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and

You must cause any modified files to carry prominent notices stating that You changed the files; and

You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or

otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.123 request-local 1.0.5

1.124 reselect 3.0.1

1.124.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015-2018 Reselect Contributors

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

1.125 sanitize-html-react 1.13.0

1.125.1 Available under license :

Copyright (c) 2013, 2014, 2015 P'unk Avenue LLC

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.126 sass-loader 6.0.6

1.126.1 Available under license :

Copyright JS Foundation and other contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.127 seamless-immutable 7.1.3

1.127.1 Available under license :

Copyright (c) 2016, Richard Feldman

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this

 list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,

 this list of conditions and the following disclaimer in the documentation

 and/or other materials provided with the distribution.

* Neither the name of seamless-immutable nor the names of its

 contributors may be used to endorse or promote products derived from

 this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.128 sinon 5.0.7

1.128.1 Available under license :

(The BSD License)

Copyright (c) 2010-2017, Christian Johansen, christian@cjohansen.no

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,

     this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice,

     this list of conditions and the following disclaimer in the documentation

and/or other materials provided with the distribution.
  * Neither the name of Christian Johansen nor the names of his contributors
    may be used to endorse or promote products derived from this software
    without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.129 snakeyaml 1.2

1.129.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability

incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.130 sonarqube 2.6.2

1.130.1 Available under license :

```
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
@ParametersAreNonnullByDefault
package org.sonarqube.ws.client.permission;

import javax.annotation.ParametersAreNonnullByDefault;
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
```

```
package org.sonarqube.ws.client.permission;

public class PermissionsWsParameters {
 public static final String CONTROLLER = "api/permissions";

 public static final String PARAM_PERMISSION = "permission";
 public static final String PARAM_ORGANIZATION = "organization";
 public static final String PARAM_GROUP_NAME = "groupName";
 public static final String PARAM_GROUP_ID = "groupId";
 public static final String PARAM_PROJECT_ID = "projectId";
 public static final String PARAM_PROJECT_KEY = "projectKey";
 public static final String PARAM_USER_LOGIN = "login";
 public static final String PARAM_TEMPLATE_ID = "templateId";
 public static final String PARAM_TEMPLATE_NAME = "templateName";
 public static final String PARAM_ID = "id";
 public static final String PARAM_NAME = "name";
 public static final String PARAM_DESCRIPTION = "description";
 public static final String PARAM_PROJECT_KEY_PATTERN = "projectKeyPattern";
 public static final String PARAM_QUALIFIER = "qualifier";

 private PermissionsWsParameters() {
   // static utils only
 }
}
```

```
package com.sonarsource.plugins.license.api;

public interface FooBar {
}
```
```
import * as React from 'react';
import { FormattedMessage } from 'react-intl';
import Tooltip from '../../../components/controls/Tooltip';
import { translate } from '../../../helpers/l10n';

interface Props {
 license?: string;
}

export default function PluginLicense({ license }: Props) {
 if (!license) {
   return null;
 }
 return (
   <Tooltip overlay={license}>
     <li className="little-spacer-bottom marketplace-plugin-license">
       <FormattedMessage
         defaultMessage={translate('marketplace.licensed_under_x')}
         id="marketplace.licensed_under_x"
         values={{
           license: <span className="js-plugin-license">{license}</span>
```

```
        }}
      />
    </li>
  </Tooltip>
);
}
// Jest Snapshot v1, https://goo.gl/fbAQLP

exports[`should display the license field 1`] = `
<Tooltip
 overlay="SonarSource license"
>
 <li
   className="little-spacer-bottom marketplace-plugin-license"
 >
   <FormattedMessage
     defaultMessage="marketplace.licensed_under_x"
     id="marketplace.licensed_under_x"
     values={
      Object {
       "license": <span
         className="js-plugin-license"
       >
        SonarSource license
       </span>,
      }
     }
   />
 </li>
</Tooltip>
`;
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
```

```
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import * as React from 'react';
import { shallow } from 'enzyme';
import PluginLicense from '../PluginLicense';

it('should display the license field', () => {
 expect(shallow(<PluginLicense license="SonarSource license" />)).toMatchSnapshot();
});

it('should not display anything', () => {
 expect(shallow(<PluginLicense />).type()).toBeNull();
});
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import { shape, arrayOf, string, number, func } from 'prop-types';

export const PermissionType = shape({
 key: string.isRequired,
 name: string.isRequired,
 description: string.isRequired,
 usersCount: number.isRequired,
 groupsCount: number.isRequired
});

export const PermissionTemplateType = shape({
 id: string.isRequired,
 name: string.isRequired,
 description: string,
 permissions: arrayOf(PermissionType).isRequired,
 defaultFor: arrayOf(string).isRequired
```

```
});

export const CallbackType = func.isRequired;
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import { lazyLoad } from '../../components/lazyLoad';

const routes = [
 {
   indexRoute: { component: lazyLoad(() => import('./components/AppContainer')) }
 }
];

export default routes;
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
```

```
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import { sortBy } from 'lodash';

export const PERMISSIONS_ORDER = ['user', 'codeviewer', 'issueadmin', 'admin', 'scan'];

/**
 * Sort list of permissions based on predefined order
 * @param {Array} permissions
 * @returns {Array}
 */
export function sortPermissions(permissions) {
  return sortBy(permissions, p => PERMISSIONS_ORDER.indexOf(p.key));
}

/**
 * Populate permissions' details in the list of permission templates
 * @param {Array} permissionTemplates
 * @param {Array} basePermissions
 * @returns {Array}
 */
export function mergePermissionsToTemplates(permissionTemplates, basePermissions) {
  return permissionTemplates.map(permissionTemplate => {
    // it's important to keep the order of the permission template's permissions
    // the same as the order of base permissions
    const permissions = basePermissions.map(basePermission => {
      const projectPermission = permissionTemplate.permissions.find(
        p => p.key === basePermission.key
      );
      return { usersCount: 0, groupsCount: 0, ...basePermission, ...projectPermission };
    });

    return { ...permissionTemplate, permissions };
  });
}

/**
 * Mark default templates
 * @param {Array} permissionTemplates
 * @param {Array} defaultTemplates
 * @returns {Array}
 */
export function mergeDefaultsToTemplates(permissionTemplates, defaultTemplates = []) {
  return permissionTemplates.map(permissionTemplate => {
    const defaultFor = [];

    defaultTemplates.forEach(defaultTemplate => {
      if (defaultTemplate.templateId === permissionTemplate.id) {
```

```
      defaultFor.push(defaultTemplate.qualifier);
    }
  });

  return { ...permissionTemplate, defaultFor };
 });
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import * as React from 'react';
import * as PropTypes from 'prop-types';
import { difference } from 'lodash';
import DeleteForm from './DeleteForm';
import Form from './Form';
import {
 setDefaultPermissionTemplate,
 deletePermissionTemplate,
 updatePermissionTemplate
} from '../../../api/permissions';
import { PermissionTemplate } from '../../../app/types';
import ActionsDropdown, { ActionsDropdownItem } from '../../../components/controls/ActionsDropdown';
import QualifierIcon from '../../../components/icons-components/QualifierIcon';
import { translate } from '../../../helpers/l10n';

export interface Props {
 fromDetails?: boolean;
 organization?: { isDefault?: boolean; key: string };
 permissionTemplate: PermissionTemplate;
 refresh: () => void;
 topQualifiers: string[];
}
```

```typescript
interface State {
 deleteForm: boolean;
 updateModal: boolean;
}

export default class ActionsCell extends React.PureComponent<Props, State> {
 mounted = false;

 static contextTypes = {
  router: PropTypes.object
 };

 state: State = { deleteForm: false, updateModal: false };

 componentDidMount() {
  this.mounted = true;
 }

 componentWillUnmount() {
  this.mounted = false;
 }

 handleUpdateClick = () => {
  this.setState({ updateModal: true });
 };

 handleCloseUpdateModal = () => {
  if (this.mounted) {
   this.setState({ updateModal: false });
  }
 };

 handleSubmitUpdateModal = (data: {
  description: string;
  name: string;
  projectKeyPattern: string;
 }) => {
  return updatePermissionTemplate({ id: this.props.permissionTemplate.id, ...data }).then(
   this.props.refresh
  );
 };

 handleDeleteClick = () => {
  this.setState({ deleteForm: true });
 };

 handleCloseDeleteForm = () => {
```

```
    if (this.mounted) {
      this.setState({ deleteForm: false });
    }
  };

  handleDeleteSubmit = () => {
    return deletePermissionTemplate({ templateId: this.props.permissionTemplate.id }).then(() => {
      const pathname = this.props.organization
        ? `/organizations/${this.props.organization.key}/permission_templates`
        : '/permission_templates';
      this.context.router.replace(pathname);
      this.props.refresh();
    });
  };

  setDefault = (qualifier: string) => () => {
    setDefaultPermissionTemplate(this.props.permissionTemplate.id, qualifier).then(
      this.props.refresh,
      () => {}
    );
  };

  getAvailableQualifiers() {
    const topQualifiers =
      this.props.organization && !this.props.organization.isDefault
        ? ['TRK']
        : this.props.topQualifiers;
    return difference(topQualifiers, this.props.permissionTemplate.defaultFor);
  }

  renderSetDefaultsControl() {
    const availableQualifiers = this.getAvailableQualifiers();

    if (availableQualifiers.length === 0) {
      return null;
    }

    return this.props.topQualifiers.length === 1
      ? this.renderIfSingleTopQualifier(availableQualifiers)
      : this.renderIfMultipleTopQualifiers(availableQualifiers);
  }

  renderSetDefaultLink(qualifier: string, child: React.ReactNode) {
    return (
      <ActionsDropdownItem
        className="js-set-default"
        data-qualifier={qualifier}
        key={qualifier}
```

```
        onClick={this.setDefault(qualifier)}>
        {child}
      </ActionsDropdownItem>
    );
  }

  renderIfSingleTopQualifier(availableQualifiers: string[]) {
    return availableQualifiers.map(qualifier =>
      this.renderSetDefaultLink(
        qualifier,
        <span>{translate('permission_templates.set_default')}</span>
      )
    );
  }

  renderIfMultipleTopQualifiers(availableQualifiers: string[]) {
    return availableQualifiers.map(qualifier =>
      this.renderSetDefaultLink(
        qualifier,
        <span>
          {translate('permission_templates.set_default_for')}{' '}
          <QualifierIcon qualifier={qualifier} /> {translate('qualifiers', qualifier)}
        </span>
      )
    );
  }

  render() {
    const { permissionTemplate: t, organization } = this.props;

    const pathname = organization
      ? `/organizations/${organization.key}/permission_templates`
      : '/permission_templates';

    return (
      <>
        <ActionsDropdown>
          {this.renderSetDefaultsControl()}

          {!this.props.fromDetails && (
            <ActionsDropdownItem to={{ pathname, query: { id: t.id } }}>
              {translate('edit_permissions')}
            </ActionsDropdownItem>
          )}

          <ActionsDropdownItem className="js-update" onClick={this.handleUpdateClick}>
            {translate('update_details')}
          </ActionsDropdownItem>
```

```
      {t.defaultFor.length === 0 && (
        <ActionsDropdownItem
          className="js-delete"
          destructive={true}
          onClick={this.handleDeleteClick}>
          {translate('delete')}
        </ActionsDropdownItem>
      )}
    </ActionsDropdown>

    {this.state.updateModal && (
      <Form
        confirmButtonText={translate('update_verb')}
        header={translate('permission_template.edit_template')}
        onClose={this.handleCloseUpdateModal}
        onSubmit={this.handleSubmitUpdateModal}
        permissionTemplate={t}
      />
    )}

    {this.state.deleteForm && (
      <DeleteForm
        onClose={this.handleCloseDeleteForm}
        onSubmit={this.handleDeleteSubmit}
        permissionTemplate={t}
      />
    )}
    </>
  );
}
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
```

```
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import React from 'react';
import PropTypes from 'prop-types';
import { Link } from 'react-router';
import ActionsCell from './ActionsCell';
import { translate } from '../../../helpers/l10n';

export default class TemplateHeader extends React.PureComponent {
  static propTypes = {
    organization: PropTypes.object,
    template: PropTypes.object.isRequired,
    loading: PropTypes.bool.isRequired,
    refresh: PropTypes.func.isRequired,
    topQualifiers: PropTypes.array.isRequired
  };

  render() {
    const { template, organization } = this.props;

    const pathname = organization
      ? `/organizations/${organization.key}/permission_templates`
      : '/permission_templates';

    return (
      <header id="project-permissions-header" className="page-header">
        <div className="note spacer-bottom">
          <Link to={pathname} className="text-muted">
            {translate('permission_templates.page')}
          </Link>
        </div>

        <h1 className="page-title">{template.name}</h1>

        {this.props.loading && <i className="spinner" />}

        <div className="pull-right">
          <ActionsCell
            organization={this.props.organization}
            permissionTemplate={this.props.template}
            topQualifiers={this.props.topQualifiers}
            refresh={this.props.refresh}
            fromDetails={true}
          />
        </div>
      </header>
    );
```

```
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import * as React from 'react';
import * as PropTypes from 'prop-types';
import Form from './Form';
import { createPermissionTemplate } from '../../../api/permissions';
import { Button } from '../../../components/ui/buttons';
import { translate } from '../../../helpers/l10n';

interface Props {
 organization?: { key: string };
 ready?: boolean;
 refresh: () => Promise<void>;
}

interface State {
 createModal: boolean;
}

export default class Header extends React.PureComponent<Props, State> {
 mounted = false;

 static contextTypes = {
  router: PropTypes.object
 };

 state: State = { createModal: false };

 componentDidMount() {
```

```
  this.mounted = true;
}

componentWillUnmount() {
  this.mounted = false;
}

handleCreateClick = () => {
  this.setState({ createModal: true });
};

handleCreateModalClose = () => {
  if (this.mounted) {
    this.setState({ createModal: false });
  }
};

handleCreateModalSubmit = (data: {
  description: string;
  name: string;
  projectKeyPattern: string;
}) => {
  const organization = this.props.organization && this.props.organization.key;
  return createPermissionTemplate({ ...data, organization }).then(response => {
    this.props.refresh().then(() => {
      const pathname = organization
        ? `/organizations/${organization}/permission_templates`
        : '/permission_templates';
      this.context.router.push({ pathname, query: { id: response.permissionTemplate.id } });
    });
  });
};

render() {
  return (
    <header className="page-header" id="project-permissions-header">
      <h1 className="page-title">{translate('permission_templates.page')}</h1>

      {!this.props.ready && <i className="spinner" />}

      <div className="page-actions">
        <Button onClick={this.handleCreateClick}>{translate('create')}</Button>

        {this.state.createModal && (
          <Form
            confirmButtonText={translate('create')}
            header={translate('permission_template.new_template')}
            onClose={this.handleCreateModalClose}
```

```
        onSubmit={this.handleCreateModalSubmit}
      />
    )}
  </div>

  <p className="page-description">{translate('permission_templates.page.description')}</p>
  </header>
);
}
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import React from 'react';
import PropTypes from 'prop-types';
import HelpTooltip from '../../../components/controls/HelpTooltip';
import { translate } from '../../../helpers/l10n';
import InstanceMessage from '../../../components/common/InstanceMessage';

export default class ListHeader extends React.PureComponent {
 static propTypes = {
   organization: PropTypes.object,
   permissions: PropTypes.array.isRequired
 };

 renderTooltip = permission =>
   permission.key === 'user' || permission.key === 'codeviewer' ? (
     <div>
       <InstanceMessage message={translate('projects_role', permission.key, 'desc')} />
       <div className="alert alert-warning spacer-top">
         {translate('projects_role.public_projects_warning')}
       </div>
```

```
        </div>
      ) : (
        <InstanceMessage message={translate('projects_role', permission.key, 'desc')} />
      );

  render() {
    const cells = this.props.permissions.map(permission => (
      <th className="permission-column" key={permission.key}>
        <span className="text-middle">{translate('projects_role', permission.key)}</span>
        <HelpTooltip className="spacer-left" overlay={this.renderTooltip(permission)} />
      </th>
    ));

    return (
      <thead>
        <tr>
          <th> </th>
          {cells}
          <th className="thin nowrap text-right"> </th>
        </tr>
      </thead>
    );
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import React from 'react';
import PropTypes from 'prop-types';
import Home from './Home';
import Template from './Template';
import OrganizationHelmet from '../../../components/common/OrganizationHelmet';
```

```
import Suggestions from '../../../app/components/embed-docs-modal/Suggestions';
import { getPermissionTemplates } from '../../../api/permissions';
import { sortPermissions, mergePermissionsToTemplates, mergeDefaultsToTemplates } from '../utils';
import { translate } from '../../../helpers/l10n';
import '../../permissions/styles.css';

export default class App extends React.PureComponent {
 static propTypes = {
   location: PropTypes.object.isRequired,
   organization: PropTypes.object,
   topQualifiers: PropTypes.array.isRequired
 };

 state = {
   ready: false,
   permissions: [],
   permissionTemplates: []
 };

 componentDidMount() {
   this.mounted = true;
   this.requestPermissions();
 }

 componentWillUnmount() {
   this.mounted = false;
 }

 requestPermissions = () => {
   const { organization } = this.props;
   const request = organization
     ? getPermissionTemplates(organization.key)
     : getPermissionTemplates();
   return request.then(r => {
    if (this.mounted) {
      const permissions = sortPermissions(r.permissions);
      const permissionTemplates = mergeDefaultsToTemplates(
        mergePermissionsToTemplates(r.permissionTemplates, permissions),
        r.defaultTemplates
      );
      this.setState({
        ready: true,
        permissionTemplates,
        permissions
      });
    }
   });
 };
```

```
renderTemplate(id) {
  if (!this.state.ready) {
    return null;
  }

  const template = this.state.permissionTemplates.find(t => t.id === id);
  return (
    <Template
      organization={this.props.organization}
      template={template}
      refresh={this.requestPermissions}
      topQualifiers={this.props.topQualifiers}
    />
  );
}

renderHome() {
  return (
    <Home
      organization={this.props.organization}
      topQualifiers={this.props.topQualifiers}
      permissions={this.state.permissions}
      permissionTemplates={this.state.permissionTemplates}
      ready={this.state.ready}
      refresh={this.requestPermissions}
    />
  );
}

render() {
  const { id } = this.props.location.query;
  return (
    <div>
      <Suggestions suggestions="permission_templates" />
      <OrganizationHelmet
        title={translate('permission_templates.page')}
        organization={this.props.organization}
      />

      {id && this.renderTemplate(id)}
      {!id && this.renderHome()}
    </div>
  );
}
}
/*
 * SonarQube
```

```
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import React from 'react';
import PropTypes from 'prop-types';
import Defaults from './Defaults';

export default class TemplateDetails extends React.PureComponent {
 static propTypes = {
   organization: PropTypes.object,
   template: PropTypes.object.isRequired
 };

 render() {
  const { template } = this.props;

  return (
    <div className="big-spacer-bottom">
      {template.defaultFor.length > 0 && (
        <div className="spacer-top js-defaults">
          <Defaults permissionTemplate={template} organization={this.props.organization} />
        </div>
      )}

      {!!template.description && (
        <div className="spacer-top js-description">{template.description}</div>
      )}

      {!!template.projectKeyPattern && (
        <div className="spacer-top js-project-key-pattern">
          Project Key Pattern: <code>{template.projectKeyPattern}</code>
        </div>
      )}
    </div>
```

```
  );
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import React from 'react';
import PropTypes from 'prop-types';
import ListHeader from './ListHeader';
import ListItem from './ListItem';
import { PermissionTemplateType, CallbackType } from '../propTypes';

export default class List extends React.PureComponent {
 static propTypes = {
   organization: PropTypes.object,
   permissionTemplates: PropTypes.arrayOf(PermissionTemplateType).isRequired,
   permissions: PropTypes.array.isRequired,
   topQualifiers: PropTypes.array.isRequired,
   refresh: CallbackType
 };

 render() {
   const permissionTemplates = this.props.permissionTemplates.map(p => (
     <ListItem
       key={p.id}
       organization={this.props.organization}
       permissionTemplate={p}
       topQualifiers={this.props.topQualifiers}
       refresh={this.props.refresh}
     />
   ));
```

```
      return (
        <div className="boxed-group boxed-group-inner">
          <table id="permission-templates" className="data zebra permissions-table">
            <ListHeader organization={this.props.organization} permissions={this.props.permissions} />
            <tbody>{permissionTemplates}</tbody>
          </table>
        </div>
      );
    }
  }
  /*
   * SonarQube
   * Copyright (C) 2009-2018 SonarSource SA
   * mailto:info AT sonarsource DOT com
   *
   * This program is free software; you can redistribute it and/or
   * modify it under the terms of the GNU Lesser General Public
   * License as published by the Free Software Foundation; either
   * version 3 of the License, or (at your option) any later version.
   *
   * This program is distributed in the hope that it will be useful,
   * but WITHOUT ANY WARRANTY; without even the implied warranty of
   * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   * Lesser General Public License for more details.
   *
   * You should have received a copy of the GNU Lesser General Public License
   * along with this program; if not, write to the Free Software Foundation,
   * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
   */
  import React from 'react';
  import PropTypes from 'prop-types';
  import Helmet from 'react-helmet';
  import { debounce } from 'lodash';
  import TemplateHeader from './TemplateHeader';
  import TemplateDetails from './TemplateDetails';
  import HoldersList from '../../permissions/shared/components/HoldersList';
  import SearchForm from '../../permissions/shared/components/SearchForm';
  import { PERMISSIONS_ORDER_FOR_PROJECT } from '../../permissions/project/constants';
  import * as api from '../../../api/permissions';
  import { translate } from '../../../helpers/l10n';

  export default class Template extends React.PureComponent {
   static propTypes = {
     organization: PropTypes.object,
     template: PropTypes.object.isRequired,
     refresh: PropTypes.func.isRequired,
     topQualifiers: PropTypes.array.isRequired
   };
```

```
state = {
 loading: false,
 users: [],
 groups: [],
 query: ",
 filter: 'all',
 selectedPermission: null
};

componentDidMount() {
 this.mounted = true;
 this.requestHolders();
}

componentWillUnmount() {
 this.mounted = false;
}

requestHolders = realQuery => {
 this.setState({ loading: true });

 const { template } = this.props;
 const { query, filter, selectedPermission } = this.state;
 const requests = [];

 const finalQuery = realQuery != null ? realQuery : query;

 if (filter !== 'groups') {
  requests.push(api.getPermissionTemplateUsers(template.id, finalQuery, selectedPermission));
 } else {
  requests.push(Promise.resolve([]));
 }

 if (filter !== 'users') {
  requests.push(api.getPermissionTemplateGroups(template.id, finalQuery, selectedPermission));
 } else {
  requests.push(Promise.resolve([]));
 }

 return Promise.all(requests).then(responses => {
  if (this.mounted) {
   this.setState({
    users: responses[0],
    groups: responses[1],
    loading: false
   });
  }
```

```javascript
      });
    };

    handleToggleUser = (user, permission) => {
      if (user.login === '<creator>') {
        return this.handleToggleProjectCreator(user, permission);
      }
      const { template, organization } = this.props;
      const hasPermission = user.permissions.includes(permission);
      const data = {
        templateId: template.id,
        login: user.login,
        permission
      };
      if (organization) {
        data.organization = organization.key;
      }
      const request = hasPermission
        ? api.revokeTemplatePermissionFromUser(data)
        : api.grantTemplatePermissionToUser(data);
      return request.then(() => this.requestHolders()).then(this.props.refresh);
    };

    handleToggleProjectCreator = (user, permission) => {
      const { template } = this.props;
      const hasPermission = user.permissions.includes(permission);
      const request = hasPermission
        ? api.removeProjectCreatorFromTemplate(template.id, permission)
        : api.addProjectCreatorToTemplate(template.id, permission);
      return request.then(() => this.requestHolders()).then(this.props.refresh);
    };

    handleToggleGroup = (group, permission) => {
      const { template, organization } = this.props;
      const hasPermission = group.permissions.includes(permission);
      const data = {
        templateId: template.id,
        groupName: group.name,
        permission
      };
      if (organization) {
        Object.assign(data, { organization: organization.key });
      }
      const request = hasPermission
        ? api.revokeTemplatePermissionFromGroup(data)
        : api.grantTemplatePermissionToGroup(data);
      return request.then(() => this.requestHolders()).then(this.props.refresh);
    };
```

```
handleSearch = query => {
  this.setState({ query });
  this.requestHolders(query);
};

handleFilter = filter => {
  this.setState({ filter }, this.requestHolders);
};

handleSelectPermission = selectedPermission => {
  if (selectedPermission === this.state.selectedPermission) {
    this.setState({ selectedPermission: null }, this.requestHolders);
  } else {
    this.setState({ selectedPermission }, this.requestHolders);
  }
};

shouldDisplayCreator = creatorPermissions => {
  const { filter, query, selectedPermission } = this.state;
  const CREATOR_NAME = translate('permission_templates.project_creators');

  const isFiltered = filter !== 'all';

  const matchQuery = !query || CREATOR_NAME.toLocaleLowerCase().includes(query.toLowerCase());

  const matchPermission =
    selectedPermission == null || creatorPermissions.includes(selectedPermission);

  return !isFiltered && matchQuery && matchPermission;
};

render() {
  const permissions = PERMISSIONS_ORDER_FOR_PROJECT.map(p => ({
    key: p,
    name: translate('projects_role', p),
    description: translate('projects_role', p, 'desc')
  }));

  const allUsers = [...this.state.users];

  const creatorPermissions = this.props.template.permissions
    .filter(p => p.withProjectCreator)
    .map(p => p.key);

  if (this.shouldDisplayCreator(creatorPermissions)) {
    const creator = {
      login: '<creator>',
```

```
        name: translate('permission_templates.project_creators'),
        permissions: creatorPermissions
      };

      allUsers.unshift(creator);
    }

    return (
      <div className="page page-limited">
        <Helmet title={this.props.template.name} />

        <TemplateHeader
          loading={this.state.loading}
          organization={this.props.organization}
          refresh={this.props.refresh}
          template={this.props.template}
          topQualifiers={this.props.topQualifiers}
        />

        <TemplateDetails organization={this.props.organization} template={this.props.template} />

        <HoldersList
          groups={this.state.groups}
          onSelectPermission={this.handleSelectPermission}
          onToggleGroup={this.handleToggleGroup}
          onToggleUser={this.handleToggleUser}
          permissions={permissions}
          selectedPermission={this.state.selectedPermission}
          showPublicProjectsWarning={true}
          users={allUsers}>
          <SearchForm
            filter={this.state.filter}
            onFilter={this.handleFilter}
            onSearch={this.handleSearch}
            query={this.state.query}
          />
        </HoldersList>
      </div>
    );
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
```

```
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import React from 'react';
import PropTypes from 'prop-types';
import { Link } from 'react-router';
import Defaults from './Defaults';
import { PermissionTemplateType } from '../propTypes';

export default class NameCell extends React.PureComponent {
 static propTypes = {
  organization: PropTypes.object,
  permissionTemplate: PermissionTemplateType.isRequired
 };

 render() {
  const { permissionTemplate: t, organization } = this.props;

  const pathname = organization
   ? `/organizations/${organization.key}/permission_templates`
   : '/permission_templates';

  return (
   <td>
    <Link to={{ pathname, query: { id: t.id } }}>
     <strong className="js-name">{t.name}</strong>
    </Link>

    {t.defaultFor.length > 0 && (
     <div className="spacer-top js-defaults">
      <Defaults
       permissionTemplate={this.props.permissionTemplate}
       organization={organization}
      />
     </div>
    )}

    {!!t.description && <div className="spacer-top js-description">{t.description}</div>}
```

```jsx
      {!!t.projectKeyPattern && (
        <div className="spacer-top js-project-key-pattern">
          Project Key Pattern: <code>{t.projectKeyPattern}</code>
        </div>
      )}
    </td>
  );
}
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import React from 'react';
import { PermissionType } from '../propTypes';
import { translate } from '../../../helpers/l10n';

export default class PermissionCell extends React.PureComponent {
  static propTypes = {
    permission: PermissionType.isRequired
  };

  render() {
    const { permission: p } = this.props;

    return (
      <td className="permission-column" data-permission={p.key}>
        <div className="permission-column-inner">
          <ul>
            {p.withProjectCreator && (
              <li className="little-spacer-bottom">
                {translate('permission_templates.project_creators')}
              </li>
```

```
        )}
        <li className="little-spacer-bottom">
          <strong>{p.usersCount}</strong>
          {' user(s)'}
        </li>
        <li>
          <strong>{p.groupsCount}</strong>
          {' group(s)'}
        </li>
      </ul>
    </div>
  </td>
);
}
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import * as React from 'react';
import { PermissionTemplate } from '../../../app/types';
import SimpleModal from '../../../components/controls/SimpleModal';
import DeferredSpinner from '../../../components/common/DeferredSpinner';
import { SubmitButton, ResetButtonLink } from '../../../components/ui/buttons';
import { translate, translateWithParameters } from '../../../helpers/l10n';

interface Props {
  onClose: () => void;
  onSubmit: () => Promise<void>;
  permissionTemplate: PermissionTemplate;
}

export default function DeleteForm({ onClose, onSubmit, permissionTemplate: t }: Props) {
```

```
 const header = translate('permission_template.delete_confirm_title');

 return (
   <SimpleModal header={header} onClose={onClose} onSubmit={onSubmit}>
    {(({ onCloseClick, onFormSubmit, submitting }) => (
     <form onSubmit={onFormSubmit}>
       <header className="modal-head">
        <h2>{header}</h2>
       </header>

       <div className="modal-body">
        {translateWithParameters(
         'permission_template.do_you_want_to_delete_template_xxx',
         t.name
        )}
       </div>

       <footer className="modal-foot">
        <DeferredSpinner className="spacer-right" loading={submitting} />
        <SubmitButton className="button-red" disabled={submitting}>
         {translate('delete')}
        </SubmitButton>
        <ResetButtonLink disabled={submitting} onClick={onCloseClick}>
         {translate('cancel')}
        </ResetButtonLink>
       </footer>
     </form>
    )}
   </SimpleModal>
 );
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
```

```
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import React from 'react';
import PropTypes from 'prop-types';
import { sortBy } from 'lodash';
import { translate, translateWithParameters } from '../../../helpers/l10n';
import { PermissionTemplateType } from '../propTypes';

export default class Defaults extends React.PureComponent {
 static propTypes = {
   organization: PropTypes.object,
   permissionTemplate: PermissionTemplateType.isRequired
 };

 render() {
   const qualifiersToDisplay =
     this.props.organization && !this.props.organization.isDefault
       ? ['TRK']
       : this.props.permissionTemplate.defaultFor;

   const qualifiers = sortBy(qualifiersToDisplay)
     .map(qualifier => translate('qualifiers', qualifier))
     .join(', ');

   return (
     <div>
       <span className="badge spacer-right">
         {translateWithParameters('permission_template.default_for', qualifiers)}
       </span>
     </div>
   );
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
```

```
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import { connect } from 'react-redux';
import App from './App';
import forSingleOrganization from '../../organizations/forSingleOrganization';
import { getAppState } from '../../../store/rootReducer';
import { getRootQualifiers } from '../../../store/appState/duck';

const mapStateToProps = state => ({
 // treat applications as portfolios
 topQualifiers: getRootQualifiers(getAppState(state)).filter(q => q !== 'APP')
});

export default forSingleOrganization(connect(mapStateToProps)(App));
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
import * as React from 'react';
import DeferredSpinner from '../../../components/common/DeferredSpinner';
import SimpleModal from '../../../components/controls/SimpleModal';
import { translate } from '../../../helpers/l10n';
import { SubmitButton, ResetButtonLink } from '../../../components/ui/buttons';

interface Props {
 confirmButtonText: string;
 header: string;
 permissionTemplate?: { description?: string; name: string; projectKeyPattern?: string };
 onClose: () => void;
 onSubmit: (
   data: { description: string; name: string; projectKeyPattern: string }
```

```typescript
  ) => Promise<void>;
}

interface State {
 description: string;
 name: string;
 projectKeyPattern: string;
}

export default class Form extends React.PureComponent<Props, State> {
 mounted = false;

 constructor(props: Props) {
  super(props);
  this.state = {
   description: (props.permissionTemplate && props.permissionTemplate.description) || '',
   name: (props.permissionTemplate && props.permissionTemplate.name) || '',
   projectKeyPattern:
    (props.permissionTemplate && props.permissionTemplate.projectKeyPattern) || ''
  };
 }

 handleSubmit = () => {
  return this.props
   .onSubmit({
    description: this.state.description,
    name: this.state.name,
    projectKeyPattern: this.state.projectKeyPattern
   })
   .then(this.props.onClose);
 };

 handleNameChange = (event: React.ChangeEvent<HTMLInputElement>) => {
  this.setState({ name: event.currentTarget.value });
 };

 handleDescriptionChange = (event: React.ChangeEvent<HTMLTextAreaElement>) => {
  this.setState({ description: event.currentTarget.value });
 };

 handleProjectKeyPatternChange = (event: React.ChangeEvent<HTMLInputElement>) => {
  this.setState({ projectKeyPattern: event.currentTarget.value });
 };

 render() {
  return (
   <SimpleModal
    header={this.props.header}
```

```
onClose={this.props.onClose}
onSubmit={this.handleSubmit}>
{(({ onCloseClick, onFormSubmit, submitting }) => (
  <form id="permission-template-form" onSubmit={onFormSubmit}>
    <header className="modal-head">
      <h2>{this.props.header}</h2>
    </header>

    <div className="modal-body">
      <div className="modal-field">
        <label htmlFor="permission-template-name">
          {translate('name')}
          <em className="mandatory">*</em>
        </label>
        <input
          autoFocus={true}
          id="permission-template-name"
          maxLength={256}
          name="name"
          onChange={this.handleNameChange}
          required={true}
          type="text"
          value={this.state.name}
        />
        <div className="modal-field-description">{translate('should_be_unique')}</div>
      </div>

      <div className="modal-field">
        <label htmlFor="permission-template-description">{translate('description')}</label>
        <textarea
          id="permission-template-description"
          name="description"
          onChange={this.handleDescriptionChange}
          value={this.state.description}
        />
      </div>

      <div className="modal-field">
        <label htmlFor="permission-template-project-key-pattern">
          {translate('permission_template.key_pattern')}
        </label>
        <input
          id="permission-template-project-key-pattern"
          maxLength={500}
          name="projectKeyPattern"
          onChange={this.handleProjectKeyPatternChange}
          type="text"
          value={this.state.projectKeyPattern}
```

```
            />
            <div className="modal-field-description">
              {translate('permission_template.key_pattern.description')}
            </div>
          </div>
        </div>

        <footer className="modal-foot">
          <DeferredSpinner className="spacer-right" loading={submitting} />
          <SubmitButton disabled={submitting} id="permission-template-submit">
            {this.props.confirmButtonText}
          </SubmitButton>
          <ResetButtonLink
            disabled={submitting}
            id="permission-template-cancel"
            onClick={onCloseClick}>
            {translate('cancel')}
          </ResetButtonLink>
        </footer>
      </form>
    )}
  </SimpleModal>
  );
 }
}
// Jest Snapshot v1, https://goo.gl/fbAQLP

exports[`should render one qualifier 1`] = `
<div>
 <span
   className="badge spacer-right"
 >
   permission_template.default_for.qualifiers.DEV
 </span>
</div>
`;

exports[`should render only projects for custom organization 1`] = `
<div>
 <span
   className="badge spacer-right"
 >
   permission_template.default_for.qualifiers.TRK
 </span>
</div>
`;

exports[`should render several qualifiers 1`] = `
```

```
<div>
 <span
  className="badge spacer-right"
 >
  permission_template.default_for.qualifiers.TRK, qualifiers.VW
 </span>
</div>
`;

exports[`should render several qualifiers for default organization 1`] = `
<div>
 <span
  className="badge spacer-right"
 >
  permission_template.default_for.qualifiers.TRK, qualifiers.VW
 </span>
</div>
`;
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import * as React from 'react';
import { shallow } from 'enzyme';
import ActionsCell, { Props } from '../ActionsCell';

const SAMPLE = {
 createdAt: '2018-01-01',
 id: 'id',
 name: 'name',
 permissions: [],
 defaultFor: []
};
```

```
function renderActionsCell(props?: Partial<Props>) {
 return shallow(
   <ActionsCell
     permissionTemplate={SAMPLE}
     refresh={() => true}
     topQualifiers={['TRK', 'VW']}
     {...props}
   />
 );
}

it('should set default', () => {
 const setDefault = renderActionsCell().find('.js-set-default');
 expect(setDefault.length).toBe(2);
 expect(setDefault.at(0).prop('data-qualifier')).toBe('TRK');
 expect(setDefault.at(1).prop('data-qualifier')).toBe('VW');
});

it('should not set default', () => {
 const permissionTemplate = { ...SAMPLE, defaultFor: ['TRK', 'VW'] };
 const setDefault = renderActionsCell({ permissionTemplate }).find('.js-set-default');
 expect(setDefault.length).toBe(0);
});

it('should display all qualifiers for default organization', () => {
 const organization = { isDefault: true, key: 'org' };
 const setDefault = renderActionsCell({ organization }).find('.js-set-default');
 expect(setDefault.length).toBe(2);
 expect(setDefault.at(0).prop('data-qualifier')).toBe('TRK');
 expect(setDefault.at(1).prop('data-qualifier')).toBe('VW');
});

it('should display only projects for custom organization', () => {
 const organization = { isDefault: false, key: 'org' };
 const setDefault = renderActionsCell({ organization }).find('.js-set-default');
 expect(setDefault.length).toBe(1);
 expect(setDefault.at(0).prop('data-qualifier')).toBe('TRK');
});
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
```

```
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import { shallow } from 'enzyme';
import React from 'react';
import Defaults from '../Defaults';

const SAMPLE = {
 id: 'id',
 name: 'name',
 permissions: []
};

it('should render one qualifier', () => {
 const sample = { ...SAMPLE, defaultFor: ['DEV'] };
 const output = shallow(<Defaults permissionTemplate={sample} />);
 expect(output).toMatchSnapshot();
});

it('should render several qualifiers', () => {
 const sample = { ...SAMPLE, defaultFor: ['TRK', 'VW'] };
 const output = shallow(<Defaults permissionTemplate={sample} />);
 expect(output).toMatchSnapshot();
});

it('should render several qualifiers for default organization', () => {
 const sample = { ...SAMPLE, defaultFor: ['TRK', 'VW'] };
 const organization = { isDefault: true };
 const output = shallow(<Defaults permissionTemplate={sample} organization={organization} />);
 expect(output).toMatchSnapshot();
});

it('should render only projects for custom organization', () => {
 const sample = { ...SAMPLE, defaultFor: ['TRK', 'VW'] };
 const organization = { isDefault: false };
 const output = shallow(<Defaults permissionTemplate={sample} organization={organization} />);
 expect(output).toMatchSnapshot();
});
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
```

```
import React from 'react';
import PropTypes from 'prop-types';
import Helmet from 'react-helmet';
import Header from './Header';
import List from './List';
import { translate } from '../../../helpers/l10n';

export default class Home extends React.PureComponent {
 static propTypes = {
   organization: PropTypes.object,
   topQualifiers: PropTypes.array.isRequired,
   permissions: PropTypes.array.isRequired,
   permissionTemplates: PropTypes.array.isRequired,
   ready: PropTypes.bool.isRequired,
   refresh: PropTypes.func.isRequired
 };

 render() {
  return (
    <div className="page page-limited">
      <Helmet title={translate('permission_templates.page')} />

      <Header
        organization={this.props.organization}
        ready={this.props.ready}
        refresh={this.props.refresh}
      />

      <List
        organization={this.props.organization}
        permissionTemplates={this.props.permissionTemplates}
        permissions={this.props.permissions}
```

```
          topQualifiers={this.props.topQualifiers}
          refresh={this.props.refresh}
        />
      </div>
    );
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
import React from 'react';
import PropTypes from 'prop-types';
import NameCell from './NameCell';
import PermissionCell from './PermissionCell';
import ActionsCell from './ActionsCell';
import { PermissionTemplateType, CallbackType } from '../propTypes';

export default class ListItem extends React.PureComponent {
  static propTypes = {
    organization: PropTypes.object,
    permissionTemplate: PermissionTemplateType.isRequired,
    topQualifiers: PropTypes.array.isRequired,
    refresh: CallbackType
  };

  render() {
    const permissions = this.props.permissionTemplate.permissions.map(p => (
      <PermissionCell key={p.key} permission={p} />
    ));

    return (
      <tr data-id={this.props.permissionTemplate.id} data-name={this.props.permissionTemplate.name}>
```

```jsx
        <NameCell
          organization={this.props.organization}
          permissionTemplate={this.props.permissionTemplate}
          topQualifiers={this.props.topQualifiers}
        />

        {permissions}

        <td className="nowrap thin text-right">
          <ActionsCell
            organization={this.props.organization}
            permissionTemplate={this.props.permissionTemplate}
            topQualifiers={this.props.topQualifiers}
            refresh={this.props.refresh}
          />
        </td>
      </tr>
    );
  }
}
```

```json
{
  "paging": {
    "pageIndex": 1,
    "pageSize": 100,
    "total": 3
  },
  "groups": [
    {
      "name": "Anyone",
      "permissions": []
    },
    {
      "id": "1",
      "name": "sonar-administrators",
      "description": "System administrators",
      "permissions": []
    },
    {
      "id": "2",
      "name": "sonar-users",
      "description": "Any new users created will automatically join this group",
      "permissions": []
    }
  ]
}
{
  "permissionTemplate": {
    "id": "af8cb8cc-1e78-4c4e-8c00-ee8e814009a5",
```

```
      "name": "Finance",
      "description": "Permissions for financially related projects",
      "projectKeyPattern": ".*\\.finance\\..*",
      "createdAt": "2001-09-09T03:46:40+0200",
      "updatedAt": "2015-08-25T16:18:48+0200"
    }
  }
{
  "paging": {
    "pageIndex": 1,
    "pageSize": 20,
    "total": 2
  },
  "users": [
    {
      "login": "admin",
      "name": "Administrator",
      "email": "admin@admin.com",
      "avatar": "64e1b8d34f425d19e1ee2ea7236d3028",
      "permissions": ["codeviewer"]
    },
    {
      "login": "george.orwell",
      "name": "George Orwell",
      "email": "george.orwell@1984.net",
      "avatar": "583af86a274c1027ef078cada831babf",
      "permissions": ["admin", "codeviewer"]
    }
  ]
}
{
  "paging": {
    "pageIndex": 1,
    "pageSize": 20,
    "total": 3
  },
  "groups": [
    {
      "name": "Anyone",
      "permissions": [
        "issueadmin",
        "user"
      ]
    },
    {
      "name": "sonar-administrators",
      "description": "System administrators",
      "permissions": [
```

```
          "issueadmin"
        ]
      },
      {
        "name": "sonar-users",
        "description": "Any new users created will automatically join this group",
        "permissions": [
          "issueadmin"
        ]
      }
    ]
}
{
    "permissionTemplates": [
      {
        "id": "AU-Tpxb--iU5OvuD2FLy",
        "name": "Default template for Projects",
        "description": "Template for new projects",
        "createdAt": "2001-09-09T03:46:40+0200",
        "updatedAt": "2001-09-09T03:46:40+0200",
        "permissions": [
          {
            "key": "admin",
            "usersCount": 0,
            "groupsCount": 1,
            "withProjectCreator": true
          },
          {
            "key": "codeviewer",
            "usersCount": 1,
            "groupsCount": 0,
            "withProjectCreator": false
          },
          {
            "key": "issueadmin",
            "usersCount": 3,
            "groupsCount": 0,
            "withProjectCreator": false
          },
          {
            "key": "scan",
            "usersCount": 0,
            "groupsCount": 0,
            "withProjectCreator": false
          },
          {
            "key": "user",
            "usersCount": 0,
```

```
      "groupsCount": 0,
      "withProjectCreator": false
    }
   ]
  },
  {
   "id": "AU-TpxcA-iU5OvuD2FLz",
   "name": "Default template for Views",
   "description": "Template for new views",
   "projectKeyPattern": ".*sonar.views.*",
   "createdAt": "2001-09-09T03:46:40+0200",
   "updatedAt": "2004-11-09T12:33:20+0100",
   "permissions": [
    {
      "key": "admin",
      "usersCount": 0,
      "groupsCount": 0,
      "withProjectCreator": false
    },
    {
      "key": "codeviewer",
      "usersCount": 0,
      "groupsCount": 0,
      "withProjectCreator": false
    },
    {
      "key": "issueadmin",
      "usersCount": 0,
      "groupsCount": 3,
      "withProjectCreator": false
    },
    {
      "key": "scan",
      "usersCount": 0,
      "groupsCount": 0,
      "withProjectCreator": false
    },
    {
      "key": "user",
      "usersCount": 2,
      "groupsCount": 0,
      "withProjectCreator": false
    }
   ]
  }
 ],
 "defaultTemplates": [
  {
```

```
    "templateId": "AU-Tpxb--iU5OvuD2FLy",
    "qualifier": "TRK"
  },
  {
    "templateId": "AU-TpxcA-iU5OvuD2FLz",
    "qualifier": "VW"
  }
 ]
}
{
 "permissionTemplate": {
  "name": "Finance",
  "description": "Permissions for financially related projects",
  "projectKeyPattern": ".*\\.finance\\..*"
 }
}
{
 "paging": {
  "pageIndex": 1,
  "pageSize": 25,
  "total": 3
 },
 "projects": [
  {
   "id": "0bd7b1e7-91d6-439e-a607-4a3a9aad3c6a",
   "key": "net.java.openjdk:jdk7",
   "name": "JDK 7",
   "qualifier": "TRK",
   "permissions": [
    {
     "key": "admin",
     "usersCount": 3,
     "groupsCount": 4
    },
    {
     "key": "issueadmin",
     "usersCount": 1,
     "groupsCount": 0
    }
   ]
  },
  {
   "id": "ce4c03d6-430f-40a9-b777-ad877c00aa4d",
   "key": "clang",
   "name": "Clang",
   "qualifier": "TRK",
   "permissions": [
    {
```

```
        "key": "issueadmin",
        "usersCount": 1,
        "groupsCount": 0
      }
    ]
  },
  {
    "id": "752d8bfd-420c-4a83-a4e5-8ab19b13c8fc",
    "key": "Java",
    "name": "Java",
    "qualifier": "VW",
    "permissions": [
      {
        "key": "admin",
        "usersCount": 0,
        "groupsCount": 1
      },
      {
        "key": "issueadmin",
        "usersCount": 1,
        "groupsCount": 0
      }
    ]
  }
],
"permissions": [
  {
    "key": "user",
    "name": "Browse",
    "description": "Ability to access a project, browse its measures, and create/edit issues for it."
  },
  {
    "key": "admin",
    "name": "Administer",
    "description": "Ability to access project settings and perform administration tasks. (Users will also need
\"Browse\" permission)"
  },
  {
    "key": "issueadmin",
    "name": "Administer Issues",
    "description": "Grants the permission to perform advanced editing on issues: marking an issue False Positive /
Won\u0027t Fix or changing an Issue\u0027s severity. (Users will also need \"Browse\" permission)"
  },
  {
    "key": "codeviewer",
    "name": "See Source Code",
    "description": "Ability to view the project\u0027s source code. (Users will also need \"Browse\" permission)"
  }
```

```
        ]
      }
      {
        "permissions": [
          {
            "key": "admin",
            "name": "Administer System",
            "description": "Ability to perform all administration functions for the instance: global configuration and
personalization of default dashboards.",
            "usersCount": 0,
            "groupsCount": 1
          },
          {
            "key": "profileadmin",
            "name": "Administer Quality Profiles",
            "description": "Ability to perform any action on the quality profiles.",
            "usersCount": 2,
            "groupsCount": 0
          },
          {
            "key": "gateadmin",
            "name": "Administer Quality Gates",
            "description": "Ability to perform any action on the quality gates.",
            "usersCount": 2,
            "groupsCount": 0
          },
          {
            "key": "scan",
            "name": "Execute Analysis",
            "description": "Ability to execute analyses, and to get all settings required to perform the analysis, even the
secured ones like the scm account password, the jira account password, and so on.",
            "usersCount": 0,
            "groupsCount": 2
          },
          {
            "key": "provisioning",
            "name": "Create Projects",
            "description": "Ability to initialize project structure before first analysis.",
            "usersCount": 1,
            "groupsCount": 1
          }
        ]
      }
      {
        "paging": {
          "pageIndex": 1,
          "pageSize": 20,
          "total": 2
```

```
  },
  "users": [
    {
      "login": "admin",
      "name": "Administrator",
      "email": "admin@admin.com",
      "avatar": "64e1b8d34f425d19e1ee2ea7236d3028",
      "permissions": ["admin", "gateadmin", "profileadmin"]
    },
    {
      "login": "george.orwell",
      "name": "George Orwell",
      "email": "george.orwell@1984.net",
      "avatar": "583af86a274c1027ef078cada831babf",
      "permissions": ["scan"]
    }
  ]
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
@ParametersAreNonnullByDefault
package org.sonar.server.permission;

import javax.annotation.ParametersAreNonnullByDefault;
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
```

package org.sonar.server.permission;

import javax.annotation.Nullable;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;

public class GroupPermissionChange extends PermissionChange {

 private final GroupIdOrAnyone groupId;

 public GroupPermissionChange(Operation operation, String permission, @Nullable ProjectId projectId,
   GroupIdOrAnyone groupId) {
   super(operation, groupId.getOrganizationUuid(), permission, projectId);
   this.groupId = groupId;
 }

 public GroupIdOrAnyone getGroupIdOrAnyone() {
   return groupId;
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,

package org.sonar.server.permission.ws;

import java.util.Optional;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.server.permission.PermissionChange;
import org.sonar.server.permission.PermissionUpdater;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.permission.UserId;
import org.sonar.server.permission.UserPermissionChange;
import org.sonar.server.user.UserSession;

import static java.util.Arrays.asList;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkProjectAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createUserLoginParameter;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;

```java
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class RemoveUserAction implements PermissionsWsAction {

 public static final String ACTION = "remove_user";

 private final DbClient dbClient;
 private final UserSession userSession;
 private final PermissionUpdater permissionUpdater;
 private final PermissionWsSupport support;

 public RemoveUserAction(DbClient dbClient, UserSession userSession, PermissionUpdater permissionUpdater,
PermissionWsSupport support) {
  this.dbClient = dbClient;
  this.userSession = userSession;
  this.permissionUpdater = permissionUpdater;
  this.support = support;
 }

 @Override
 public void define(WebService.NewController context) {
  WebService.NewAction action = context.createAction(ACTION)
   .setDescription("Remove permission from a user.<br /> " +
    "This service defaults to global permissions, but can be limited to project permissions by providing project id or
project key.<br /> " +
    "Requires one of the following permissions:" +
    "<ul>" +
    "<li>'Administer System'</li>" +
    "<li>'Administer' rights on the specified project</li>" +
    "</ul>")
   .setSince("5.2")
   .setPost(true)
   .setHandler(this);

  createPermissionParameter(action);
  createUserLoginParameter(action);
  createProjectParameters(action);
  createOrganizationParameter(action).setSince("6.2");
 }

 @Override
 public void handle(Request request, Response response) throws Exception {
  try (DbSession dbSession = dbClient.openSession(false)) {
   UserId user = support.findUser(dbSession, request.mandatoryParam(PARAM_USER_LOGIN));
   Optional<ProjectId> projectId = support.findProjectId(dbSession, request);
   OrganizationDto org = support.findOrganization(dbSession, request.param(PARAM_ORGANIZATION));
```

```java
      checkProjectAdmin(userSession, org.getUuid(), projectId);

      PermissionChange change = new UserPermissionChange(
        PermissionChange.Operation.REMOVE,
        org.getUuid(),
        request.mandatoryParam(PARAM_PERMISSION),
        projectId.orElse(null),
        user);
      permissionUpdater.apply(dbSession, asList(change));
      response.noContent();
    }
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.sonar.api.server.ws.WebService.NewAction;
import org.sonar.api.server.ws.WebService.NewParam;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.core.util.Uuids;

import static java.lang.String.format;
import static org.sonar.server.ws.KeyExamples.KEY_PROJECT_EXAMPLE_001;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_DESCRIPTION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
```

```java
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY_PATTERN;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class PermissionsWsParametersBuilder {

  private static final String PERMISSION_PARAM_DESCRIPTION = format("Permission" +
    "<ul>" +
    "<li>Possible values for global permissions: %s</li>" +
    "<li>Possible values for project permissions %s</li>" +
    "</ul>",
    GlobalPermissions.ALL_ON_ONE_LINE,
    ProjectPermissions.ALL_ON_ONE_LINE);
  public static final String PROJECT_PERMISSION_PARAM_DESCRIPTION = format("Permission" +
    "<ul>" +
    "<li>Possible values for project permissions %s</li>" +
    "</ul>",
    ProjectPermissions.ALL_ON_ONE_LINE);

  private PermissionsWsParametersBuilder() {
    // static methods only
  }

  public static NewParam createPermissionParameter(NewAction action) {
    return action.createParam(PARAM_PERMISSION)
      .setDescription(PERMISSION_PARAM_DESCRIPTION)
      .setRequired(true);
  }

  public static NewParam createProjectPermissionParameter(NewAction action, boolean required) {
    return action.createParam(PARAM_PERMISSION)
      .setDescription(PROJECT_PERMISSION_PARAM_DESCRIPTION)
      .setPossibleValues(ProjectPermissions.ALL)
      .setRequired(required);
  }

  public static NewParam createProjectPermissionParameter(NewAction action) {
    return createProjectPermissionParameter(action, true);
  }

  public static void createGroupNameParameter(NewAction action) {
    action.createParam(PARAM_GROUP_NAME)
      .setDescription("Group name or 'anyone' (case insensitive)")
      .setExampleValue("sonar-administrators");
  }
```

```java
public static NewParam createOrganizationParameter(NewAction action) {
  return action.createParam(PARAM_ORGANIZATION)
    .setDescription("Key of organization, used when group name is set")
    .setExampleValue("my-org")
    .setInternal(true);
}

public static void createGroupIdParameter(NewAction action) {
  action.createParam(PARAM_GROUP_ID)
    .setDescription("Group id")
    .setExampleValue("42");
}

public static void createProjectParameters(NewAction action) {
  createProjectIdParameter(action);
  createProjectKeyParameter(action);
}

private static void createProjectIdParameter(NewAction action) {
  action.createParam(PARAM_PROJECT_ID)
    .setDescription("Project id")
    .setExampleValue("ce4c03d6-430f-40a9-b777-ad877c00aa4d");
}

private static void createProjectKeyParameter(NewAction action) {
  action.createParam(PARAM_PROJECT_KEY)
    .setDescription("Project key")
    .setExampleValue(KEY_PROJECT_EXAMPLE_001);
}

public static void createUserLoginParameter(NewAction action) {
  action.createParam(PARAM_USER_LOGIN)
    .setRequired(true)
    .setDescription("User login")
    .setExampleValue("g.hopper");
}

public static void createTemplateParameters(NewAction action) {
  createTemplateIdParameter(action);
  createOrganizationParameter(action);
  createTemplateNameParameter(action);
}

private static void createTemplateIdParameter(NewAction action) {
  action.createParam(PARAM_TEMPLATE_ID)
    .setDescription("Template id")
    .setExampleValue(Uuids.UUID_EXAMPLE_01);
```

```java
  }

  private static void createTemplateNameParameter(NewAction action) {
    action.createParam(PARAM_TEMPLATE_NAME)
      .setDescription("Template name")
      .setExampleValue("Default Permission Template for Projects");
  }

  public static void createTemplateProjectKeyPatternParameter(NewAction action) {
    action.createParam(PARAM_PROJECT_KEY_PATTERN)
      .setDescription("Project key pattern. Must be a valid Java regular expression")
      .setExampleValue(".*\\.finance\\..*");
  }

  public static void createTemplateDescriptionParameter(NewAction action) {
    action.createParam(PARAM_DESCRIPTION)
      .setDescription("Description")
      .setExampleValue("Permissions for all projects related to the financial service");
  }

  public static void createIdParameter(NewAction action) {
    action.createParam(PARAM_ID)
      .setRequired(true)
      .setDescription("Id")
      .setExampleValue("af8cb8cc-1e78-4c4e-8c00-ee8e814009a5");
  }
}
```

```java
@ParametersAreNonnullByDefault
package org.sonar.server.permission.ws.template;
```

import javax.annotation.ParametersAreNonnullByDefault;
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
*/
package org.sonar.server.permission.ws.template;

import java.util.Collection;
import java.util.HashSet;
import java.util.List;
import org.sonar.api.i18n.I18n;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.resources.ResourceTypes;
import org.sonar.api.server.ws.Change;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.db.DatabaseUtils;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.PermissionTemplateService;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.project.Visibility;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

```java
import static java.util.Collections.singleton;
import static java.util.Objects.requireNonNull;
import static java.lang.String.format;
import static org.sonar.api.utils.DateUtils.parseDateOrDateTime;
import static org.sonar.core.util.Protobuf.setNullable;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.permission.ws.template.WsTemplateRef.newTemplateRef;
import static org.sonar.server.ws.KeyExamples.KEY_PROJECT_EXAMPLE_001;
import static org.sonar.server.ws.KeyExamples.KEY_PROJECT_EXAMPLE_002;
import static org.sonar.server.ws.WsParameterBuilder.QualifierParameterContext.newQualifierParameterContext;
import static org.sonar.server.ws.WsParameterBuilder.createRootQualifiersParameter;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_QUALIFIER;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_ANALYZED_BEFORE;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_ON_PROVISIONED_ONLY;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_PROJECTS;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_QUALIFIERS;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_VISIBILITY;

public class BulkApplyTemplateAction implements PermissionsWsAction {

  private final DbClient dbClient;
  private final UserSession userSession;
  private final PermissionTemplateService permissionTemplateService;
  private final PermissionWsSupport wsSupport;
  private final I18n i18n;
  private final ResourceTypes resourceTypes;

  public BulkApplyTemplateAction(DbClient dbClient, UserSession userSession, PermissionTemplateService
permissionTemplateService, PermissionWsSupport wsSupport, I18n i18n,
    ResourceTypes resourceTypes) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.permissionTemplateService = permissionTemplateService;
    this.wsSupport = wsSupport;
    this.i18n = i18n;
    this.resourceTypes = resourceTypes;
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("bulk_apply_template")
      .setDescription("Apply a permission template to several projects.<br />" +
        "The template id or name must be provided.<br />" +
```

```
      "Requires the following permission: 'Administer System'.")
    .setPost(true)
    .setSince("5.5")
    .setChangelog(new Change("6.7.2", format("Parameter %s accepts maximum %d values", PARAM_PROJECTS,
DatabaseUtils.PARTITION_SIZE_FOR_ORACLE)))
    .setHandler(this);

  action.createParam(Param.TEXT_QUERY)
    .setDescription("Limit search to: <ul>" +
      "<li>project names that contain the supplied string</li>" +
      "<li>project keys that are exactly the same as the supplied string</li>" +
      "</ul>")
    .setExampleValue("apac");

  createRootQualifiersParameter(action, newQualifierParameterContext(i18n, resourceTypes))
    .setDefaultValue(Qualifiers.PROJECT)
    .setDeprecatedKey(PARAM_QUALIFIER, "6.6");

  createTemplateParameters(action);

  action
    .createParam(PARAM_PROJECTS)
    .setDescription("Comma-separated list of project keys")
    .setSince("6.6")
    // Limitation of ComponentDao#selectByQuery(), max 1000 values are accepted.
    // Restricting size of HTTP parameter allows to not fail with SQL error
    .setMaxValuesAllowed(DatabaseUtils.PARTITION_SIZE_FOR_ORACLE)
    .setExampleValue(String.join(",", KEY_PROJECT_EXAMPLE_001, KEY_PROJECT_EXAMPLE_002));

  action.createParam(PARAM_VISIBILITY)
    .setDescription("Filter the projects that should be visible to everyone (%s), or only specific user/groups
(%s).<br/>" +
      "If no visibility is specified, the default project visibility of the organization will be used.",
      Visibility.PUBLIC.getLabel(), Visibility.PRIVATE.getLabel())
    .setRequired(false)
    .setInternal(true)
    .setSince("6.6")
    .setPossibleValues(Visibility.getLabels());

  action.createParam(PARAM_ANALYZED_BEFORE)
    .setDescription("Filter the projects for which last analysis is older than the given date (exclusive).<br> " +
      "Either a date (server timezone) or datetime can be provided.")
    .setSince("6.6")
    .setExampleValue("2017-10-19 or 2017-10-19T13:00:00+0200")
    ;

  action.createParam(PARAM_ON_PROVISIONED_ONLY)
    .setDescription("Filter the projects that are provisioned")
```

```java
        .setBooleanPossibleValues()
        .setDefaultValue("false")
        .setSince("6.6");
    }

    @Override
    public void handle(Request request, Response response) throws Exception {
      doHandle(toBulkApplyTemplateWsRequest(request));
      response.noContent();
    }

    private void doHandle(BulkApplyTemplateRequest request) {
      try (DbSession dbSession = dbClient.openSession(false)) {
        PermissionTemplateDto template = wsSupport.findTemplate(dbSession, newTemplateRef(
          request.getTemplateId(), request.getOrganization(), request.getTemplateName()));
        checkGlobalAdmin(userSession, template.getOrganizationUuid());

        ComponentQuery componentQuery = buildDbQuery(request);
        List<ComponentDto> projects = dbClient.componentDao().selectByQuery(dbSession,
template.getOrganizationUuid(), componentQuery, 0, Integer.MAX_VALUE);

        permissionTemplateService.applyAndCommit(dbSession, template, projects);
      }
    }

    private static BulkApplyTemplateRequest toBulkApplyTemplateWsRequest(Request request) {
      return new BulkApplyTemplateRequest()
        .setOrganization(request.param(PARAM_ORGANIZATION))
        .setTemplateId(request.param(PARAM_TEMPLATE_ID))
        .setTemplateName(request.param(PARAM_TEMPLATE_NAME))
        .setQualifiers(request.mandatoryParamAsStrings(PARAM_QUALIFIERS))
        .setQuery(request.param(Param.TEXT_QUERY))
        .setVisibility(request.param(PARAM_VISIBILITY))
        .setOnProvisionedOnly(request.mandatoryParamAsBoolean(PARAM_ON_PROVISIONED_ONLY))
        .setAnalyzedBefore(request.param(PARAM_ANALYZED_BEFORE))
        .setProjects(request.paramAsStrings(PARAM_PROJECTS));
    }

    private static ComponentQuery buildDbQuery(BulkApplyTemplateRequest request) {
      Collection<String> qualifiers = request.getQualifiers();
      ComponentQuery.Builder query = ComponentQuery.builder()
        .setQualifiers(qualifiers.toArray(new String[qualifiers.size()]));

      setNullable(request.getQuery(), q -> {
        query.setNameOrKeyQuery(q);
        query.setPartialMatchOnKey(true);
        return query;
      });
```

```java
      setNullable(request.getVisibility(), v -> query.setPrivate(Visibility.isPrivate(v)));
      setNullable(request.getAnalyzedBefore(), d -> query.setAnalyzedBefore(parseDateOrDateTime(d).getTime()));
      setNullable(request.isOnProvisionedOnly(), query::setOnProvisionedOnly);
      setNullable(request.getProjects(), keys -> query.setComponentKeys(new HashSet<>(keys)));

      return query.build();
    }

    private static class BulkApplyTemplateRequest {
      private String templateId;
      private String organization;
      private String templateName;
      private String query;
      private Collection<String> qualifiers = singleton(Qualifiers.PROJECT);
      private String visibility;
      private String analyzedBefore;
      private boolean onProvisionedOnly = false;
      private Collection<String> projects;

      @CheckForNull
      public String getTemplateId() {
        return templateId;
      }

      public BulkApplyTemplateRequest setTemplateId(@Nullable String templateId) {
        this.templateId = templateId;
        return this;
      }

      @CheckForNull
      public String getOrganization() {
        return organization;
      }

      public BulkApplyTemplateRequest setOrganization(@Nullable String s) {
        this.organization = s;
        return this;
      }

      @CheckForNull
      public String getTemplateName() {
        return templateName;
      }

      public BulkApplyTemplateRequest setTemplateName(@Nullable String templateName) {
        this.templateName = templateName;
        return this;
      }
```

```java
@CheckForNull
public String getQuery() {
  return query;
}

public BulkApplyTemplateRequest setQuery(@Nullable String query) {
  this.query = query;
  return this;
}

public Collection<String> getQualifiers() {
  return qualifiers;
}

public BulkApplyTemplateRequest setQualifiers(Collection<String> qualifiers) {
  this.qualifiers = requireNonNull(qualifiers);
  return this;
}

@CheckForNull
public String getVisibility() {
  return visibility;
}

public BulkApplyTemplateRequest setVisibility(@Nullable String visibility) {
  this.visibility = visibility;
  return this;
}

@CheckForNull
public String getAnalyzedBefore() {
  return analyzedBefore;
}

public BulkApplyTemplateRequest setAnalyzedBefore(@Nullable String analyzedBefore) {
  this.analyzedBefore = analyzedBefore;
  return this;
}

public boolean isOnProvisionedOnly() {
  return onProvisionedOnly;
}

public BulkApplyTemplateRequest setOnProvisionedOnly(boolean onProvisionedOnly) {
  this.onProvisionedOnly = onProvisionedOnly;
  return this;
}
```

```java
    @CheckForNull
    public Collection<String> getProjects() {
      return projects;
    }

    public BulkApplyTemplateRequest setProjects(@Nullable Collection<String> projects) {
      this.projects = projects;
      return this;
    }
  }
}
```

```java
package org.sonar.server.permission.ws.template;

import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.utils.System2;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDao;
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
```

```java
import static java.util.Objects.requireNonNull;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPermission;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class RemoveProjectCreatorFromTemplateAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final PermissionWsSupport wsSupport;
  private final UserSession userSession;
  private final System2 system;

  public RemoveProjectCreatorFromTemplateAction(DbClient dbClient, PermissionWsSupport wsSupport,
UserSession userSession, System2 system) {
    this.dbClient = dbClient;
    this.wsSupport = wsSupport;
    this.userSession = userSession;
    this.system = system;
  }

  private static RemoveProjectCreatorFromTemplateRequest toWsRequest(Request request) {
    RemoveProjectCreatorFromTemplateRequest wsRequest =
RemoveProjectCreatorFromTemplateRequest.builder()
      .setPermission(request.mandatoryParam(PARAM_PERMISSION))
      .setTemplateId(request.param(PARAM_TEMPLATE_ID))
      .setOrganization(request.param(PARAM_ORGANIZATION))
      .setTemplateName(request.param(PARAM_TEMPLATE_NAME))
      .build();
    validateProjectPermission(wsRequest.getPermission());
    return wsRequest;
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("remove_project_creator_from_template")
      .setDescription("Remove a project creator from a permission template.<br>" +
        "Requires the following permission: 'Administer System'.")
      .setSince("6.0")
      .setPost(true)
      .setHandler(this);

    createTemplateParameters(action);
    createProjectPermissionParameter(action);
```

```java
    }

    @Override
    public void handle(Request request, Response response) throws Exception {
      doHandle(toWsRequest(request));
      response.noContent();
    }

    private void doHandle(RemoveProjectCreatorFromTemplateRequest request) {
      try (DbSession dbSession = dbClient.openSession(false)) {
        PermissionTemplateDto template = wsSupport.findTemplate(dbSession, WsTemplateRef.newTemplateRef(
          request.getTemplateId(), request.getOrganization(), request.getTemplateName()));
        checkGlobalAdmin(userSession, template.getOrganizationUuid());

        PermissionTemplateCharacteristicDao dao = dbClient.permissionTemplateCharacteristicDao();
        dao.selectByPermissionAndTemplateId(dbSession, request.getPermission(), template.getId())
          .ifPresent(permissionTemplateCharacteristicDto -> updateTemplateCharacteristic(dbSession,
permissionTemplateCharacteristicDto));
      }
    }

    private void updateTemplateCharacteristic(DbSession dbSession, PermissionTemplateCharacteristicDto
templatePermission) {
      PermissionTemplateCharacteristicDto targetTemplatePermission = templatePermission
        .setUpdatedAt(system.now())
        .setWithProjectCreator(false);
      dbClient.permissionTemplateCharacteristicDao().update(dbSession, targetTemplatePermission);
      dbSession.commit();
    }

    private static class RemoveProjectCreatorFromTemplateRequest {
      private final String templateId;
      private final String organization;
      private final String templateName;
      private final String permission;

      private RemoveProjectCreatorFromTemplateRequest(Builder builder) {
        this.templateId = builder.templateId;
        this.organization = builder.organization;
        this.templateName = builder.templateName;
        this.permission = requireNonNull(builder.permission);
      }

      @CheckForNull
      public String getTemplateId() {
        return templateId;
      }
```

```java
    @CheckForNull
    public String getOrganization() {
      return organization;
    }

    @CheckForNull
    public String getTemplateName() {
      return templateName;
    }

    public String getPermission() {
      return permission;
    }

    public static Builder builder() {
      return new Builder();
    }
  }

  public static class Builder {
   private String templateId;
   private String organization;
   private String templateName;
   private String permission;

   private Builder() {
     // enforce method constructor
   }

   public Builder setTemplateId(@Nullable String templateId) {
     this.templateId = templateId;
     return this;
   }

   public Builder setOrganization(@Nullable String s) {
     this.organization = s;
     return this;
   }

   public Builder setTemplateName(@Nullable String templateName) {
     this.templateName = templateName;
     return this;
   }

   public Builder setPermission(@Nullable String permission) {
     this.permission = permission;
     return this;
   }
```

```java
    public RemoveProjectCreatorFromTemplateRequest build() {
      return new RemoveProjectCreatorFromTemplateRequest(this);
    }
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.DefaultTemplates;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.permission.ws.template.WsTemplateRef.newTemplateRef;
import static org.sonar.server.ws.WsUtils.checkFoundWithOptional;
import static org.sonar.server.ws.WsUtils.checkRequest;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
```

```java
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class DeleteTemplateAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final UserSession userSession;
  private final PermissionWsSupport finder;
  private final DefaultTemplatesResolver defaultTemplatesResolver;

  public DeleteTemplateAction(DbClient dbClient, UserSession userSession, PermissionWsSupport support,
DefaultTemplatesResolver defaultTemplatesResolver) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.finder = support;
    this.defaultTemplatesResolver = defaultTemplatesResolver;
  }

  private static DeleteTemplateRequest toDeleteTemplateWsRequest(Request request) {
    return new DeleteTemplateRequest()
      .setTemplateId(request.param(PARAM_TEMPLATE_ID))
      .setOrganization(request.param(PARAM_ORGANIZATION))
      .setTemplateName(request.param(PARAM_TEMPLATE_NAME));
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("delete_template")
      .setDescription("Delete a permission template.<br />" +
        "Requires the following permission: 'Administer System'.")
      .setSince("5.2")
      .setPost(true)
      .setHandler(this);

    createTemplateParameters(action);
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    userSession.checkLoggedIn();
    doHandle(toDeleteTemplateWsRequest(request));
    response.noContent();
  }

  private void doHandle(DeleteTemplateRequest request) {
    try (DbSession dbSession = dbClient.openSession(false)) {
      PermissionTemplateDto template = finder.findTemplate(dbSession, newTemplateRef(
        request.getTemplateId(), request.getOrganization(), request.getTemplateName()));
      checkGlobalAdmin(userSession, template.getOrganizationUuid());
```

```java
    DefaultTemplates defaultTemplates = retrieveDefaultTemplates(dbSession, template);

    checkTemplateUuidIsNotDefault(template, defaultTemplates);
    dbClient.permissionTemplateDao().deleteById(dbSession, template.getId());
    updateViewDefaultTemplateWhenGovernanceIsNotInstalled(dbSession, template, defaultTemplates);

    dbSession.commit();
  }
}

/**
 * The default template for view can be removed when Governance is not installed. To avoid keeping a reference
 * to a non existing template, we update the default templates.
 */
private void updateViewDefaultTemplateWhenGovernanceIsNotInstalled(DbSession dbSession,
PermissionTemplateDto template, DefaultTemplates defaultTemplates) {
  String viewDefaultTemplateUuid = defaultTemplates.getViewUuid();
  if (viewDefaultTemplateUuid != null && viewDefaultTemplateUuid.equals(template.getUuid())) {
    defaultTemplates.setViewUuid(null);
    dbClient.organizationDao().setDefaultTemplates(dbSession, template.getOrganizationUuid(), defaultTemplates);
  }
}

private DefaultTemplates retrieveDefaultTemplates(DbSession dbSession, PermissionTemplateDto template) {
  return checkFoundWithOptional(
    dbClient.organizationDao().getDefaultTemplates(dbSession, template.getOrganizationUuid()),
    "Can't find default templates of Organization with uuid '%s' to which template with uuid '%s' belongs",
    template.getOrganizationUuid(), template.getUuid());
}

private void checkTemplateUuidIsNotDefault(PermissionTemplateDto template, DefaultTemplates
defaultTemplates) {
  DefaultTemplatesResolverImpl.ResolvedDefaultTemplates resolvedDefaultTemplates =
defaultTemplatesResolver.resolve(defaultTemplates);
  checkRequest(!resolvedDefaultTemplates.getProject().equals(template.getUuid()),
    "It is not possible to delete the default permission template for projects");
  resolvedDefaultTemplates.getView()
    .ifPresent(viewDefaultTemplateUuid -> checkRequest(
      !viewDefaultTemplateUuid.equals(template.getUuid()),
      "It is not possible to delete the default permission template for views"));
}

private static class DeleteTemplateRequest {
  private String templateId;
  private String organization;
  private String templateName;

  @CheckForNull
```

```java
  public String getTemplateId() {
    return templateId;
  }

  public DeleteTemplateRequest setTemplateId(@Nullable String templateId) {
    this.templateId = templateId;
    return this;
  }

  @CheckForNull
  public String getOrganization() {
    return organization;
  }

  public DeleteTemplateRequest setOrganization(@Nullable String s) {
    this.organization = s;
    return this;
  }

  @CheckForNull
  public String getTemplateName() {
    return templateName;
  }

  public DeleteTemplateRequest setTemplateName(@Nullable String templateName) {
    this.templateName = templateName;
    return this;
  }
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
```

```java
*/
package org.sonar.server.permission.ws.template;

import java.util.Date;
import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.utils.System2;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions.PermissionTemplate;
import org.sonarqube.ws.Permissions.UpdateTemplateWsResponse;

import static com.google.common.base.MoreObjects.firstNonNull;
import static java.lang.String.format;
import static java.util.Objects.requireNonNull;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static
org.sonar.server.permission.ws.PermissionRequestValidator.MSG_TEMPLATE_WITH_SAME_NAME;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPattern;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateTemplateNameFormat;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createIdParameter;
import static
org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateDescriptionParameter;
import static
org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateProjectKeyPatternParameter;
import static
org.sonar.server.permission.ws.template.PermissionTemplateDtoToPermissionTemplateResponse.toPermissionTem
plateResponse;
import static org.sonar.server.ws.WsUtils.checkRequest;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_DESCRIPTION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY_PATTERN;

public class UpdateTemplateAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final UserSession userSession;
  private final System2 system;
  private final PermissionWsSupport wsSupport;
```

```java
  public UpdateTemplateAction(DbClient dbClient, UserSession userSession, System2 system,
PermissionWsSupport wsSupport) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.system = system;
    this.wsSupport = wsSupport;
  }

  private static UpdateTemplateRequest toUpdateTemplateWsRequest(Request request) {
    return new UpdateTemplateRequest()
      .setId(request.mandatoryParam(PARAM_ID))
      .setName(request.param(PARAM_NAME))
      .setDescription(request.param(PARAM_DESCRIPTION))
      .setProjectKeyPattern(request.param(PARAM_PROJECT_KEY_PATTERN));
  }

  private static UpdateTemplateWsResponse buildResponse(PermissionTemplateDto permissionTemplate) {
    PermissionTemplate permissionTemplateBuilder = toPermissionTemplateResponse(permissionTemplate);
    return UpdateTemplateWsResponse.newBuilder().setPermissionTemplate(permissionTemplateBuilder).build();
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("update_template")
      .setDescription("Update a permission template.<br />" +
        "Requires the following permission: 'Administer System'.")
      .setResponseExample(getClass().getResource("update_template-example.json"))
      .setSince("5.2")
      .setPost(true)
      .setHandler(this);

    createIdParameter(action);

    action.createParam(PARAM_NAME)
      .setDescription("Name")
      .setExampleValue("Financial Service Permissions");

    createTemplateProjectKeyPatternParameter(action);
    createTemplateDescriptionParameter(action);
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    UpdateTemplateWsResponse updateTemplateWsResponse = doHandle(toUpdateTemplateWsRequest(request));
    writeProtobuf(updateTemplateWsResponse, request, response);
  }

  private UpdateTemplateWsResponse doHandle(UpdateTemplateRequest request) {
```

```java
    String uuid = request.getId();
    String nameParam = request.getName();
    String descriptionParam = request.getDescription();
    String projectPatternParam = request.getProjectKeyPattern();

    try (DbSession dbSession = dbClient.openSession(false)) {
      PermissionTemplateDto templateToUpdate = getAndBuildTemplateToUpdate(dbSession, uuid, nameParam,
descriptionParam, projectPatternParam);
      checkGlobalAdmin(userSession, templateToUpdate.getOrganizationUuid());

      validateTemplate(dbSession, templateToUpdate);
      PermissionTemplateDto updatedTemplate = updateTemplate(dbSession, templateToUpdate);
      dbSession.commit();

      return buildResponse(updatedTemplate);
    }
  }

  private void validateTemplate(DbSession dbSession, PermissionTemplateDto templateToUpdate) {
    validateTemplateNameForUpdate(dbSession, templateToUpdate.getOrganizationUuid(),
templateToUpdate.getName(), templateToUpdate.getId());
    validateProjectPattern(templateToUpdate.getKeyPattern());
  }

  private PermissionTemplateDto getAndBuildTemplateToUpdate(DbSession dbSession, String uuid, @Nullable
String newName, @Nullable String newDescription,
    @Nullable String newProjectKeyPattern) {
    PermissionTemplateDto templateToUpdate = wsSupport.findTemplate(dbSession,
WsTemplateRef.newTemplateRef(uuid, null, null));
    templateToUpdate.setName(firstNonNull(newName, templateToUpdate.getName()));
    templateToUpdate.setDescription(firstNonNull(newDescription, templateToUpdate.getDescription()));
    templateToUpdate.setKeyPattern(firstNonNull(newProjectKeyPattern, templateToUpdate.getKeyPattern()));
    templateToUpdate.setUpdatedAt(new Date(system.now()));

    return templateToUpdate;
  }

  private PermissionTemplateDto updateTemplate(DbSession dbSession, PermissionTemplateDto
templateToUpdate) {
    return dbClient.permissionTemplateDao().update(dbSession, templateToUpdate);
  }

  private void validateTemplateNameForUpdate(DbSession dbSession, String organizationUuid, String name, long
id) {
    validateTemplateNameFormat(name);

    PermissionTemplateDto permissionTemplateWithSameName =
dbClient.permissionTemplateDao().selectByName(dbSession, organizationUuid, name);
```

```java
  checkRequest(permissionTemplateWithSameName == null || permissionTemplateWithSameName.getId() == id,
    format(MSG_TEMPLATE_WITH_SAME_NAME, name));
}

private static class UpdateTemplateRequest {
 private String id;
 private String description;
 private String name;
 private String projectKeyPattern;

 public String getId() {
   return id;
 }

 public UpdateTemplateRequest setId(String id) {
  this.id = requireNonNull(id);
  return this;
 }

 @CheckForNull
 public String getDescription() {
  return description;
 }

 public UpdateTemplateRequest setDescription(@Nullable String description) {
  this.description = description;
  return this;
 }

 @CheckForNull
 public String getName() {
  return name;
 }

 public UpdateTemplateRequest setName(@Nullable String name) {
  this.name = name;
  return this;
 }

 @CheckForNull
 public String getProjectKeyPattern() {
  return projectKeyPattern;
 }

 public UpdateTemplateRequest setProjectKeyPattern(@Nullable String projectKeyPattern) {
  this.projectKeyPattern = projectKeyPattern;
  return this;
 }
```

```
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import java.util.Date;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.utils.System2;
import org.sonar.core.util.Uuids;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions.CreateTemplateWsResponse;
import org.sonarqube.ws.Permissions.PermissionTemplate;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static java.lang.String.format;
import static java.util.Objects.requireNonNull;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static
org.sonar.server.permission.ws.PermissionRequestValidator.MSG_TEMPLATE_WITH_SAME_NAME;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPattern;
```

```java
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateTemplateNameFormat;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static
org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateDescriptionParameter;
import static
org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateProjectKeyPatternParameter;
import static
org.sonar.server.permission.ws.template.PermissionTemplateDtoToPermissionTemplateResponse.toPermissionTem
plateResponse;
import static org.sonar.server.ws.WsUtils.checkRequest;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_DESCRIPTION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY_PATTERN;

public class CreateTemplateAction implements PermissionsWsAction {
 private final DbClient dbClient;
 private final UserSession userSession;
 private final System2 system;
 private final PermissionWsSupport wsSupport;

 public CreateTemplateAction(DbClient dbClient, UserSession userSession, System2 system, PermissionWsSupport
wsSupport) {
  this.dbClient = dbClient;
  this.userSession = userSession;
  this.system = system;
  this.wsSupport = wsSupport;
 }

 private static CreateTemplateRequest toCreateTemplateWsRequest(Request request) {
  return new CreateTemplateRequest()
   .setName(request.mandatoryParam(PARAM_NAME))
   .setDescription(request.param(PARAM_DESCRIPTION))
   .setProjectKeyPattern(request.param(PARAM_PROJECT_KEY_PATTERN))
   .setOrganization(request.param(PARAM_ORGANIZATION));
 }

 private static CreateTemplateWsResponse buildResponse(PermissionTemplateDto permissionTemplateDto) {
  PermissionTemplate permissionTemplateBuilder = toPermissionTemplateResponse(permissionTemplateDto);
  return CreateTemplateWsResponse.newBuilder().setPermissionTemplate(permissionTemplateBuilder).build();
 }

 @Override
 public void define(WebService.NewController context) {
  WebService.NewAction action = context.createAction("create_template")
   .setDescription("Create a permission template.<br />" +
    "Requires the following permission: 'Administer System'.")
```

```
      .setResponseExample(getClass().getResource("create_template-example.json"))
      .setSince("5.2")
      .setPost(true)
      .setHandler(this);

    action.createParam(PARAM_NAME)
      .setRequired(true)
      .setDescription("Name")
      .setExampleValue("Financial Service Permissions");

    createTemplateProjectKeyPatternParameter(action);
    createTemplateDescriptionParameter(action);
    createOrganizationParameter(action).setSince("6.2");
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    CreateTemplateWsResponse createTemplateWsResponse = doHandle(toCreateTemplateWsRequest(request));
    writeProtobuf(createTemplateWsResponse, request, response);
  }

  private CreateTemplateWsResponse doHandle(CreateTemplateRequest request) {
    try (DbSession dbSession = dbClient.openSession(false)) {
      OrganizationDto org = wsSupport.findOrganization(dbSession, request.getOrganization());
      checkGlobalAdmin(userSession, org.getUuid());

      validateTemplateNameForCreation(dbSession, org, request.getName());
      validateProjectPattern(request.getProjectKeyPattern());

      PermissionTemplateDto permissionTemplate = insertTemplate(dbSession, org, request);

      return buildResponse(permissionTemplate);
    }
  }

  private void validateTemplateNameForCreation(DbSession dbSession, OrganizationDto org, String name) {
    validateTemplateNameFormat(name);

    PermissionTemplateDto permissionTemplateWithSameName = dbClient.permissionTemplateDao()
      .selectByName(dbSession, org.getUuid(), name);
    checkRequest(permissionTemplateWithSameName == null, format(MSG_TEMPLATE_WITH_SAME_NAME,
name));
  }

  private PermissionTemplateDto insertTemplate(DbSession dbSession, OrganizationDto org,
CreateTemplateRequest request) {
    Date now = new Date(system.now());
    PermissionTemplateDto template = dbClient.permissionTemplateDao().insert(dbSession, new
```

```java
PermissionTemplateDto()
    .setUuid(Uuids.create())
    .setOrganizationUuid(org.getUuid())
    .setName(request.getName())
    .setDescription(request.getDescription())
    .setKeyPattern(request.getProjectKeyPattern())
    .setCreatedAt(now)
    .setUpdatedAt(now));
  dbSession.commit();
  return template;
}

private static class CreateTemplateRequest {
  private String description;
  private String name;
  private String projectKeyPattern;
  private String organization;

  @CheckForNull
  public String getDescription() {
    return description;
  }

  public CreateTemplateRequest setDescription(@Nullable String description) {
    this.description = description;
    return this;
  }

  public String getName() {
    return name;
  }

  public CreateTemplateRequest setName(String name) {
    this.name = requireNonNull(name);
    return this;
  }

  @CheckForNull
  public String getProjectKeyPattern() {
    return projectKeyPattern;
  }

  public CreateTemplateRequest setProjectKeyPattern(@Nullable String projectKeyPattern) {
    this.projectKeyPattern = projectKeyPattern;
    return this;
  }

  @CheckForNull
```

```java
  public String getOrganization() {
    return organization;
  }

  public CreateTemplateRequest setOrganization(@Nullable String s) {
    this.organization = s;
    return this;
  }
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;

import static java.lang.String.format;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupIdParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupNameParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
```

```java
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.permission.ws.template.WsTemplateRef.fromRequest;
import static org.sonar.server.ws.WsUtils.checkRequest;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;


public class AddGroupToTemplateAction implements PermissionsWsAction {
 private final DbClient dbClient;
 private final PermissionWsSupport support;
 private final UserSession userSession;

 public AddGroupToTemplateAction(DbClient dbClient, PermissionWsSupport support, UserSession userSession) {
  this.dbClient = dbClient;
  this.support = support;
  this.userSession = userSession;
 }

 @Override
 public void define(WebService.NewController context) {
  WebService.NewAction action = context
    .createAction("add_group_to_template")
    .setPost(true)
    .setSince("5.2")
    .setDescription("Add a group to a permission template.<br /> " +
      "The group id or group name must be provided. <br />" +
      "Requires the following permission: 'Administer System'.")
    .setHandler(this);

  createTemplateParameters(action);
  createProjectPermissionParameter(action);
  createGroupIdParameter(action);
  createGroupNameParameter(action);
 }

 @Override
 public void handle(Request request, Response response) {
  try (DbSession dbSession = dbClient.openSession(false)) {
   String permission = request.mandatoryParam(PARAM_PERMISSION);
   GroupIdOrAnyone groupId = support.findGroup(dbSession, request);
   checkRequest(!SYSTEM_ADMIN.equals(permission) || !groupId.isAnyone(),
     format("It is not possible to add the '%s' permission to the group 'Anyone'.", permission));

   PermissionTemplateDto template = support.findTemplate(dbSession, fromRequest(request));
   checkGlobalAdmin(userSession, template.getOrganizationUuid());

   if (!groupAlreadyAdded(dbSession, template.getId(), permission, groupId)) {
    dbClient.permissionTemplateDao().insertGroupPermission(dbSession, template.getId(), groupId.getId(),
permission);
     dbSession.commit();
```

```
        }
      }
      response.noContent();
    }

    private boolean groupAlreadyAdded(DbSession dbSession, long templateId, String permission, GroupIdOrAnyone
    group) {
      return dbClient.permissionTemplateDao().hasGroupsWithPermission(dbSession, templateId, permission,
    group.getId());
    }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import org.sonar.api.resources.Qualifiers;
import org.sonar.api.resources.ResourceType;
import org.sonar.api.resources.ResourceTypes;
import org.sonar.db.organization.DefaultTemplates;

import static java.util.Optional.ofNullable;

public class DefaultTemplatesResolverImpl implements DefaultTemplatesResolver {
  private final ResourceTypes resourceTypes;

  public DefaultTemplatesResolverImpl(ResourceTypes resourceTypes) {
    this.resourceTypes = resourceTypes;
  }

  @Override
  public ResolvedDefaultTemplates resolve(DefaultTemplates defaultTemplates) {
```

```java
    String projectDefaultTemplate = defaultTemplates.getProjectUuid();

    return new ResolvedDefaultTemplates(
      projectDefaultTemplate,
      isViewsEnabled(resourceTypes) ? ofNullable(defaultTemplates.getViewUuid()).orElse(projectDefaultTemplate)
: null);
  }

  private static boolean isViewsEnabled(ResourceTypes resourceTypes) {
    return resourceTypes.getRoots()
      .stream()
      .map(ResourceType::getQualifier)
      .anyMatch(Qualifiers.VIEW::equals);
  }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import com.google.common.collect.Multimap;
import com.google.common.collect.Ordering;
import com.google.common.collect.TreeMultimap;
import java.util.List;
import java.util.stream.Collectors;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.api.utils.Paging;
import org.sonar.db.DbClient;
```

```java
import org.sonar.db.DbSession;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.permission.template.PermissionTemplateUserDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.issue.ws.AvatarResolver;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions;
import org.sonarqube.ws.Permissions.UsersWsResponse;

import static com.google.common.base.Strings.emptyToNull;
import static org.sonar.api.server.ws.WebService.Param.PAGE;
import static org.sonar.api.server.ws.WebService.Param.PAGE_SIZE;
import static org.sonar.api.server.ws.WebService.Param.TEXT_QUERY;
import static org.sonar.core.util.Protobuf.setNullable;
import static org.sonar.db.permission.PermissionQuery.DEFAULT_PAGE_SIZE;
import static org.sonar.db.permission.PermissionQuery.RESULTS_MAX_SIZE;
import static org.sonar.db.permission.PermissionQuery.SEARCH_QUERY_MIN_LENGTH;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPermission;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;

public class TemplateUsersAction implements PermissionsWsAction {

  private final DbClient dbClient;
  private final UserSession userSession;
  private final PermissionWsSupport support;
  private final AvatarResolver avatarResolver;

  public TemplateUsersAction(DbClient dbClient, UserSession userSession, PermissionWsSupport support,
    AvatarResolver avatarResolver) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.support = support;
    this.avatarResolver = avatarResolver;
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context
      .createAction("template_users")
      .setSince("5.2")
      .setDescription("Lists the users with their permission as individual users rather than through group affiliation on
```

```
    the chosen template. <br />" +
      "This service defaults to all users, but can be limited to users with a specific permission by providing the desired
permission.<br>" +
      "Requires the following permission: 'Administer System'.")
    .addPagingParams(DEFAULT_PAGE_SIZE, RESULTS_MAX_SIZE)
    .setInternal(true)
    .setResponseExample(getClass().getResource("template_users-example.json"))
    .setHandler(this);

  action.createParam(Param.TEXT_QUERY)
    .setMinimumLength(SEARCH_QUERY_MIN_LENGTH)
    .setDescription("Limit search to user names that contain the supplied string. <br/>" +
      "When this parameter is not set, only users having at least one permission are returned.")
    .setExampleValue("eri");
  createProjectPermissionParameter(action).setRequired(false);
  createTemplateParameters(action);
}

@Override
public void handle(Request wsRequest, Response wsResponse) throws Exception {
  try (DbSession dbSession = dbClient.openSession(false)) {
    WsTemplateRef templateRef = WsTemplateRef.fromRequest(wsRequest);
    PermissionTemplateDto template = support.findTemplate(dbSession, templateRef);
    checkGlobalAdmin(userSession, template.getOrganizationUuid());

    PermissionQuery query = buildQuery(wsRequest, template);
    int total = dbClient.permissionTemplateDao().countUserLoginsByQueryAndTemplate(dbSession, query,
template.getId());
    Paging paging =
Paging.forPageIndex(wsRequest.mandatoryParamAsInt(PAGE)).withPageSize(wsRequest.mandatoryParamAsInt(P
AGE_SIZE)).andTotal(total);
    List<UserDto> users = findUsers(dbSession, query, template);
    List<PermissionTemplateUserDto> permissionTemplateUsers =
dbClient.permissionTemplateDao().selectUserPermissionsByTemplateIdAndUserLogins(dbSession,
template.getId(),
      users.stream().map(UserDto::getLogin).collect(Collectors.toList()));
    Permissions.UsersWsResponse templateUsersResponse = buildResponse(users, permissionTemplateUsers,
paging);
    writeProtobuf(templateUsersResponse, wsRequest, wsResponse);
  }
}

private static PermissionQuery buildQuery(Request wsRequest, PermissionTemplateDto template) {
  String textQuery = wsRequest.param(TEXT_QUERY);
  String permission = wsRequest.param(PARAM_PERMISSION);
  PermissionQuery.Builder query = PermissionQuery.builder()
    .setOrganizationUuid(template.getOrganizationUuid())
    .setTemplate(template.getUuid())
```

```java
      .setPermission(permission != null ? validateProjectPermission(permission) : null)
      .setPageIndex(wsRequest.mandatoryParamAsInt(PAGE))
      .setPageSize(wsRequest.mandatoryParamAsInt(PAGE_SIZE))
      .setSearchQuery(textQuery);
    if (textQuery == null) {
      query.withAtLeastOnePermission();
    }
    return query.build();
  }

  private Permissions.UsersWsResponse buildResponse(List<UserDto> users, List<PermissionTemplateUserDto>
permissionTemplateUsers, Paging paging) {
    Multimap<Integer, String> permissionsByUserId = TreeMultimap.create();
    permissionTemplateUsers.forEach(userPermission -> permissionsByUserId.put(userPermission.getUserId(),
userPermission.getPermission()));

    UsersWsResponse.Builder responseBuilder = UsersWsResponse.newBuilder();
    users.forEach(user -> {
      Permissions.User.Builder userResponse = responseBuilder.addUsersBuilder()
        .setLogin(user.getLogin())
        .addAllPermissions(permissionsByUserId.get(user.getId()));
      setNullable(user.getEmail(), userResponse::setEmail);
      setNullable(user.getName(), userResponse::setName);
      setNullable(emptyToNull(user.getEmail()), u -> userResponse.setAvatar(avatarResolver.create(user)));
    });
    responseBuilder.getPagingBuilder()
      .setPageIndex(paging.pageIndex())
      .setPageSize(paging.pageSize())
      .setTotal(paging.total())
      .build();
    return responseBuilder.build();
  }

  private List<UserDto> findUsers(DbSession dbSession, PermissionQuery query, PermissionTemplateDto template)
{
    List<String> orderedLogins =
dbClient.permissionTemplateDao().selectUserLoginsByQueryAndTemplate(dbSession, query, template.getId());
    return
Ordering.explicit(orderedLogins).onResultOf(UserDto::getLogin).immutableSortedCopy(dbClient.userDao().select
ByLogins(dbSession, orderedLogins));
  }

}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
```

```
package org.sonar.server.permission.ws.template;

import java.util.Optional;
import javax.annotation.Nullable;
import javax.annotation.concurrent.Immutable;
import org.sonar.db.organization.DefaultTemplates;

import static java.util.Objects.requireNonNull;
import static java.util.Optional.ofNullable;

public interface DefaultTemplatesResolver {
 /**
  * Resolve the effective default templates uuid for the specified {@link DefaultTemplates}.
  * <ul>
  *   <li>{@link ResolvedDefaultTemplates#project} is always the same as {@link
DefaultTemplates#projectUuid}</li>
  *   <li>when Governance is not installed, {@link ResolvedDefaultTemplates#view} is always {@code null}</li>
  *   <li>when Governance is not installed, {@link ResolvedDefaultTemplates#view} is  {@link
DefaultTemplates#viewUuid}
  *      when it is non {@code null}, otherwise it is {@link DefaultTemplates#projectUuid}</li>
  * </ul>
  */
 ResolvedDefaultTemplates resolve(DefaultTemplates defaultTemplates);

 @Immutable
 final class ResolvedDefaultTemplates {
  private final String project;
  private final String view;

  ResolvedDefaultTemplates(String project, @Nullable String view) {
   this.project = requireNonNull(project, "project can't be null");
   this.view = view;
  }
```

```java
  public String getProject() {
    return project;
  }

  public Optional<String> getView() {
    return ofNullable(view);
  }
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import java.util.List;
import java.util.Locale;

import com.google.common.collect.Lists;
import com.google.common.collect.Table;
import com.google.common.collect.TreeBasedTable;
import org.sonar.api.i18n.I18n;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.DefaultTemplates;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.CountByTemplateAndPermissionDto;
```

```java
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions;
import org.sonarqube.ws.Permissions.Permission;
import org.sonarqube.ws.Permissions.PermissionTemplate;
import org.sonarqube.ws.Permissions.SearchTemplatesWsResponse;
import org.sonarqube.ws.Permissions.SearchTemplatesWsResponse.TemplateIdQualifier;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static org.sonar.api.utils.DateUtils.formatDateTime;
import static org.sonar.core.util.Protobuf.setNullable;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.permission.ws.template.SearchTemplatesData.builder;
import static org.sonar.server.ws.WsUtils.checkFoundWithOptional;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;

public class SearchTemplatesAction implements PermissionsWsAction {
 private static final String PROPERTY_PREFIX = "projects_role.";
 private static final String DESCRIPTION_SUFFIX = ".desc";

 private final DbClient dbClient;
 private final UserSession userSession;
 private final I18n i18n;
 private final PermissionWsSupport support;
 private final DefaultTemplatesResolver defaultTemplatesResolver;

 public SearchTemplatesAction(DbClient dbClient, UserSession userSession, I18n i18n, PermissionWsSupport
support, DefaultTemplatesResolver defaultTemplatesResolver) {
   this.dbClient = dbClient;
   this.userSession = userSession;
   this.i18n = i18n;
   this.support = support;
   this.defaultTemplatesResolver = defaultTemplatesResolver;
 }

 @Override
 public void define(WebService.NewController context) {
   WebService.NewAction action = context.createAction("search_templates")
     .setDescription("List permission templates.<br />" +
       "Requires the following permission: 'Administer System'.")
     .setResponseExample(getClass().getResource("search_templates-example.json"))
```

```java
      .setSince("5.2")
      .addSearchQuery("defau", "permission template names")
      .setHandler(this);

    createOrganizationParameter(action).setSince("6.2");
  }

  @Override
  public void handle(Request wsRequest, Response wsResponse) throws Exception {
    try (DbSession dbSession = dbClient.openSession(false)) {
      OrganizationDto org = support.findOrganization(dbSession, wsRequest.param(PARAM_ORGANIZATION));
      SearchTemplatesRequest request = new SearchTemplatesRequest()
        .setOrganizationUuid(org.getUuid())
        .setQuery(wsRequest.param(Param.TEXT_QUERY));
      checkGlobalAdmin(userSession, request.getOrganizationUuid());

      SearchTemplatesWsResponse searchTemplatesWsResponse = buildResponse(load(dbSession, request));
      writeProtobuf(searchTemplatesWsResponse, wsRequest, wsResponse);
    }
  }

  private static void buildDefaultTemplatesResponse(SearchTemplatesWsResponse.Builder response,
  SearchTemplatesData data) {
    TemplateIdQualifier.Builder templateUuidQualifierBuilder = TemplateIdQualifier.newBuilder();

    DefaultTemplatesResolverImpl.ResolvedDefaultTemplates resolvedDefaultTemplates = data.defaultTemplates();
    response.addDefaultTemplates(templateUuidQualifierBuilder
      .setQualifier(Qualifiers.PROJECT)
      .setTemplateId(resolvedDefaultTemplates.getProject()));

    resolvedDefaultTemplates.getView()
      .ifPresent(viewDefaultTemplate -> response.addDefaultTemplates(
        templateUuidQualifierBuilder
          .clear()
          .setQualifier(Qualifiers.VIEW)
          .setTemplateId(viewDefaultTemplate)));
  }

  private static void buildTemplatesResponse(Permissions.SearchTemplatesWsResponse.Builder response,
  SearchTemplatesData data) {
    Permission.Builder permissionResponse = Permission.newBuilder();
    PermissionTemplate.Builder templateBuilder = PermissionTemplate.newBuilder();

    for (PermissionTemplateDto templateDto : data.templates()) {
      templateBuilder
        .clear()
        .setId(templateDto.getUuid())
        .setName(templateDto.getName())
```

```
        .setCreatedAt(formatDateTime(templateDto.getCreatedAt()))
        .setUpdatedAt(formatDateTime(templateDto.getUpdatedAt()));
      setNullable(templateDto.getKeyPattern(), templateBuilder::setProjectKeyPattern);
      setNullable(templateDto.getDescription(), templateBuilder::setDescription);
      for (String permission : ProjectPermissions.ALL) {
        templateBuilder.addPermissions(
          permissionResponse
            .clear()
            .setKey(permission)
            .setUsersCount(data.userCount(templateDto.getId(), permission))
            .setGroupsCount(data.groupCount(templateDto.getId(), permission))
            .setWithProjectCreator(data.withProjectCreator(templateDto.getId(), permission)));
      }
      response.addPermissionTemplates(templateBuilder);
    }
  }

  private Permissions.SearchTemplatesWsResponse buildResponse(SearchTemplatesData data) {
    SearchTemplatesWsResponse.Builder response = SearchTemplatesWsResponse.newBuilder();

    buildTemplatesResponse(response, data);
    buildDefaultTemplatesResponse(response, data);
    buildPermissionsResponse(response);

    return response.build();
  }

  private void buildPermissionsResponse(SearchTemplatesWsResponse.Builder response) {
    Permission.Builder permissionResponse = Permission.newBuilder();
    for (String permissionKey : ProjectPermissions.ALL) {
      response.addPermissions(
        permissionResponse
          .clear()
          .setKey(permissionKey)
          .setName(i18nName(permissionKey))
          .setDescription(i18nDescriptionMessage(permissionKey)));
    }
  }

  private String i18nDescriptionMessage(String permissionKey) {
    return i18n.message(Locale.ENGLISH, PROPERTY_PREFIX + permissionKey + DESCRIPTION_SUFFIX, "");
  }

  private String i18nName(String permissionKey) {
    return i18n.message(Locale.ENGLISH, PROPERTY_PREFIX + permissionKey, permissionKey);
  }

  private SearchTemplatesData load(DbSession dbSession, SearchTemplatesRequest request) {
```

```java
        SearchTemplatesData.Builder data = builder();
        List<PermissionTemplateDto> templates = searchTemplates(dbSession, request);
        List<Long> templateIds = Lists.transform(templates, PermissionTemplateDto::getId);

        DefaultTemplates defaultTemplates = checkFoundWithOptional(
            dbClient.organizationDao().getDefaultTemplates(dbSession, request.getOrganizationUuid()),
            "No Default templates for organization with uuid '%s'", request.getOrganizationUuid());
        DefaultTemplatesResolver.ResolvedDefaultTemplates resolvedDefaultTemplates =
defaultTemplatesResolver.resolve(defaultTemplates);

        data.templates(templates)
            .defaultTemplates(resolvedDefaultTemplates)
            .userCountByTemplateIdAndPermission(userCountByTemplateIdAndPermission(dbSession, templateIds))
            .groupCountByTemplateIdAndPermission(groupCountByTemplateIdAndPermission(dbSession,
templateIds))
.withProjectCreatorByTemplateIdAndPermission(withProjectCreatorsByTemplateIdAndPermission(dbSession, tem
plateIds));

        return data.build();
    }

    private List<PermissionTemplateDto> searchTemplates(DbSession dbSession, SearchTemplatesRequest request) {
        return dbClient.permissionTemplateDao().selectAll(dbSession, request.getOrganizationUuid(),
request.getQuery());
    }

    private Table<Long, String, Integer> userCountByTemplateIdAndPermission(DbSession dbSession, List<Long>
templateIds) {
        final Table<Long, String, Integer> userCountByTemplateIdAndPermission = TreeBasedTable.create();

        dbClient.permissionTemplateDao().usersCountByTemplateIdAndPermission(dbSession, templateIds, context -> {
            CountByTemplateAndPermissionDto row = context.getResultObject();
            userCountByTemplateIdAndPermission.put(row.getTemplateId(), row.getPermission(), row.getCount());
        });

        return userCountByTemplateIdAndPermission;
    }

    private Table<Long, String, Integer> groupCountByTemplateIdAndPermission(DbSession dbSession, List<Long>
templateIds) {
        final Table<Long, String, Integer> userCountByTemplateIdAndPermission = TreeBasedTable.create();

        dbClient.permissionTemplateDao().groupsCountByTemplateIdAndPermission(dbSession, templateIds, context ->
{
            CountByTemplateAndPermissionDto row = context.getResultObject();
            userCountByTemplateIdAndPermission.put(row.getTemplateId(), row.getPermission(), row.getCount());
        });
```

```java
      return userCountByTemplateIdAndPermission;
  }

  private Table<Long, String, Boolean> withProjectCreatorsByTemplateIdAndPermission(DbSession dbSession,
List<Long> templateIds) {
    final Table<Long, String, Boolean> templatePermissionsByTemplateIdAndPermission = TreeBasedTable.create();

    List<PermissionTemplateCharacteristicDto> templatePermissions =
dbClient.permissionTemplateCharacteristicDao().selectByTemplateIds(dbSession, templateIds);
    templatePermissions.stream()
        .forEach(templatePermission ->
templatePermissionsByTemplateIdAndPermission.put(templatePermission.getTemplateId(),
templatePermission.getPermission(),
            templatePermission.getWithProjectCreator()));

    return templatePermissionsByTemplateIdAndPermission;
  }

  private static class SearchTemplatesRequest {
    private String query;
    private String organizationUuid;

    @CheckForNull
    public String getQuery() {
      return query;
    }

    public SearchTemplatesRequest setQuery(@Nullable String query) {
      this.query = query;
      return this;
    }

    public String getOrganizationUuid() {
      return organizationUuid;
    }

    public SearchTemplatesRequest setOrganizationUuid(String s) {
      this.organizationUuid = s;
      return this;
    }
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
```

package org.sonar.server.permission.ws.template;

import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.UserId;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static java.util.Objects.requireNonNull;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPermission;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createUserLoginParameter;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class RemoveUserFromTemplateAction implements PermissionsWsAction {
 private final DbClient dbClient;
 private final PermissionWsSupport wsSupport;
 private final UserSession userSession;

 public RemoveUserFromTemplateAction(DbClient dbClient, PermissionWsSupport wsSupport, UserSession
 userSession) {

```java
    this.dbClient = dbClient;
    this.wsSupport = wsSupport;
    this.userSession = userSession;
  }

  private static RemoveUserFromTemplateRequest toRemoveUserFromTemplateWsRequest(Request request) {
    return new RemoveUserFromTemplateRequest()
      .setPermission(request.mandatoryParam(PARAM_PERMISSION))
      .setLogin(request.mandatoryParam(PARAM_USER_LOGIN))
      .setTemplateId(request.param(PARAM_TEMPLATE_ID))
      .setOrganization(request.param(PARAM_ORGANIZATION))
      .setTemplateName(request.param(PARAM_TEMPLATE_NAME));
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context
      .createAction("remove_user_from_template")
      .setPost(true)
      .setSince("5.2")
      .setDescription("Remove a user from a permission template.<br /> " +
        "Requires the following permission: 'Administer System'.")
      .setHandler(this);

    createTemplateParameters(action);
    createProjectPermissionParameter(action);
    createUserLoginParameter(action);
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    doHandle(toRemoveUserFromTemplateWsRequest(request));
    response.noContent();
  }

  private void doHandle(RemoveUserFromTemplateRequest request) {
    String permission = request.getPermission();
    String userLogin = request.getLogin();

    try (DbSession dbSession = dbClient.openSession(false)) {
      validateProjectPermission(permission);
      PermissionTemplateDto template = wsSupport.findTemplate(dbSession, WsTemplateRef.newTemplateRef(
        request.getTemplateId(), request.getOrganization(), request.getTemplateName()));
      checkGlobalAdmin(userSession, template.getOrganizationUuid());

      UserId user = wsSupport.findUser(dbSession, userLogin);

      dbClient.permissionTemplateDao().deleteUserPermission(dbSession, template.getId(), user.getId(), permission);
```

```java
      dbSession.commit();
    }
  }

  private static class RemoveUserFromTemplateRequest {
    private String login;
    private String permission;
    private String templateId;
    private String organization;
    private String templateName;

    public String getLogin() {
      return login;
    }

    public RemoveUserFromTemplateRequest setLogin(String login) {
      this.login = requireNonNull(login);
      return this;
    }

    public String getPermission() {
      return permission;
    }

    public RemoveUserFromTemplateRequest setPermission(String permission) {
      this.permission = requireNonNull(permission);
      return this;
    }

    @CheckForNull
    public String getTemplateId() {
      return templateId;
    }

    public RemoveUserFromTemplateRequest setTemplateId(@Nullable String templateId) {
      this.templateId = templateId;
      return this;
    }

    @CheckForNull
    public String getOrganization() {
      return organization;
    }

    public RemoveUserFromTemplateRequest setOrganization(@Nullable String s) {
      this.organization = s;
      return this;
    }
```

```java
  @CheckForNull
  public String getTemplateName() {
    return templateName;
  }

  public RemoveUserFromTemplateRequest setTemplateName(@Nullable String templateName) {
    this.templateName = templateName;
    return this;
  }
 }
}
```

```java
package org.sonar.server.permission.ws.template;

import java.util.List;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.UserId;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
```

```java
import javax.annotation.Nullable;

import static java.util.Objects.requireNonNull;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createUserLoginParameter;
import static org.sonar.server.permission.ws.template.WsTemplateRef.newTemplateRef;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class AddUserToTemplateAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final PermissionWsSupport wsSupport;
  private final UserSession userSession;

  public AddUserToTemplateAction(DbClient dbClient, PermissionWsSupport wsSupport, UserSession userSession)
  {
    this.dbClient = dbClient;
    this.wsSupport = wsSupport;
    this.userSession = userSession;
  }

  private static AddUserToTemplateRequest toAddUserToTemplateWsRequest(Request request) {
    return new AddUserToTemplateRequest()
      .setLogin(request.mandatoryParam(PARAM_USER_LOGIN))
      .setPermission(request.mandatoryParam(PARAM_PERMISSION))
      .setTemplateId(request.param(PARAM_TEMPLATE_ID))
      .setOrganization(request.param(PARAM_ORGANIZATION))
      .setTemplateName(request.param(PARAM_TEMPLATE_NAME));
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context
      .createAction("add_user_to_template")
      .setPost(true)
      .setSince("5.2")
      .setDescription("Add a user to a permission template.<br /> " +
        "Requires the following permission: 'Administer System'.")
      .setHandler(this);

    createTemplateParameters(action);
    createProjectPermissionParameter(action);
    createUserLoginParameter(action);
```

```java
    }

    @Override
    public void handle(Request request, Response response) throws Exception {
      doHandle(toAddUserToTemplateWsRequest(request));
      response.noContent();
    }

    private void doHandle(AddUserToTemplateRequest request) {
      String permission = request.getPermission();
      String userLogin = request.getLogin();

      try (DbSession dbSession = dbClient.openSession(false)) {
        PermissionTemplateDto template = wsSupport.findTemplate(dbSession, newTemplateRef(
          request.getTemplateId(), request.getOrganization(), request.getTemplateName()));
        OrganizationDto organizationDto = wsSupport.findOrganization(dbSession, request.getOrganization());
        checkGlobalAdmin(userSession, organizationDto.getUuid());
        UserId user = wsSupport.findUser(dbSession, userLogin);
        wsSupport.checkMembership(dbSession, organizationDto, user);

        if (!isUserAlreadyAdded(dbSession, organizationDto, template.getId(), userLogin, permission)) {
          dbClient.permissionTemplateDao().insertUserPermission(dbSession, template.getId(), user.getId(), permission);
          dbSession.commit();
        }
      }
    }

    private boolean isUserAlreadyAdded(DbSession dbSession, OrganizationDto organizationDto, long templateId,
    String userLogin, String permission) {
      PermissionQuery permissionQuery =
    PermissionQuery.builder().setOrganizationUuid(organizationDto.getUuid()).setPermission(permission).build();
      List<String> usersWithPermission =
    dbClient.permissionTemplateDao().selectUserLoginsByQueryAndTemplate(dbSession, permissionQuery,
    templateId);
      return usersWithPermission.stream().anyMatch(s -> s.equals(userLogin));
    }

    private static class AddUserToTemplateRequest {
      private String login;
      private String permission;
      private String templateId;
      private String organization;
      private String templateName;

      public String getLogin() {
        return login;
      }
```

```java
  public AddUserToTemplateRequest setLogin(String login) {
    this.login = requireNonNull(login);
    return this;
  }

  public String getPermission() {
    return permission;
  }

  public AddUserToTemplateRequest setPermission(String permission) {
    this.permission = requireNonNull(permission);
    return this;
  }

  @CheckForNull
  public String getTemplateId() {
    return templateId;
  }

  public AddUserToTemplateRequest setTemplateId(@Nullable String templateId) {
    this.templateId = templateId;
    return this;
  }

  @CheckForNull
  public String getTemplateName() {
    return templateName;
  }

  public AddUserToTemplateRequest setTemplateName(@Nullable String templateName) {
    this.templateName = templateName;
    return this;
  }

  @CheckForNull
  public String getOrganization() {
    return organization;
  }

  public AddUserToTemplateRequest setOrganization(@Nullable String s) {
    this.organization = s;
    return this;
  }
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
```

```java
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
import org.sonar.api.server.ws.Request;

import static org.sonar.server.ws.WsUtils.checkRequest;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

/**
 * Reference to a template as defined by WS request. Guaranties one of template id or
 * template name is provided, not both.
 */
public class WsTemplateRef {

 private final String uuid;
 private final String organization;
 private final String name;

 private WsTemplateRef(@Nullable String uuid, @Nullable String organization, @Nullable String name) {
   checkRequest(uuid != null ^ name != null, "Template name or template id must be provided, not both.");

   this.uuid = uuid;
   this.organization = organization;
   this.name = name;
 }

 public static WsTemplateRef fromRequest(Request wsRequest) {
   String uuid = wsRequest.param(PARAM_TEMPLATE_ID);
   String organization = wsRequest.param(PARAM_ORGANIZATION);
```

```java
    String name = wsRequest.param(PARAM_TEMPLATE_NAME);

    return new WsTemplateRef(uuid, organization, name);
  }

  public static WsTemplateRef newTemplateRef(@Nullable String uuid, @Nullable String organization, @Nullable
String name) {
    return new WsTemplateRef(uuid, organization, name);
  }

  @CheckForNull
  public String uuid() {
    return this.uuid;
  }

  @CheckForNull
  public String getOrganization() {
    return this.organization;
  }

  @CheckForNull
  public String name() {
    return this.name;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import com.google.common.base.Function;
import javax.annotation.Nonnull;
```

```java
import org.sonar.api.utils.DateUtils;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonarqube.ws.Permissions.PermissionTemplate;

import static org.sonar.core.util.Protobuf.setNullable;

public class PermissionTemplateDtoToPermissionTemplateResponse {

 private PermissionTemplateDtoToPermissionTemplateResponse() {
   // prevent instantiation
 }

 public static PermissionTemplate toPermissionTemplateResponse(PermissionTemplateDto dto) {
   return Singleton.INSTANCE.apply(dto);
 }

 private enum Singleton implements Function<PermissionTemplateDto, PermissionTemplate> {
   INSTANCE;
   @Override
   public PermissionTemplate apply(@Nonnull PermissionTemplateDto permissionTemplate) {
     PermissionTemplate.Builder permissionTemplateBuilder = PermissionTemplate.newBuilder()
       .setId(permissionTemplate.getUuid())
       .setName(permissionTemplate.getName())
       .setCreatedAt(DateUtils.formatDateTime(permissionTemplate.getCreatedAt()))
       .setUpdatedAt(DateUtils.formatDateTime(permissionTemplate.getUpdatedAt()));
     setNullable(permissionTemplate.getDescription(), permissionTemplateBuilder::setDescription);
     setNullable(permissionTemplate.getKeyPattern(), permissionTemplateBuilder::setProjectKeyPattern);
     return permissionTemplateBuilder.build();
   }
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
```

```
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
package org.sonar.server.permission.ws.template;

import org.sonar.api.i18n.I18n;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.resources.ResourceTypes;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.DefaultTemplates;
import org.sonar.db.organization.OrganizationDao;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateQualifier;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.permission.ws.template.WsTemplateRef.newTemplateRef;
import static org.sonar.server.ws.WsParameterBuilder.QualifierParameterContext.newQualifierParameterContext;
import static org.sonar.server.ws.WsParameterBuilder.createDefaultTemplateQualifierParameter;
import static org.sonar.server.ws.WsUtils.checkFoundWithOptional;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_QUALIFIER;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class SetDefaultTemplateAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final PermissionWsSupport wsSupport;
  private final ResourceTypes resourceTypes;
  private final UserSession userSession;
  private final I18n i18n;

  public SetDefaultTemplateAction(DbClient dbClient, PermissionWsSupport wsSupport, ResourceTypes
resourceTypes,
    UserSession userSession, I18n i18n) {
    this.dbClient = dbClient;
    this.wsSupport = wsSupport;
    this.resourceTypes = resourceTypes;
    this.userSession = userSession;
```

```java
    this.i18n = i18n;
  }

  private static SetDefaultTemplateRequest toSetDefaultTemplateWsRequest(Request request) {
    return new SetDefaultTemplateRequest()
      .setQualifier(request.param(PARAM_QUALIFIER))
      .setTemplateId(request.param(PARAM_TEMPLATE_ID))
      .setOrganization(request.param(PARAM_ORGANIZATION))
      .setTemplateName(request.param(PARAM_TEMPLATE_NAME));
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("set_default_template")
      .setDescription("Set a permission template as default.<br />" +
        "Requires the following permission: 'Administer System'.")
      .setPost(true)
      .setSince("5.2")
      .setHandler(this);

    createTemplateParameters(action);
    createDefaultTemplateQualifierParameter(action, newQualifierParameterContext(i18n, resourceTypes))
      .setDefaultValue(Qualifiers.PROJECT);
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    doHandle(toSetDefaultTemplateWsRequest(request));
    response.noContent();
  }

  private void doHandle(SetDefaultTemplateRequest request) {
    try (DbSession dbSession = dbClient.openSession(false)) {
      String qualifier = request.getQualifier();
      PermissionTemplateDto template = findTemplate(dbSession, request);
      checkGlobalAdmin(userSession, template.getOrganizationUuid());
      validateQualifier(qualifier, resourceTypes);
      setDefaultTemplateUuid(dbSession, template, qualifier);
      dbSession.commit();
    }
  }

  private PermissionTemplateDto findTemplate(DbSession dbSession, SetDefaultTemplateRequest request) {
    return wsSupport.findTemplate(dbSession, newTemplateRef(request.getTemplateId(),
      request.getOrganization(), request.getTemplateName()));
  }

  private void setDefaultTemplateUuid(DbSession dbSession, PermissionTemplateDto permissionTemplateDto,
```

```java
      String qualifier) {
        String organizationUuid = permissionTemplateDto.getOrganizationUuid();
        OrganizationDao organizationDao = dbClient.organizationDao();

        DefaultTemplates defaultTemplates = checkFoundWithOptional(
          organizationDao.getDefaultTemplates(dbSession, organizationUuid),
          "No Default templates for organization with uuid '%s'", organizationUuid);
        if (Qualifiers.PROJECT.equals(qualifier)) {
          defaultTemplates.setProjectUuid(permissionTemplateDto.getUuid());
        } else if (Qualifiers.VIEW.equals(qualifier)) {
          defaultTemplates.setViewUuid(permissionTemplateDto.getUuid());
        }
        organizationDao.setDefaultTemplates(dbSession, organizationUuid, defaultTemplates);
      }

      private static class SetDefaultTemplateRequest {
        private String qualifier;
        private String templateId;
        private String organization;
        private String templateName;

        @CheckForNull
        public String getQualifier() {
          return qualifier;
        }

        public SetDefaultTemplateRequest setQualifier(@Nullable String qualifier) {
          this.qualifier = qualifier;
          return this;
        }

        @CheckForNull
        public String getTemplateId() {
          return templateId;
        }

        public SetDefaultTemplateRequest setTemplateId(@Nullable String templateId) {
          this.templateId = templateId;
          return this;
        }

        @CheckForNull
        public String getOrganization() {
          return organization;
        }

        public SetDefaultTemplateRequest setOrganization(@Nullable String s) {
          this.organization = s;
```

```java
      return this;
    }

    @CheckForNull
    public String getTemplateName() {
      return templateName;
    }

    public SetDefaultTemplateRequest setTemplateName(@Nullable String templateName) {
      this.templateName = templateName;
      return this;
    }
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import com.google.common.collect.Table;
import java.util.List;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.template.DefaultTemplatesResolver.ResolvedDefaultTemplates;

import static com.google.common.base.MoreObjects.firstNonNull;
import static com.google.common.base.Preconditions.checkState;
import static com.google.common.collect.ImmutableList.copyOf;
import static com.google.common.collect.ImmutableTable.copyOf;

class SearchTemplatesData {
  private final List<PermissionTemplateDto> templates;
  private final ResolvedDefaultTemplates defaultTemplates;
```

```java
  private final Table<Long, String, Integer> userCountByTemplateIdAndPermission;
  private final Table<Long, String, Integer> groupCountByTemplateIdAndPermission;
  private final Table<Long, String, Boolean> withProjectCreatorByTemplateIdAndPermission;

  private SearchTemplatesData(Builder builder) {
    this.templates = copyOf(builder.templates);
    this.defaultTemplates = builder.defaultTemplates;
    this.userCountByTemplateIdAndPermission = copyOf(builder.userCountByTemplateIdAndPermission);
    this.groupCountByTemplateIdAndPermission = copyOf(builder.groupCountByTemplateIdAndPermission);
    this.withProjectCreatorByTemplateIdAndPermission =
  copyOf(builder.withProjectCreatorByTemplateIdAndPermission);
  }

  public static Builder builder() {
    return new Builder();
  }

  public List<PermissionTemplateDto> templates() {
    return templates;
  }

  public ResolvedDefaultTemplates defaultTemplates() {
    return defaultTemplates;
  }

  public int userCount(long templateId, String permission) {
    return firstNonNull(userCountByTemplateIdAndPermission.get(templateId, permission), 0);
  }

  public int groupCount(long templateId, String permission) {
    return firstNonNull(groupCountByTemplateIdAndPermission.get(templateId, permission), 0);
  }

  public boolean withProjectCreator(long templateId, String permission) {
    return firstNonNull(withProjectCreatorByTemplateIdAndPermission.get(templateId, permission), false);
  }

  public static class Builder {
    private List<PermissionTemplateDto> templates;
    private ResolvedDefaultTemplates defaultTemplates;
    private Table<Long, String, Integer> userCountByTemplateIdAndPermission;
    private Table<Long, String, Integer> groupCountByTemplateIdAndPermission;
    private Table<Long, String, Boolean> withProjectCreatorByTemplateIdAndPermission;

    private Builder() {
      // prevents instantiation outside main class
    }
```

```java
  public SearchTemplatesData build() {
    checkState(templates != null);
    checkState(defaultTemplates != null);
    checkState(userCountByTemplateIdAndPermission != null);
    checkState(groupCountByTemplateIdAndPermission != null);
    checkState(withProjectCreatorByTemplateIdAndPermission != null);

    return new SearchTemplatesData(this);
  }

  public Builder templates(List<PermissionTemplateDto> templates) {
    this.templates = templates;
    return this;
  }

  public Builder defaultTemplates(ResolvedDefaultTemplates defaultTemplates) {
    this.defaultTemplates = defaultTemplates;
    return this;
  }

  public Builder userCountByTemplateIdAndPermission(Table<Long, String, Integer>
userCountByTemplateIdAndPermission) {
    this.userCountByTemplateIdAndPermission = userCountByTemplateIdAndPermission;
    return this;
  }

  public Builder groupCountByTemplateIdAndPermission(Table<Long, String, Integer>
groupCountByTemplateIdAndPermission) {
    this.groupCountByTemplateIdAndPermission = groupCountByTemplateIdAndPermission;
    return this;
  }

  public Builder withProjectCreatorByTemplateIdAndPermission(Table<Long, String, Boolean>
withProjectCreatorByTemplateIdAndPermission) {
    this.withProjectCreatorByTemplateIdAndPermission = withProjectCreatorByTemplateIdAndPermission;
    return this;
  }
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
```

```
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;

import static com.google.common.base.Preconditions.checkArgument;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupIdParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupNameParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;

public class RemoveGroupFromTemplateAction implements PermissionsWsAction {
 private final DbClient dbClient;
 private final PermissionWsSupport wsSupport;
 private final UserSession userSession;

 public RemoveGroupFromTemplateAction(DbClient dbClient, PermissionWsSupport wsSupport, UserSession
userSession) {
   this.dbClient = dbClient;
   this.wsSupport = wsSupport;
   this.userSession = userSession;
 }

 @Override
 public void define(WebService.NewController context) {
   WebService.NewAction action = context
     .createAction("remove_group_from_template")
     .setPost(true)
```

```java
      .setSince("5.2")
      .setDescription("Remove a group from a permission template.<br /> " +
        "The group id or group name must be provided. <br />" +
        "Requires the following permission: 'Administer System'.")
      .setHandler(this);

    createTemplateParameters(action);
    createProjectPermissionParameter(action);
    createGroupIdParameter(action);
    createGroupNameParameter(action);
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    try (DbSession dbSession = dbClient.openSession(false)) {
      String permission = request.mandatoryParam(PARAM_PERMISSION);
      PermissionTemplateDto template = wsSupport.findTemplate(dbSession, WsTemplateRef.fromRequest(request));
      checkGlobalAdmin(userSession, template.getOrganizationUuid());
      GroupIdOrAnyone groupId = wsSupport.findGroup(dbSession, request);
      checkArgument(groupId.getOrganizationUuid().equals(template.getOrganizationUuid()), "Group and template
are on different organizations");

      dbClient.permissionTemplateDao().deleteGroupPermission(dbSession, template.getId(), groupId.getId(),
permission);
      dbSession.commit();
    }
    response.noContent();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
```

```java
package org.sonar.server.permission.ws.template;

import com.google.common.collect.Multimap;
import com.google.common.collect.Ordering;
import com.google.common.collect.TreeMultimap;
import java.util.List;
import java.util.stream.Collectors;
import org.sonar.api.security.DefaultGroups;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.utils.Paging;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.permission.template.PermissionTemplateGroupDto;
import org.sonar.db.user.GroupDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions;

import static org.sonar.api.server.ws.WebService.Param.PAGE;
import static org.sonar.api.server.ws.WebService.Param.PAGE_SIZE;
import static org.sonar.api.server.ws.WebService.Param.TEXT_QUERY;
import static org.sonar.core.util.Protobuf.setNullable;
import static org.sonar.db.permission.PermissionQuery.DEFAULT_PAGE_SIZE;
import static org.sonar.db.permission.PermissionQuery.RESULTS_MAX_SIZE;
import static org.sonar.db.permission.PermissionQuery.SEARCH_QUERY_MIN_LENGTH;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPermission;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;

public class TemplateGroupsAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final UserSession userSession;
  private final PermissionWsSupport support;

  public TemplateGroupsAction(DbClient dbClient, UserSession userSession, PermissionWsSupport support) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.support = support;
  }
```

```java
@Override
public void define(WebService.NewController context) {
  WebService.NewAction action = context.createAction("template_groups")
    .setSince("5.2")
    .setInternal(true)
    .setDescription("Lists the groups with their permission as individual groups rather than through user affiliation on
the chosen template.<br />" +
      "This service defaults to all groups, but can be limited to groups with a specific permission by providing the
desired permission.<br>" +
      "Requires the following permission: 'Administer System'.")
    .addPagingParams(DEFAULT_PAGE_SIZE, RESULTS_MAX_SIZE)
    .setResponseExample(getClass().getResource("template_groups-example.json"))
    .setHandler(this);

  action.createParam(TEXT_QUERY)
    .setMinimumLength(SEARCH_QUERY_MIN_LENGTH)
    .setDescription("Limit search to group names that contain the supplied string. <br/>" +
      "When this parameter is not set, only group having at least one permission are returned.")
    .setExampleValue("eri");

  createProjectPermissionParameter(action, false);
  createTemplateParameters(action);
}

@Override
public void handle(Request wsRequest, Response wsResponse) throws Exception {
  try (DbSession dbSession = dbClient.openSession(false)) {
    WsTemplateRef templateRef = WsTemplateRef.fromRequest(wsRequest);
    PermissionTemplateDto template = support.findTemplate(dbSession, templateRef);
    checkGlobalAdmin(userSession, template.getOrganizationUuid());

    PermissionQuery query = buildPermissionQuery(wsRequest, template);
    int total = dbClient.permissionTemplateDao().countGroupNamesByQueryAndTemplate(dbSession, query,
template.getOrganizationUuid(), template.getId());
    Paging paging =
Paging.forPageIndex(wsRequest.mandatoryParamAsInt(PAGE)).withPageSize(wsRequest.mandatoryParamAsInt(P
AGE_SIZE)).andTotal(total);
    List<GroupDto> groups = findGroups(dbSession, query, template);
    List<PermissionTemplateGroupDto> groupPermissions = findGroupPermissions(dbSession, groups, template);
    Permissions.WsGroupsResponse groupsResponse = buildResponse(groups, groupPermissions, paging);
    writeProtobuf(groupsResponse, wsRequest, wsResponse);
  }
}

private static PermissionQuery buildPermissionQuery(Request request, PermissionTemplateDto template) {
  String textQuery = request.param(TEXT_QUERY);
  String permission = request.param(PARAM_PERMISSION);
  PermissionQuery.Builder permissionQuery = PermissionQuery.builder()
```

```java
      .setOrganizationUuid(template.getOrganizationUuid())
      .setPermission(permission != null ? validateProjectPermission(permission) : null)
      .setPageIndex(request.mandatoryParamAsInt(PAGE))
      .setPageSize(request.mandatoryParamAsInt(PAGE_SIZE))
      .setSearchQuery(textQuery);
    if (textQuery == null) {
     permissionQuery.withAtLeastOnePermission();
    }
    return permissionQuery.build();
  }

  private static Permissions.WsGroupsResponse buildResponse(List<GroupDto> groups,
List<PermissionTemplateGroupDto> groupPermissions, Paging paging) {
    Multimap<Integer, String> permissionsByGroupId = TreeMultimap.create();
    groupPermissions.forEach(groupPermission -> permissionsByGroupId.put(groupPermission.getGroupId(),
groupPermission.getPermission()));
    Permissions.WsGroupsResponse.Builder response = Permissions.WsGroupsResponse.newBuilder();

    groups.forEach(group -> {
     Permissions.Group.Builder wsGroup = response.addGroupsBuilder()
       .setName(group.getName());
     if (group.getId() != 0) {
      wsGroup.setId(String.valueOf(group.getId()));
     }
     setNullable(group.getDescription(), wsGroup::setDescription);
     wsGroup.addAllPermissions(permissionsByGroupId.get(group.getId()));
    });

    response.getPagingBuilder()
     .setPageIndex(paging.pageIndex())
     .setPageSize(paging.pageSize())
     .setTotal(paging.total());
    return response.build();
  }

  private List<GroupDto> findGroups(DbSession dbSession, PermissionQuery dbQuery, PermissionTemplateDto
template) {
    List<String> orderedNames =
dbClient.permissionTemplateDao().selectGroupNamesByQueryAndTemplate(dbSession, dbQuery,
template.getId());
    List<GroupDto> groups = dbClient.groupDao().selectByNames(dbSession, template.getOrganizationUuid(),
orderedNames);
    if (orderedNames.contains(DefaultGroups.ANYONE)) {
     groups.add(0, new GroupDto().setId(0).setName(DefaultGroups.ANYONE));
    }
    return Ordering.explicit(orderedNames).onResultOf(GroupDto::getName).immutableSortedCopy(groups);
  }
```

```java
 private List<PermissionTemplateGroupDto> findGroupPermissions(DbSession dbSession, List<GroupDto>
groups, PermissionTemplateDto template) {
   List<String> names = groups.stream().map(GroupDto::getName).collect(Collectors.toList());
   return dbClient.permissionTemplateDao().selectGroupPermissionsByTemplateIdAndGroupNames(dbSession,
template.getId(), names);
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
package org.sonar.server.permission.ws.template;

import java.util.Collections;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.PermissionTemplateService;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonar.server.permission.ws.ProjectWsRef.newWsProjectRef;
```

```java
import static org.sonar.server.permission.ws.template.WsTemplateRef.newTemplateRef;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class ApplyTemplateAction implements PermissionsWsAction {
 private final DbClient dbClient;
 private final UserSession userSession;
 private final PermissionTemplateService permissionTemplateService;
 private final PermissionWsSupport wsSupport;

 public ApplyTemplateAction(DbClient dbClient, UserSession userSession, PermissionTemplateService
permissionTemplateService,
  PermissionWsSupport wsSupport) {
  this.dbClient = dbClient;
  this.userSession = userSession;
  this.permissionTemplateService = permissionTemplateService;
  this.wsSupport = wsSupport;
 }

 private static ApplyTemplateRequest toApplyTemplateWsRequest(Request request) {
  return new ApplyTemplateRequest()
   .setProjectId(request.param(PARAM_PROJECT_ID))
   .setProjectKey(request.param(PARAM_PROJECT_KEY))
   .setTemplateId(request.param(PARAM_TEMPLATE_ID))
   .setTemplateName(request.param(PARAM_TEMPLATE_NAME))
   .setOrganization(request.param(PARAM_ORGANIZATION));
 }

 @Override
 public void define(WebService.NewController context) {
  WebService.NewAction action = context.createAction("apply_template")
   .setDescription("Apply a permission template to one project.<br>" +
     "The project id or project key must be provided.<br>" +
     "The template id or name must be provided.<br>" +
     "Requires the following permission: 'Administer System'.")
   .setPost(true)
   .setSince("5.2")
   .setHandler(this);

  createTemplateParameters(action);
  createProjectParameters(action);
 }

 @Override
 public void handle(Request request, Response response) throws Exception {
```

```java
      doHandle(toApplyTemplateWsRequest(request));
      response.noContent();
    }

    private void doHandle(ApplyTemplateRequest request) {
      try (DbSession dbSession = dbClient.openSession(false)) {
        PermissionTemplateDto template = wsSupport.findTemplate(dbSession, newTemplateRef(
          request.getTemplateId(), request.getOrganization(), request.getTemplateName()));

        ComponentDto project = wsSupport.getRootComponentOrModule(dbSession,
newWsProjectRef(request.getProjectId(), request.getProjectKey()));
        checkGlobalAdmin(userSession, template.getOrganizationUuid());

        permissionTemplateService.applyAndCommit(dbSession, template, Collections.singletonList(project));
      }
    }

    private static class ApplyTemplateRequest {
      private String projectId;
      private String projectKey;
      private String templateId;
      private String organization;
      private String templateName;

      @CheckForNull
      public String getProjectId() {
        return projectId;
      }

      public ApplyTemplateRequest setProjectId(@Nullable String projectId) {
        this.projectId = projectId;
        return this;
      }

      @CheckForNull
      public String getProjectKey() {
        return projectKey;
      }

      public ApplyTemplateRequest setProjectKey(@Nullable String projectKey) {
        this.projectKey = projectKey;
        return this;
      }

      @CheckForNull
      public String getTemplateId() {
        return templateId;
      }
```

```java
    public ApplyTemplateRequest setTemplateId(@Nullable String templateId) {
      this.templateId = templateId;
      return this;
    }

    @CheckForNull
    public String getOrganization() {
      return organization;
    }

    public ApplyTemplateRequest setOrganization(@Nullable String s) {
      this.organization = s;
      return this;
    }
    @CheckForNull
    public String getTemplateName() {
      return templateName;
    }

    public ApplyTemplateRequest setTemplateName(@Nullable String templateName) {
      this.templateName = templateName;
      return this;
    }
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import java.util.Optional;
```

```java
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.utils.System2;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.permission.ws.PermissionsWsAction;
import org.sonar.server.user.UserSession;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static java.util.Objects.requireNonNull;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPermission;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createTemplateParameters;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class AddProjectCreatorToTemplateAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final PermissionWsSupport wsSupport;
  private final UserSession userSession;
  private final System2 system;

  public AddProjectCreatorToTemplateAction(DbClient dbClient, PermissionWsSupport wsSupport, UserSession
userSession, System2 system) {
    this.dbClient = dbClient;
    this.wsSupport = wsSupport;
    this.userSession = userSession;
    this.system = system;
  }

  private static AddProjectCreatorToTemplateRequest toWsRequest(Request request) {
    AddProjectCreatorToTemplateRequest wsRequest = AddProjectCreatorToTemplateRequest.builder()
      .setPermission(request.mandatoryParam(PARAM_PERMISSION))
      .setTemplateId(request.param(PARAM_TEMPLATE_ID))
      .setOrganization(request.param(PARAM_ORGANIZATION))
      .setTemplateName(request.param(PARAM_TEMPLATE_NAME))
      .build();
    validateProjectPermission(wsRequest.getPermission());
    return wsRequest;
```

```
    }

    @Override
    public void define(WebService.NewController context) {
      WebService.NewAction action = context.createAction("add_project_creator_to_template")
        .setDescription("Add a project creator to a permission template.<br>" +
          "Requires the following permission: 'Administer System'.")
        .setSince("6.0")
        .setPost(true)
        .setHandler(this);

      createTemplateParameters(action);
      createProjectPermissionParameter(action);
    }

    @Override
    public void handle(Request request, Response response) throws Exception {
      doHandle(toWsRequest(request));
      response.noContent();
    }

    private void doHandle(AddProjectCreatorToTemplateRequest request) {
      try (DbSession dbSession = dbClient.openSession(false)) {
        PermissionTemplateDto template = wsSupport.findTemplate(dbSession, WsTemplateRef.newTemplateRef(
          request.getTemplateId(), request.getOrganization(), request.getTemplateName()));
        checkGlobalAdmin(userSession, template.getOrganizationUuid());

        Optional<PermissionTemplateCharacteristicDto> templatePermission =
dbClient.permissionTemplateCharacteristicDao()
          .selectByPermissionAndTemplateId(dbSession, request.getPermission(), template.getId());
        if (templatePermission.isPresent()) {
          updateTemplatePermission(dbSession, templatePermission.get());
        } else {
          addTemplatePermission(dbSession, request, template);
        }
      }
    }

    private void addTemplatePermission(DbSession dbSession, AddProjectCreatorToTemplateRequest request,
PermissionTemplateDto template) {
      long now = system.now();
      dbClient.permissionTemplateCharacteristicDao().insert(dbSession, new PermissionTemplateCharacteristicDto()
        .setPermission(request.getPermission())
        .setTemplateId(template.getId())
        .setWithProjectCreator(true)
        .setCreatedAt(now)
        .setUpdatedAt(now));
      dbSession.commit();
```

```java
  }

  private void updateTemplatePermission(DbSession dbSession, PermissionTemplateCharacteristicDto
templatePermission) {
    PermissionTemplateCharacteristicDto targetTemplatePermission = templatePermission
      .setUpdatedAt(system.now())
      .setWithProjectCreator(true);
    dbClient.permissionTemplateCharacteristicDao().update(dbSession, targetTemplatePermission);
    dbSession.commit();
  }

  private static class AddProjectCreatorToTemplateRequest {
    private final String templateId;
    private final String organization;
    private final String templateName;
    private final String permission;

    private AddProjectCreatorToTemplateRequest(Builder builder) {
      this.templateId = builder.templateId;
      this.organization = builder.organization;
      this.templateName = builder.templateName;
      this.permission = requireNonNull(builder.permission);
    }

    @CheckForNull
    public String getTemplateId() {
      return templateId;
    }

    @CheckForNull
    public String getOrganization() {
      return organization;
    }

    @CheckForNull
    public String getTemplateName() {
      return templateName;
    }

    public String getPermission() {
      return permission;
    }

    public static Builder builder() {
      return new Builder();
    }
  }
```

```java
private static class Builder {
  private String templateId;
  private String organization;
  private String templateName;
  private String permission;

  private Builder() {
    // enforce method constructor
  }

  public Builder setTemplateId(@Nullable String templateId) {
    this.templateId = templateId;
    return this;
  }

  public Builder setOrganization(@Nullable String s) {
    this.organization = s;
    return this;
  }

  public Builder setTemplateName(@Nullable String templateName) {
    this.templateName = templateName;
    return this;
  }

  public Builder setPermission(@Nullable String permission) {
    this.permission = permission;
    return this;
  }

  public AddProjectCreatorToTemplateRequest build() {
    return new AddProjectCreatorToTemplateRequest(this);
  }
}
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
```

```
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import java.util.Optional;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.server.permission.GroupPermissionChange;
import org.sonar.server.permission.PermissionChange;
import org.sonar.server.permission.PermissionUpdater;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.user.UserSession;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;

import static java.util.Arrays.asList;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkProjectAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupIdParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupNameParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;

public class AddGroupAction implements PermissionsWsAction {

 public static final String ACTION = "add_group";

 private final DbClient dbClient;
 private final UserSession userSession;
 private final PermissionUpdater permissionUpdater;
 private final PermissionWsSupport support;

 public AddGroupAction(DbClient dbClient, UserSession userSession, PermissionUpdater permissionUpdater,
PermissionWsSupport support) {
  this.dbClient = dbClient;
  this.userSession = userSession;
  this.permissionUpdater = permissionUpdater;
  this.support = support;
 }

 @Override
```

```java
 public void define(WebService.NewController context) {
   WebService.NewAction action = context.createAction(ACTION)
     .setDescription("Add permission to a group.<br /> " +
       "This service defaults to global permissions, but can be limited to project permissions by providing project id or
project key.<br /> " +
       "The group name or group id must be provided. <br />" +
       "Requires one of the following permissions:" +
       "<ul>" +
       "<li>'Administer System'</li>" +
       "<li>'Administer' rights on the specified project</li>" +
       "</ul>")
     .setSince("5.2")
     .setPost(true)
     .setHandler(this);

   createPermissionParameter(action);
   createOrganizationParameter(action).setSince("6.2");
   createGroupNameParameter(action);
   createGroupIdParameter(action);
   createProjectParameters(action);
 }

 @Override
 public void handle(Request request, Response response) throws Exception {
   try (DbSession dbSession = dbClient.openSession(false)) {
     GroupIdOrAnyone group = support.findGroup(dbSession, request);
     Optional<ProjectId> projectId = support.findProjectId(dbSession, request);

     checkProjectAdmin(userSession, group.getOrganizationUuid(), projectId);

     PermissionChange change = new GroupPermissionChange(
       PermissionChange.Operation.ADD,
       request.mandatoryParam(PARAM_PERMISSION),
       projectId.orElse(null),
       group);
     permissionUpdater.apply(dbSession, asList(change));
   }
   response.noContent();
 }

}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
```

package org.sonar.server.permission.ws;

import org.sonar.server.ws.WsAction;

public interface PermissionsWsAction extends WsAction {
 // marker interface
}
package org.sonar.server.permission.ws;

import com.google.common.collect.FluentIterable;
import java.util.Set;
import java.util.regex.Pattern;
import java.util.regex.PatternSyntaxException;
import javax.annotation.Nullable;
import org.sonar.api.resources.ResourceType;
import org.sonar.api.resources.ResourceTypes;
import org.sonar.core.permission.GlobalPermissions;

```java
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;

import static com.google.common.base.Strings.isNullOrEmpty;
import static java.lang.String.format;
import static org.apache.commons.lang.StringUtils.isBlank;
import static org.sonar.server.ws.WsUtils.checkRequest;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY_PATTERN;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_QUALIFIER;

public class PermissionRequestValidator {
 public static final String MSG_TEMPLATE_WITH_SAME_NAME = "A template with the name '%s' already
exists (case insensitive).";
 public static final String MSG_TEMPLATE_NAME_NOT_BLANK = "The template name must not be blank";

 private PermissionRequestValidator() {
  // static methods only
 }

 public static String validateProjectPermission(String permission) {
  checkRequest(ProjectPermissions.ALL.contains(permission),
    format("The '%s' parameter for project permissions must be one of %s. '%s' was passed.",
PARAM_PERMISSION, ProjectPermissions.ALL_ON_ONE_LINE, permission));
  return permission;
 }

 public static void validateGlobalPermission(String permission) {
  checkRequest(GlobalPermissions.ALL.contains(permission),
    format("The '%s' parameter for global permissions must be one of %s. '%s' was passed.",
PARAM_PERMISSION, GlobalPermissions.ALL_ON_ONE_LINE, permission));
 }

 public static void validateNotAnyoneAndAdminPermission(String permission, GroupIdOrAnyone group) {
  checkRequest(!GlobalPermissions.SYSTEM_ADMIN.equals(permission) || !group.isAnyone(),
    format("It is not possible to add the '%s' permission to group 'Anyone'.", permission));
 }

 public static void validateTemplateNameFormat(String name) {
  checkRequest(!isBlank(name), MSG_TEMPLATE_NAME_NOT_BLANK);
 }

 public static void validateQualifier(String qualifier, Set<String> rootQualifiers) {
  checkRequest(rootQualifiers.contains(qualifier),
    format("The '%s' parameter must be one of %s. '%s' was passed.", PARAM_QUALIFIER, rootQualifiers,
qualifier));
 }
```

```java
public static void validateQualifier(@Nullable String qualifier, ResourceTypes resourceTypes) {
  if (qualifier == null) {
    return;
  }
  Set<String> rootQualifiers = FluentIterable.from(resourceTypes.getRoots())
    .transform(ResourceType::getQualifier)
    .toSet();
  checkRequest(rootQualifiers.contains(qualifier),
    format("The '%s' parameter must be one of %s. '%s' was passed.", PARAM_QUALIFIER, rootQualifiers,
qualifier));
}

public static void validateProjectPattern(@Nullable String projectPattern) {
  if (isNullOrEmpty(projectPattern)) {
    return;
  }

  try {
    Pattern.compile(projectPattern);
  } catch (PatternSyntaxException e) {
    throw BadRequestException.create(format("The '%s' parameter must be a valid Java regular expression. '%s' was
passed", PARAM_PROJECT_KEY_PATTERN, projectPattern));
  }
}
}
```

package org.sonar.server.permission.ws;

import com.google.common.collect.FluentIterable;

```java
import com.google.common.collect.Iterables;
import com.google.common.collect.Ordering;
import com.google.common.collect.Table;
import java.util.List;
import java.util.Set;
import org.sonar.api.utils.Paging;
import org.sonar.db.component.ComponentDto;

import static com.google.common.base.MoreObjects.firstNonNull;
import static com.google.common.base.Preconditions.checkState;
import static com.google.common.collect.ImmutableList.copyOf;
import static com.google.common.collect.ImmutableTable.copyOf;

class SearchProjectPermissionsData {
 private final List<ComponentDto> rootComponents;
 private final Paging paging;
 private final Table<Long, String, Integer> userCountByProjectIdAndPermission;
 private final Table<Long, String, Integer> groupCountByProjectIdAndPermission;

 private SearchProjectPermissionsData(Builder builder) {
  this.rootComponents = copyOf(builder.projects);
  this.paging = builder.paging;
  this.userCountByProjectIdAndPermission = copyOf(builder.userCountByProjectIdAndPermission);
  this.groupCountByProjectIdAndPermission = copyOf(builder.groupCountByProjectIdAndPermission);
 }

 static Builder newBuilder() {
  return new Builder();
 }

 List<ComponentDto> rootComponents() {
  return rootComponents;
 }

 Paging paging() {
  return paging;
 }

 int userCount(long rootComponentId, String permission) {
  return firstNonNull(userCountByProjectIdAndPermission.get(rootComponentId, permission), 0);
 }

 int groupCount(long rootComponentId, String permission) {
  return firstNonNull(groupCountByProjectIdAndPermission.get(rootComponentId, permission), 0);
 }

 Set<String> permissions(long rootComponentId) {
  return FluentIterable.from(
```

```
    Iterables.concat(
      userCountByProjectIdAndPermission.row(rootComponentId).keySet(),
      groupCountByProjectIdAndPermission.row(rootComponentId).keySet()))
    .toSortedSet(Ordering.natural());
}

static class Builder {
  private List<ComponentDto> projects;
  private Paging paging;
  private Table<Long, String, Integer> userCountByProjectIdAndPermission;
  private Table<Long, String, Integer> groupCountByProjectIdAndPermission;

  private Builder() {
    // prevents instantiation outside main class
  }

  SearchProjectPermissionsData build() {
    checkState(projects != null);
    checkState(userCountByProjectIdAndPermission != null);
    checkState(groupCountByProjectIdAndPermission != null);

    return new SearchProjectPermissionsData(this);
  }

  Builder rootComponents(List<ComponentDto> projects) {
    this.projects = projects;
    return this;
  }

  Builder paging(Paging paging) {
    this.paging = paging;
    return this;
  }

  Builder userCountByProjectIdAndPermission(Table<Long, String, Integer>
userCountByProjectIdAndPermission) {
    this.userCountByProjectIdAndPermission = userCountByProjectIdAndPermission;
    return this;
  }

  Builder groupCountByProjectIdAndPermission(Table<Long, String, Integer>
groupCountByProjectIdAndPermission) {
    this.groupCountByProjectIdAndPermission = groupCountByProjectIdAndPermission;
    return this;
  }
}
}
```

```java
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.sonar.core.platform.Module;
import org.sonar.server.permission.ws.template.AddGroupToTemplateAction;
import org.sonar.server.permission.ws.template.AddProjectCreatorToTemplateAction;
import org.sonar.server.permission.ws.template.AddUserToTemplateAction;
import org.sonar.server.permission.ws.template.ApplyTemplateAction;
import org.sonar.server.permission.ws.template.BulkApplyTemplateAction;
import org.sonar.server.permission.ws.template.CreateTemplateAction;
import org.sonar.server.permission.ws.template.DeleteTemplateAction;
import org.sonar.server.permission.ws.template.RemoveGroupFromTemplateAction;
import org.sonar.server.permission.ws.template.RemoveProjectCreatorFromTemplateAction;
import org.sonar.server.permission.ws.template.RemoveUserFromTemplateAction;
import org.sonar.server.permission.ws.template.SearchTemplatesAction;
import org.sonar.server.permission.ws.template.SetDefaultTemplateAction;
import org.sonar.server.permission.ws.template.TemplateGroupsAction;
import org.sonar.server.permission.ws.template.TemplateUsersAction;
import org.sonar.server.permission.ws.template.UpdateTemplateAction;

public class PermissionsWsModule extends Module {
  @Override
  protected void configureModule() {
    add(
      PermissionsWs.class,
      // actions
      AddGroupAction.class,
      AddUserAction.class,
      RemoveGroupAction.class,
      RemoveUserAction.class,
```

```
            UsersAction.class,
            GroupsAction.class,
            SearchGlobalPermissionsAction.class,
            SearchProjectPermissionsAction.class,
            RemoveUserFromTemplateAction.class,
            AddUserToTemplateAction.class,
            AddGroupToTemplateAction.class,
            AddProjectCreatorToTemplateAction.class,
            RemoveProjectCreatorFromTemplateAction.class,
            RemoveGroupFromTemplateAction.class,
            CreateTemplateAction.class,
            UpdateTemplateAction.class,
            DeleteTemplateAction.class,
            ApplyTemplateAction.class,
            SetDefaultTemplateAction.class,
            SearchTemplatesAction.class,
            TemplateUsersAction.class,
            TemplateGroupsAction.class,
            BulkApplyTemplateAction.class,
            // utility classes
            PermissionWsSupport.class);
    }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import java.util.Optional;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
```

```java
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.permission.PermissionChange;
import org.sonar.server.permission.PermissionUpdater;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.permission.UserId;
import org.sonar.server.permission.UserPermissionChange;
import org.sonar.server.user.UserSession;

import static com.google.common.base.Preconditions.checkArgument;
import static java.util.Collections.singletonList;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkProjectAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createUserLoginParameter;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class AddUserAction implements PermissionsWsAction {

 public static final String ACTION = "add_user";

 private final DbClient dbClient;
 private final UserSession userSession;
 private final PermissionUpdater permissionUpdater;
 private final PermissionWsSupport support;

 public AddUserAction(DbClient dbClient, UserSession userSession, PermissionUpdater permissionUpdater,
PermissionWsSupport support) {
   this.dbClient = dbClient;
   this.userSession = userSession;
   this.permissionUpdater = permissionUpdater;
   this.support = support;
 }

 @Override
 public void define(WebService.NewController context) {
   WebService.NewAction action = context.createAction(ACTION)
     .setDescription("Add permission to a user.<br /> " +
       "This service defaults to global permissions, but can be limited to project permissions by providing project id or
project key.<br />" +
```

```
    "Requires one of the following permissions:" +
    "<ul>" +
    "<li>'Administer System'</li>" +
    "<li>'Administer' rights on the specified project</li>" +
    "</ul>")
    .setSince("5.2")
    .setPost(true)
    .setHandler(this);

  createPermissionParameter(action);
  createUserLoginParameter(action);
  createProjectParameters(action);
  createOrganizationParameter(action)
    .setSince("6.2")
    .setDescription("Key of organization, cannot be used at the same time with %s and %s",
PARAM_PROJECT_ID, PARAM_PROJECT_KEY);
 }

 @Override
 public void handle(Request request, Response response) throws Exception {
  try (DbSession dbSession = dbClient.openSession(false)) {
   UserId user = support.findUser(dbSession, request.mandatoryParam(PARAM_USER_LOGIN));
   Optional<ComponentDto> project = support.findProject(dbSession, request);
   String organizationKey = request.param(PARAM_ORGANIZATION);
   checkArgument(!project.isPresent() || organizationKey == null, "Organization must not be set when project is
set.");
   OrganizationDto org = project
     .map(dto -> dbClient.organizationDao().selectByUuid(dbSession, dto.getOrganizationUuid()))
     .orElseGet(() -> Optional.ofNullable(support.findOrganization(dbSession, organizationKey)))
     .orElseThrow(() -> new NotFoundException(String.format("Organization with key '%s' not found",
organizationKey)));
   support.checkMembership(dbSession, org, user);

   Optional<ProjectId> projectId = project.map(ProjectId::new);
   checkProjectAdmin(userSession, org.getUuid(), projectId);

   PermissionChange change = new UserPermissionChange(
     PermissionChange.Operation.ADD,
     org.getUuid(),
     request.mandatoryParam(PARAM_PERMISSION),
     projectId.orElse(null),
     user);
   permissionUpdater.apply(dbSession, singletonList(change));
  }
  response.noContent();
 }
}
```

```
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import java.util.List;
import java.util.Locale;
import java.util.Optional;

import com.google.common.collect.Collections2;
import com.google.common.collect.Lists;
import com.google.common.collect.Table;
import com.google.common.collect.TreeBasedTable;
import org.sonar.api.i18n.I18n;
import org.sonar.api.resources.ResourceType;
import org.sonar.api.resources.ResourceTypes;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.api.utils.Paging;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentQuery;
import org.sonar.db.permission.CountPerProjectPermission;
import org.sonar.server.permission.PermissionPrivilegeChecker;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Common;
import org.sonarqube.ws.Permissions.Permission;
```

```java
import org.sonarqube.ws.Permissions.SearchProjectPermissionsWsResponse;
import org.sonarqube.ws.Permissions.SearchProjectPermissionsWsResponse.Project;

import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static java.util.Collections.singletonList;
import static org.sonar.api.utils.Paging.forPageIndex;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateQualifier;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonar.server.permission.ws.ProjectWsRef.newOptionalWsProjectRef;
import static org.sonar.server.permission.ws.SearchProjectPermissionsData.newBuilder;
import static org.sonar.server.ws.WsParameterBuilder.createRootQualifierParameter;
import static org.sonar.server.ws.WsParameterBuilder.QualifierParameterContext.newQualifierParameterContext;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_QUALIFIER;

public class SearchProjectPermissionsAction implements PermissionsWsAction {
  private static final String PROPERTY_PREFIX = "projects_role.";
  private static final String DESCRIPTION_SUFFIX = ".desc";

  private final DbClient dbClient;
  private final UserSession userSession;
  private final I18n i18n;
  private final ResourceTypes resourceTypes;
  private final PermissionWsSupport wsSupport;
  private final String[] rootQualifiers;

  public SearchProjectPermissionsAction(DbClient dbClient, UserSession userSession, I18n i18n, ResourceTypes resourceTypes,
    PermissionWsSupport wsSupport) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.i18n = i18n;
    this.resourceTypes = resourceTypes;
    this.wsSupport = wsSupport;
    this.rootQualifiers = Collections2.transform(resourceTypes.getRoots(), ResourceType::getQualifier).toArray(new
String[resourceTypes.getRoots().size()]);
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("search_project_permissions")
      .setDescription("List project permissions. A project can be a technical project, a view or a developer.<br />" +
        "Requires one of the following permissions:" +
        "<ul>" +
```

```java
      "<li>'Administer System'</li>" +
      "<li>'Administer' rights on the specified project</li>" +
      "</ul>")
    .setResponseExample(getClass().getResource("search_project_permissions-example.json"))
    .setSince("5.2")
    .setDeprecatedSince("6.5")
    .addPagingParams(25)
    .setHandler(this);

  action.createParam(Param.TEXT_QUERY)
    .setDescription("Limit search to: <ul>" +
      "<li>project names that contain the supplied string</li>" +
      "<li>project keys that are exactly the same as the supplied string</li>" +
      "</ul>")
    .setExampleValue("apac");
  createProjectParameters(action);
  createRootQualifierParameter(action, newQualifierParameterContext(i18n, resourceTypes))
    .setSince("5.3");
}

@Override
public void handle(Request wsRequest, Response wsResponse) throws Exception {
  SearchProjectPermissionsWsResponse searchProjectPermissionsWsResponse =
doHandle(toSearchProjectPermissionsWsRequest(wsRequest));
  writeProtobuf(searchProjectPermissionsWsResponse, wsRequest, wsResponse);
}

private SearchProjectPermissionsWsResponse doHandle(SearchProjectPermissionsRequest request) {
  try (DbSession dbSession = dbClient.openSession(false)) {
    checkAuthorized(dbSession, request);
    validateQualifier(request.getQualifier(), resourceTypes);
    SearchProjectPermissionsData data = load(dbSession, request);
    return buildResponse(data);
  }
}

private static SearchProjectPermissionsRequest toSearchProjectPermissionsWsRequest(Request request) {
  return new SearchProjectPermissionsRequest()
    .setProjectId(request.param(PARAM_PROJECT_ID))
    .setProjectKey(request.param(PARAM_PROJECT_KEY))
    .setQualifier(request.param(PARAM_QUALIFIER))
    .setPage(request.mandatoryParamAsInt(Param.PAGE))
    .setPageSize(request.mandatoryParamAsInt(Param.PAGE_SIZE))
    .setQuery(request.param(Param.TEXT_QUERY));
}

private void checkAuthorized(DbSession dbSession, SearchProjectPermissionsRequest request) {
  com.google.common.base.Optional<ProjectWsRef> projectRef =
```

```java
newOptionalWsProjectRef(request.getProjectId(), request.getProjectKey());
    if (projectRef.isPresent()) {
      ComponentDto project = wsSupport.getRootComponentOrModule(dbSession, projectRef.get());
      PermissionPrivilegeChecker.checkProjectAdmin(userSession, project.getOrganizationUuid(), Optional.of(new
ProjectId(project)));
    } else {
      userSession.checkLoggedIn().checkIsSystemAdministrator();
    }
  }

  private SearchProjectPermissionsWsResponse buildResponse(SearchProjectPermissionsData data) {
    SearchProjectPermissionsWsResponse.Builder response = SearchProjectPermissionsWsResponse.newBuilder();
    Permission.Builder permissionResponse = Permission.newBuilder();

    Project.Builder rootComponentBuilder = Project.newBuilder();
    for (ComponentDto rootComponent : data.rootComponents()) {
      rootComponentBuilder
        .clear()
        .setId(rootComponent.uuid())
        .setKey(rootComponent.getDbKey())
        .setQualifier(rootComponent.qualifier())
        .setName(rootComponent.name());
      for (String permission : data.permissions(rootComponent.getId())) {
        rootComponentBuilder.addPermissions(
          permissionResponse
            .clear()
            .setKey(permission)
            .setUsersCount(data.userCount(rootComponent.getId(), permission))
            .setGroupsCount(data.groupCount(rootComponent.getId(), permission)));
      }
      response.addProjects(rootComponentBuilder);
    }

    for (String permissionKey : ProjectPermissions.ALL) {
      response.addPermissions(
        permissionResponse
          .clear()
          .setKey(permissionKey)
          .setName(i18nName(permissionKey))
          .setDescription(i18nDescriptionMessage(permissionKey)));
    }

    Paging paging = data.paging();
    response.setPaging(
      Common.Paging.newBuilder()
        .setPageIndex(paging.pageIndex())
        .setPageSize(paging.pageSize())
        .setTotal(paging.total()));
```

```java
    return response.build();
  }

  private String i18nDescriptionMessage(String permissionKey) {
    return i18n.message(Locale.ENGLISH, PROPERTY_PREFIX + permissionKey + DESCRIPTION_SUFFIX, "");
  }

  private String i18nName(String permissionKey) {
    return i18n.message(Locale.ENGLISH, PROPERTY_PREFIX + permissionKey, permissionKey);
  }

  private SearchProjectPermissionsData load(DbSession dbSession, SearchProjectPermissionsRequest request) {
    SearchProjectPermissionsData.Builder data = newBuilder();
    int countRootComponents = countRootComponents(dbSession, request);
    List<ComponentDto> rootComponents = searchRootComponents(dbSession, request, paging(request,
countRootComponents));
    List<Long> rootComponentIds = Lists.transform(rootComponents, ComponentDto::getId);

    data.rootComponents(rootComponents)
        .paging(paging(request, countRootComponents))
        .userCountByProjectIdAndPermission(userCountByRootComponentIdAndPermission(dbSession,
rootComponentIds))
        .groupCountByProjectIdAndPermission(groupCountByRootComponentIdAndPermission(dbSession,
rootComponentIds));

    return data.build();
  }

  private static Paging paging(SearchProjectPermissionsRequest request, int total) {
    return forPageIndex(request.getPage())
        .withPageSize(request.getPageSize())
        .andTotal(total);
  }

  private int countRootComponents(DbSession dbSession, SearchProjectPermissionsRequest request) {
    return dbClient.componentDao().countByQuery(dbSession, toDbQuery(request));
  }

  private List<ComponentDto> searchRootComponents(DbSession dbSession, SearchProjectPermissionsRequest
request, Paging paging) {
    com.google.common.base.Optional<ProjectWsRef> project = newOptionalWsProjectRef(request.getProjectId(),
request.getProjectKey());

    if (project.isPresent()) {
      return singletonList(wsSupport.getRootComponentOrModule(dbSession, project.get()));
    }
```

```java
    return dbClient.componentDao().selectByQuery(dbSession, toDbQuery(request), paging.offset(),
  paging.pageSize());
  }

  private ComponentQuery toDbQuery(SearchProjectPermissionsRequest wsRequest) {
    return ComponentQuery.builder()
        .setQualifiers(qualifiers(wsRequest.getQualifier()))
        .setNameOrKeyQuery(wsRequest.getQuery())
        .build();
  }

  private String[] qualifiers(@Nullable String requestQualifier) {
    return requestQualifier == null
        ? rootQualifiers
        : (new String[] {requestQualifier});
  }

  private Table<Long, String, Integer> userCountByRootComponentIdAndPermission(DbSession dbSession,
  List<Long> rootComponentIds) {
    final Table<Long, String, Integer> userCountByRootComponentIdAndPermission = TreeBasedTable.create();

    dbClient.userPermissionDao().countUsersByProjectPermission(dbSession, rootComponentIds).forEach(
        row -> userCountByRootComponentIdAndPermission.put(row.getComponentId(), row.getPermission(),
  row.getCount()));

    return userCountByRootComponentIdAndPermission;
  }

  private Table<Long, String, Integer> groupCountByRootComponentIdAndPermission(DbSession dbSession,
  List<Long> rootComponentIds) {
    final Table<Long, String, Integer> userCountByRootComponentIdAndPermission = TreeBasedTable.create();

    dbClient.groupPermissionDao().groupsCountByComponentIdAndPermission(dbSession, rootComponentIds,
  context -> {
      CountPerProjectPermission row = (CountPerProjectPermission) context.getResultObject();
      userCountByRootComponentIdAndPermission.put(row.getComponentId(), row.getPermission(),
  row.getCount());
    });

    return userCountByRootComponentIdAndPermission;
  }

  private static class SearchProjectPermissionsRequest {
    private String projectId;
    private String projectKey;
    private String qualifier;
    private Integer page;
    private Integer pageSize;
```

```java
private String query;

@CheckForNull
public String getProjectId() {
  return projectId;
}

public SearchProjectPermissionsRequest setProjectId(@Nullable String projectId) {
  this.projectId = projectId;
  return this;
}

@CheckForNull
public String getProjectKey() {
  return projectKey;
}

public SearchProjectPermissionsRequest setProjectKey(@Nullable String projectKey) {
  this.projectKey = projectKey;
  return this;
}

@CheckForNull
public Integer getPage() {
  return page;
}

public SearchProjectPermissionsRequest setPage(int page) {
  this.page = page;
  return this;
}

@CheckForNull
public Integer getPageSize() {
  return pageSize;
}

public SearchProjectPermissionsRequest setPageSize(int pageSize) {
  this.pageSize = pageSize;
  return this;
}

@CheckForNull
public String getQuery() {
  return query;
}

public SearchProjectPermissionsRequest setQuery(@Nullable String query) {
```

```java
    this.query = query;
    return this;
  }

  @CheckForNull
  public String getQualifier() {
    return qualifier;
  }

  public SearchProjectPermissionsRequest setQualifier(@Nullable String qualifier) {
    this.qualifier = qualifier;
    return this;
  }
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import java.util.Optional;
import javax.annotation.Nullable;
import org.sonar.api.server.ws.Request;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.component.ComponentFinder;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.permission.UserId;
```

```java
import org.sonar.server.permission.ws.template.WsTemplateRef;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;
import org.sonar.server.usergroups.ws.GroupWsRef;
import org.sonar.server.usergroups.ws.GroupWsSupport;
import org.sonarqube.ws.client.permission.PermissionsWsParameters;

import static com.google.common.base.Preconditions.checkArgument;
import static org.sonar.server.ws.WsUtils.checkFound;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;

public class PermissionWsSupport {

  private final DbClient dbClient;
  private final ComponentFinder componentFinder;
  private final GroupWsSupport groupWsSupport;

  public PermissionWsSupport(DbClient dbClient, ComponentFinder componentFinder, GroupWsSupport
groupWsSupport) {
    this.dbClient = dbClient;
    this.componentFinder = componentFinder;
    this.groupWsSupport = groupWsSupport;
  }

  public OrganizationDto findOrganization(DbSession dbSession, @Nullable String organizationKey) {
    return groupWsSupport.findOrganizationByKey(dbSession, organizationKey);
  }

  public Optional<ProjectId> findProjectId(DbSession dbSession, Request request) {
    return findProject(dbSession, request)
      .map(ProjectId::new);
  }

  public Optional<ComponentDto> findProject(DbSession dbSession, Request request) {
    String uuid = request.param(PermissionsWsParameters.PARAM_PROJECT_ID);
    String key = request.param(PermissionsWsParameters.PARAM_PROJECT_KEY);
    if (uuid != null || key != null) {
      ProjectWsRef ref = ProjectWsRef.newWsProjectRef(uuid, key);
      return Optional.of(componentFinder.getRootComponentByUuidOrKey(dbSession, ref.uuid(), ref.key()));
    }
    return Optional.empty();
  }

  public ComponentDto getRootComponentOrModule(DbSession dbSession, ProjectWsRef projectRef) {
    return componentFinder.getRootComponentByUuidOrKey(dbSession, projectRef.uuid(), projectRef.key());
  }
```

```java
public GroupIdOrAnyone findGroup(DbSession dbSession, Request request) {
  Integer groupId = request.paramAsInt(PARAM_GROUP_ID);
  String orgKey = request.param(PARAM_ORGANIZATION);
  String groupName = request.param(PARAM_GROUP_NAME);
  GroupWsRef groupRef = GroupWsRef.create(groupId, orgKey, groupName);
  return groupWsSupport.findGroupOrAnyone(dbSession, groupRef);
}

public UserId findUser(DbSession dbSession, String login) {
  UserDto dto = dbClient.userDao().selectActiveUserByLogin(dbSession, login);
  checkFound(dto, "User with login '%s' is not found'", login);
  return new UserId(dto.getId(), dto.getLogin());
}

public PermissionTemplateDto findTemplate(DbSession dbSession, WsTemplateRef ref) {
  if (ref.uuid() != null) {
    return checkFound(
      dbClient.permissionTemplateDao().selectByUuid(dbSession, ref.uuid()),
      "Permission template with id '%s' is not found", ref.uuid());
  } else {
    OrganizationDto org = findOrganization(dbSession, ref.getOrganization());
    return checkFound(
      dbClient.permissionTemplateDao().selectByName(dbSession, org.getUuid(), ref.name()),
      "Permission template with name '%s' is not found (case insensitive) in organization with key '%s'", ref.name(),
org.getKey());
  }
}

public void checkMembership(DbSession dbSession, OrganizationDto organization, UserId user) {
  checkArgument(dbClient.organizationMemberDao().select(dbSession, organization.getUuid(),
user.getId()).isPresent(),
    "User '%s' is not member of organization '%s'", user.getLogin(), organization.getKey());
}
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
```

```java
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
package org.sonar.server.permission.ws;

import org.sonar.api.server.ws.WebService;

import static org.sonarqube.ws.client.permission.PermissionsWsParameters.CONTROLLER;

public class PermissionsWs implements WebService {
 private final PermissionsWsAction[] actions;

 public PermissionsWs(PermissionsWsAction... actions) {
   this.actions = actions;
 }

 @Override
 public void define(Context context) {
   NewController controller = context.createController(CONTROLLER);
   controller.setDescription("Manage permission templates, and the granting and revoking of permissions at the
global and project levels.");
   controller.setSince("3.7");

   for (PermissionsWsAction action : actions) {
     action.define(controller);
   }

   controller.done();
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
```

```
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import java.util.Locale;
import org.sonar.api.i18n.I18n;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.OrganizationPermission;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions.Permission;
import org.sonarqube.ws.Permissions.WsSearchGlobalPermissionsResponse;

import static org.sonar.server.permission.PermissionPrivilegeChecker.checkGlobalAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.Permissions.Permission.newBuilder;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;

public class SearchGlobalPermissionsAction implements PermissionsWsAction {

  public static final String ACTION = "search_global_permissions";
  private static final String PROPERTY_PREFIX = "global_permissions.";
  private static final String DESCRIPTION_SUFFIX = ".desc";

  private final DbClient dbClient;
  private final UserSession userSession;
  private final I18n i18n;
  private final PermissionWsSupport support;

  public SearchGlobalPermissionsAction(DbClient dbClient, UserSession userSession, I18n i18n,
PermissionWsSupport support) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.i18n = i18n;
    this.support = support;
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction(ACTION)
      .setDescription("List global permissions. <br />" +
```

```java
        "Requires the following permission: 'Administer System'")
      .setResponseExample(getClass().getResource("search_global_permissions-example.json"))
      .setSince("5.2")
      .setDeprecatedSince("6.5")
      .setHandler(this);

    createOrganizationParameter(action).setSince("6.2");
  }

  @Override
  public void handle(Request wsRequest, Response wsResponse) throws Exception {
    try (DbSession dbSession = dbClient.openSession(false)) {
      OrganizationDto org = support.findOrganization(dbSession, wsRequest.param(PARAM_ORGANIZATION));
      checkGlobalAdmin(userSession, org.getUuid());

      WsSearchGlobalPermissionsResponse response = buildResponse(dbSession, org);
      writeProtobuf(response, wsRequest, wsResponse);
    }
  }

  private WsSearchGlobalPermissionsResponse buildResponse(DbSession dbSession, OrganizationDto org) {
    WsSearchGlobalPermissionsResponse.Builder response = WsSearchGlobalPermissionsResponse.newBuilder();
    Permission.Builder permission = newBuilder();

    OrganizationPermission.all()
      .map(OrganizationPermission::getKey)
      .forEach(permissionKey -> {
        PermissionQuery query = permissionQuery(permissionKey, org);
        response.addPermissions(
          permission
            .clear()
            .setKey(permissionKey)
            .setName(i18nName(permissionKey))
            .setDescription(i18nDescriptionMessage(permissionKey))
            .setUsersCount(countUsers(dbSession, query))
            .setGroupsCount(countGroups(dbSession, org, permissionKey)));
      });

    return response.build();
  }

  private String i18nDescriptionMessage(String permissionKey) {
    return i18n.message(Locale.ENGLISH, PROPERTY_PREFIX + permissionKey + DESCRIPTION_SUFFIX, "");
  }

  private String i18nName(String permissionKey) {
    return i18n.message(Locale.ENGLISH, PROPERTY_PREFIX + permissionKey, permissionKey);
  }
```

```
  private int countGroups(DbSession dbSession, OrganizationDto org, String permission) {
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(org.getUuid()).setPermission(permission).build();
    return dbClient.groupPermissionDao().countGroupsByQuery(dbSession, query);
  }

  private int countUsers(DbSession dbSession, PermissionQuery permissionQuery) {
    return dbClient.userPermissionDao().countUsersByQuery(dbSession, permissionQuery);
  }

  private static PermissionQuery permissionQuery(String permissionKey, OrganizationDto org) {
    return PermissionQuery.builder()
      .setOrganizationUuid(org.getUuid())
      .setPermission(permissionKey)
      .withAtLeastOnePermission()
      .build();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import com.google.common.base.Optional;
import javax.annotation.CheckForNull;
import javax.annotation.Nullable;

import static org.sonar.server.ws.WsUtils.checkRequest;

/**
 * Reference to a project <b>as defined by web service callers</b>. It allows to reference a project
```

* by its (functional) key or by its (technical) id. It's then converted to { @link org.sonar.server.permission.ProjectId}.
*
* <p>Factory methods guarantee that the project id and project key are not provided at the same time.</p>
*/
public class ProjectWsRef {
 private static final String MSG_ID_OR_KEY_MUST_BE_PROVIDED = "Project id or project key can be provided, not both.";
 private final String uuid;
 private final String key;

 private ProjectWsRef(@Nullable String uuid, @Nullable String key) {
  this.uuid = uuid;
  this.key = key;
  checkRequest(this.uuid != null ^ this.key != null, MSG_ID_OR_KEY_MUST_BE_PROVIDED);
 }

 public static Optional<ProjectWsRef> newOptionalWsProjectRef(@Nullable String uuid, @Nullable String key) {
  if (uuid == null && key == null) {
   return Optional.absent();
  }

  return Optional.of(new ProjectWsRef(uuid, key));
 }

 public static ProjectWsRef newWsProjectRef(@Nullable String uuid, @Nullable String key) {
  return new ProjectWsRef(uuid, key);
 }

 @CheckForNull
 public String uuid() {
  return this.uuid;
 }

 @CheckForNull
 public String key() {
  return this.key;
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*

```
package org.sonar.server.permission.ws;

import com.google.common.collect.Multimap;
import com.google.common.collect.Ordering;
import com.google.common.collect.TreeMultimap;
import com.google.common.io.Resources;
import java.util.List;
import java.util.Optional;
import org.sonar.api.security.DefaultGroups;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.api.utils.Paging;
import org.sonar.core.util.stream.MoreCollectors;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.GroupPermissionDto;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.user.GroupDto;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions.Group;
import org.sonarqube.ws.Permissions.WsGroupsResponse;

import static java.util.Collections.emptyList;
import static org.sonar.core.util.Protobuf.setNullable;
import static org.sonar.db.permission.PermissionQuery.DEFAULT_PAGE_SIZE;
import static org.sonar.db.permission.PermissionQuery.RESULTS_MAX_SIZE;
import static org.sonar.db.permission.PermissionQuery.SEARCH_QUERY_MIN_LENGTH;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkProjectAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
```

```java
public class GroupsAction implements PermissionsWsAction {
  private final DbClient dbClient;
  private final UserSession userSession;
  private final PermissionWsSupport support;

  public GroupsAction(DbClient dbClient, UserSession userSession, PermissionWsSupport support) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.support = support;
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("groups")
      .setSince("5.2")
      .setInternal(true)
      .setDescription("Lists the groups with their permissions.<br>" +
        "This service defaults to global permissions, but can be limited to project permissions by providing project id or
project key.<br> " +
        "This service defaults to all groups, but can be limited to groups with a specific permission by providing the
desired permission.<br>" +
        "Requires one of the following permissions:" +
        "<ul>" +
        "<li>'Administer System'</li>" +
        "<li>'Administer' rights on the specified project</li>" +
        "</ul>")
      .addPagingParams(DEFAULT_PAGE_SIZE, RESULTS_MAX_SIZE)
      .setResponseExample(Resources.getResource(getClass(), "groups-example.json"))
      .setHandler(this);

    action.createSearchQuery("sonar", "names")
      .setDescription("Limit search to group names that contain the supplied string. When this parameter is not set,
only groups having at least one permission are returned.")
      .setMinimumLength(SEARCH_QUERY_MIN_LENGTH);

    createOrganizationParameter(action).setSince("6.2");
    createPermissionParameter(action).setRequired(false);
    createProjectParameters(action);
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    try (DbSession dbSession = dbClient.openSession(false)) {
      OrganizationDto org = support.findOrganization(dbSession, request.param(PARAM_ORGANIZATION));
      Optional<ProjectId> projectId = support.findProjectId(dbSession, request);
      checkProjectAdmin(userSession, org.getUuid(), projectId);

      PermissionQuery query = buildPermissionQuery(request, org, projectId);
```

```java
    // TODO validatePermission(groupsRequest.getPermission(), wsProjectRef);
    List<GroupDto> groups = findGroups(dbSession, org, query);
    int total = dbClient.groupPermissionDao().countGroupsByQuery(dbSession, query);
    List<GroupPermissionDto> groupsWithPermission = findGroupPermissions(dbSession, org, groups, projectId);
    Paging paging =
Paging.forPageIndex(request.mandatoryParamAsInt(Param.PAGE)).withPageSize(query.getPageSize()).andTotal(total);
    WsGroupsResponse groupsResponse = buildResponse(groups, groupsWithPermission, paging);
    writeProtobuf(groupsResponse, request, response);
  }
}

 private static PermissionQuery buildPermissionQuery(Request request, OrganizationDto org, Optional<ProjectId>
project) {
  String textQuery = request.param(Param.TEXT_QUERY);
  PermissionQuery.Builder permissionQuery = PermissionQuery.builder()
    .setOrganizationUuid(org.getUuid())
    .setPermission(request.param(PARAM_PERMISSION))
    .setPageIndex(request.mandatoryParamAsInt(Param.PAGE))
    .setPageSize(request.mandatoryParamAsInt(Param.PAGE_SIZE))
    .setSearchQuery(textQuery);
  if (project.isPresent()) {
   permissionQuery.setComponentUuid(project.get().getUuid());
  }
  if (textQuery == null) {
   permissionQuery.withAtLeastOnePermission();
  }
  return permissionQuery.build();
 }

 private static WsGroupsResponse buildResponse(List<GroupDto> groups, List<GroupPermissionDto>
groupPermissions, Paging paging) {
  Multimap<Integer, String> permissionsByGroupId = TreeMultimap.create();
  groupPermissions.forEach(groupPermission -> permissionsByGroupId.put(groupPermission.getGroupId(),
groupPermission.getRole()));
  WsGroupsResponse.Builder response = WsGroupsResponse.newBuilder();

  groups.forEach(group -> {
   Group.Builder wsGroup = response.addGroupsBuilder()
     .setName(group.getName());
   if (group.getId() != 0) {
    wsGroup.setId(String.valueOf(group.getId()));
   }
   setNullable(group.getDescription(), wsGroup::setDescription);
   wsGroup.addAllPermissions(permissionsByGroupId.get(group.getId()));
  });

  response.getPagingBuilder()
```

```java
        .setPageIndex(paging.pageIndex())
        .setPageSize(paging.pageSize())
        .setTotal(paging.total());

    return response.build();
  }

  private List<GroupDto> findGroups(DbSession dbSession, OrganizationDto org, PermissionQuery dbQuery) {
    List<String> orderedNames = dbClient.groupPermissionDao().selectGroupNamesByQuery(dbSession, dbQuery);
    List<GroupDto> groups = dbClient.groupDao().selectByNames(dbSession, org.getUuid(), orderedNames);
    if (orderedNames.contains(DefaultGroups.ANYONE)) {
      groups.add(0, new
GroupDto().setId(0).setName(DefaultGroups.ANYONE).setOrganizationUuid(org.getUuid()));
    }
    return Ordering.explicit(orderedNames).onResultOf(GroupDto::getName).immutableSortedCopy(groups);
  }

  private List<GroupPermissionDto> findGroupPermissions(DbSession dbSession, OrganizationDto org,
List<GroupDto> groups, Optional<ProjectId> project) {
    if (groups.isEmpty()) {
      return emptyList();
    }
    List<Integer> ids = groups.stream().map(GroupDto::getId).collect(MoreCollectors.toList(groups.size()));
    return dbClient.groupPermissionDao().selectByGroupIds(dbSession, org.getUuid(), ids, project.isPresent() ?
project.get().getId() : null);
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;
```

```java
import java.util.Optional;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.server.permission.GroupPermissionChange;
import org.sonar.server.permission.PermissionChange;
import org.sonar.server.permission.PermissionUpdater;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.user.UserSession;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;

import static java.util.Arrays.asList;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkProjectAdmin;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupIdParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createGroupNameParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;

public class RemoveGroupAction implements PermissionsWsAction {

  public static final String ACTION = "remove_group";

  private final DbClient dbClient;
  private final UserSession userSession;
  private final PermissionUpdater permissionUpdater;
  private final PermissionWsSupport support;

  public RemoveGroupAction(DbClient dbClient, UserSession userSession, PermissionUpdater permissionUpdater,
  PermissionWsSupport support) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.permissionUpdater = permissionUpdater;
    this.support = support;
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction(ACTION)
      .setDescription("Remove a permission from a group.<br /> " +
        "This service defaults to global permissions, but can be limited to project permissions by providing project id or
project key.<br /> " +
        "The group id or group name must be provided, not both.<br />" +
        "Requires one of the following permissions:" +
        "<ul>" +
```

```java
      "<li>'Administer System'</li>" +
      "<li>'Administer' rights on the specified project</li>" +
      "</ul>")
    .setSince("5.2")
    .setPost(true)
    .setHandler(this);

  createPermissionParameter(action);
  createOrganizationParameter(action).setSince("6.2");
  createGroupNameParameter(action);
  createGroupIdParameter(action);
  createProjectParameters(action);
}

@Override
public void handle(Request request, Response response) throws Exception {
  try (DbSession dbSession = dbClient.openSession(false)) {
    GroupIdOrAnyone group = support.findGroup(dbSession, request);
    Optional<ProjectId> projectId = support.findProjectId(dbSession, request);

    checkProjectAdmin(userSession, group.getOrganizationUuid(), projectId);

    PermissionChange change = new GroupPermissionChange(
      PermissionChange.Operation.REMOVE,
      request.mandatoryParam(PARAM_PERMISSION),
      projectId.orElse(null),
      group);
    permissionUpdater.apply(dbSession, asList(change));
  }
  response.noContent();
}
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
```

```
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
package org.sonar.server.permission.ws;

import com.google.common.collect.Multimap;
import com.google.common.collect.Ordering;
import com.google.common.collect.TreeMultimap;
import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;
import org.sonar.api.server.ws.Request;
import org.sonar.api.server.ws.Response;
import org.sonar.api.server.ws.WebService;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.api.utils.Paging;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.UserPermissionDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.issue.ws.AvatarResolver;
import org.sonar.server.permission.ProjectId;
import org.sonar.server.user.UserSession;
import org.sonarqube.ws.Permissions;
import org.sonarqube.ws.Permissions.UsersWsResponse;

import static com.google.common.base.Strings.emptyToNull;
import static java.util.Collections.emptyList;
import static org.sonar.core.util.Protobuf.setNullable;
import static org.sonar.db.permission.PermissionQuery.DEFAULT_PAGE_SIZE;
import static org.sonar.db.permission.PermissionQuery.RESULTS_MAX_SIZE;
import static org.sonar.db.permission.PermissionQuery.SEARCH_QUERY_MIN_LENGTH;
import static org.sonar.server.permission.PermissionPrivilegeChecker.checkProjectAdmin;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateGlobalPermission;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateProjectPermission;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createOrganizationParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createPermissionParameter;
import static org.sonar.server.permission.ws.PermissionsWsParametersBuilder.createProjectParameters;
import static org.sonar.server.ws.WsUtils.writeProtobuf;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;

public class UsersAction implements PermissionsWsAction {

  private final DbClient dbClient;
  private final UserSession userSession;
```

```java
  private final PermissionWsSupport support;
  private final AvatarResolver avatarResolver;

  public UsersAction(DbClient dbClient, UserSession userSession, PermissionWsSupport support, AvatarResolver
avatarResolver) {
    this.dbClient = dbClient;
    this.userSession = userSession;
    this.support = support;
    this.avatarResolver = avatarResolver;
  }

  @Override
  public void define(WebService.NewController context) {
    WebService.NewAction action = context.createAction("users")
      .setSince("5.2")
      .setDescription("Lists the users with their permissions as individual users rather than through group
affiliation.<br>" +
        "This service defaults to global permissions, but can be limited to project permissions by providing project id or
project key.<br> " +
        "This service defaults to all users, but can be limited to users with a specific permission by providing the desired
permission.<br>" +
        "Requires one of the following permissions:" +
        "<ul>" +
        "<li>'Administer System'</li>" +
        "<li>'Administer' rights on the specified project</li>" +
        "</ul>")
      .addPagingParams(DEFAULT_PAGE_SIZE, RESULTS_MAX_SIZE)
      .setInternal(true)
      .setResponseExample(getClass().getResource("users-example.json"))
      .setHandler(this);

    action.createParam(Param.TEXT_QUERY)
      .setMinimumLength(SEARCH_QUERY_MIN_LENGTH)
      .setDescription("Limit search to user names that contain the supplied string. <br/>" +
        "When this parameter is not set, only users having at least one permission are returned.")
      .setExampleValue("eri");

    createOrganizationParameter(action).setSince("6.2");
    createPermissionParameter(action).setRequired(false);
    createProjectParameters(action);
  }

  @Override
  public void handle(Request request, Response response) throws Exception {
    try (DbSession dbSession = dbClient.openSession(false)) {
      OrganizationDto org = support.findOrganization(dbSession, request.param(PARAM_ORGANIZATION));
      Optional<ProjectId> projectId = support.findProjectId(dbSession, request);
      checkProjectAdmin(userSession, org.getUuid(), projectId);
```

```java
    PermissionQuery query = buildPermissionQuery(request, org, projectId);
    List<UserDto> users = findUsers(dbSession, query);
    int total = dbClient.userPermissionDao().countUsersByQuery(dbSession, query);
    List<UserPermissionDto> userPermissions = findUserPermissions(dbSession, org, users, projectId);
    Paging paging =
Paging.forPageIndex(request.mandatoryParamAsInt(Param.PAGE)).withPageSize(query.getPageSize()).andTotal(to
tal);
    UsersWsResponse usersWsResponse = buildResponse(users, userPermissions, paging);
    writeProtobuf(usersWsResponse, request, response);
  }
 }

 private static PermissionQuery buildPermissionQuery(Request request, OrganizationDto organization,
Optional<ProjectId> project) {
  String textQuery = request.param(Param.TEXT_QUERY);
  String permission = request.param(PARAM_PERMISSION);
  PermissionQuery.Builder permissionQuery = PermissionQuery.builder()
    .setOrganizationUuid(organization.getUuid())
    .setPermission(permission)
    .setPageIndex(request.mandatoryParamAsInt(Param.PAGE))
    .setPageSize(request.mandatoryParamAsInt(Param.PAGE_SIZE))
    .setSearchQuery(textQuery);
  project.ifPresent(projectId -> permissionQuery.setComponentUuid(projectId.getUuid()));
  if (permission != null) {
   if (project.isPresent()) {
    validateProjectPermission(permission);
   } else {
    validateGlobalPermission(permission);
   }
  }
  if (textQuery == null) {
   permissionQuery.withAtLeastOnePermission();
  }
  return permissionQuery.build();
 }

 private UsersWsResponse buildResponse(List<UserDto> users, List<UserPermissionDto> userPermissions, Paging
paging) {
  Multimap<Integer, String> permissionsByUserId = TreeMultimap.create();
  userPermissions.forEach(userPermission -> permissionsByUserId.put(userPermission.getUserId(),
userPermission.getPermission()));

  UsersWsResponse.Builder response = UsersWsResponse.newBuilder();
  users.forEach(user -> {
   Permissions.User.Builder userResponse = response.addUsersBuilder()
     .setLogin(user.getLogin())
     .addAllPermissions(permissionsByUserId.get(user.getId()));
```

```java
      setNullable(user.getEmail(), userResponse::setEmail);
      setNullable(emptyToNull(user.getEmail()), u -> userResponse.setAvatar(avatarResolver.create(user)));
      setNullable(user.getName(), userResponse::setName);
    });

    response.getPagingBuilder()
      .setPageIndex(paging.pageIndex())
      .setPageSize(paging.pageSize())
      .setTotal(paging.total())
      .build();

    return response.build();
  }

  private List<UserDto> findUsers(DbSession dbSession, PermissionQuery query) {
    List<Integer> orderedIds = dbClient.userPermissionDao().selectUserIdsByQuery(dbSession, query);
    return
Ordering.explicit(orderedIds).onResultOf(UserDto::getId).immutableSortedCopy(dbClient.userDao().selectByIds(d
bSession, orderedIds));
  }

  private List<UserPermissionDto> findUserPermissions(DbSession dbSession, OrganizationDto org, List<UserDto>
users, Optional<ProjectId> project) {
    if (users.isEmpty()) {
      return emptyList();
    }
    List<Integer> userIds = users.stream().map(UserDto::getId).collect(Collectors.toList());
    PermissionQuery query = PermissionQuery.builder()
      .setOrganizationUuid(org.getUuid())
      .setComponentUuid(project.map(ProjectId::getUuid).orElse(null))
      .withAtLeastOnePermission()
      .build();
    return dbClient.userPermissionDao().selectUserPermissionsByQuery(dbSession, query, userIds);
  }
}
```

```java
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import java.util.Optional;
import org.sonar.db.DbSession;
import org.sonar.server.es.ProjectIndexer;
import org.sonar.server.es.ProjectIndexers;

/**
 * Add or remove global/project permissions to a group. This class
 * does not verify that caller has administration right on the related
 * organization or project.
 */
public class PermissionUpdater {

  private final ProjectIndexers projectIndexers;
  private final UserPermissionChanger userPermissionChanger;
  private final GroupPermissionChanger groupPermissionChanger;

  public PermissionUpdater(ProjectIndexers projectIndexers,
    UserPermissionChanger userPermissionChanger, GroupPermissionChanger groupPermissionChanger) {
    this.projectIndexers = projectIndexers;
    this.userPermissionChanger = userPermissionChanger;
    this.groupPermissionChanger = groupPermissionChanger;
  }

  public void apply(DbSession dbSession, Collection<PermissionChange> changes) {
    List<String> projectOrViewUuids = new ArrayList<>();
    for (PermissionChange change : changes) {
      boolean changed = doApply(dbSession, change);
      Optional<ProjectId> projectId = change.getProjectId();
      if (changed && projectId.isPresent()) {
        projectOrViewUuids.add(projectId.get().getUuid());
      }
    }
    projectIndexers.commitAndIndexByProjectUuids(dbSession, projectOrViewUuids,
ProjectIndexer.Cause.PERMISSION_CHANGE);
  }

  private boolean doApply(DbSession dbSession, PermissionChange change) {
```

```
  if (change instanceof UserPermissionChange) {
    return userPermissionChanger.apply(dbSession, (UserPermissionChange) change);
  }
  if (change instanceof GroupPermissionChange) {
    return groupPermissionChanger.apply(dbSession, (GroupPermissionChange) change);
  }
  throw new UnsupportedOperationException("Unsupported permission change: " + change.getClass());


 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
@ParametersAreNonnullByDefault
package org.sonar.server.permission.index;

import javax.annotation.ParametersAreNonnullByDefault;
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
```

package org.sonar.server.permission.index;

/**
* An {@link NeedAuthorizationIndexer} defines how
* a {@link org.sonar.server.es.ProjectIndexer} populates
* the type named {@link AuthorizationTypeSupport#TYPE_AUTHORIZATION}, which
* is used to verify that a user can access to projects.
*/
public interface NeedAuthorizationIndexer {

 /**
  * Returns the metadata required by {@link PermissionIndexer} to
  * populate "authorization" types.
  */
 AuthorizationScope getAuthorizationScope();

}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
*/
package org.sonar.server.permission.index;

import java.util.function.Predicate;
import javax.annotation.concurrent.Immutable;
import org.sonar.server.es.IndexType;

import static java.util.Objects.requireNonNull;

```java
@Immutable
public final class AuthorizationScope {
 private final IndexType indexType;
 private final Predicate<PermissionIndexerDao.Dto> projectPredicate;

 public AuthorizationScope(IndexType indexType, Predicate<PermissionIndexerDao.Dto> projectPredicate) {
   this.indexType = AuthorizationTypeSupport.getAuthorizationIndexType(indexType);
   this.projectPredicate = requireNonNull(projectPredicate);
 }

 /**
  * Identifier of the authorization type (in the same index than the original IndexType, passed into the constructor).
  */
 public IndexType getIndexType() {
   return indexType;
 }

 /**
  * Predicates that filters the projects to be involved in
  * authorization.
  */
 public Predicate<PermissionIndexerDao.Dto> getProjectPredicate() {
   return projectPredicate;
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;

import com.google.common.collect.ImmutableList;
import java.sql.PreparedStatement;
```

```java
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import org.apache.commons.lang.StringUtils;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;

import static org.apache.commons.lang.StringUtils.repeat;
import static org.sonar.db.DatabaseUtils.executeLargeInputs;

/**
 * No streaming because of union of joins -> no need to use ResultSetIterator
 */
public class PermissionIndexerDao {

 public static final class Dto {
   private final String projectUuid;
   private final String qualifier;
   private final List<Integer> userIds = new ArrayList<>();
   private final List<Integer> groupIds = new ArrayList<>();
   private boolean allowAnyone = false;

   public Dto(String projectUuid, String qualifier) {
     this.projectUuid = projectUuid;
     this.qualifier = qualifier;
   }

   public String getProjectUuid() {
     return projectUuid;
   }

   public String getQualifier() {
     return qualifier;
   }

   public List<Integer> getUserIds() {
     return userIds;
   }

   public Dto addUserId(int l) {
     userIds.add(l);
     return this;
   }
```

```java
public Dto addGroupId(int id) {
  groupIds.add(id);
  return this;
}

public List<Integer> getGroupIds() {
  return groupIds;
}

public void allowAnyone() {
  this.allowAnyone = true;
}

public boolean isAllowAnyone() {
  return allowAnyone;
}
}

private enum RowKind {
  USER, GROUP, ANYONE, NONE
}

private static final String SQL_TEMPLATE = "SELECT " +
  " project_authorization.kind as kind, " +
  " project_authorization.project as project, " +
  " project_authorization.user_id as user_id, " +
  " project_authorization.group_id as group_id, " +
  " project_authorization.qualifier as qualifier " +
  "FROM ( " +

  // users

  "    SELECT '" + RowKind.USER + "' as kind," +
  "    projects.uuid AS project, " +
  "    projects.qualifier AS qualifier, " +
  "    user_roles.user_id  AS user_id, " +
  "    NULL  AS group_id " +
  "    FROM projects " +
  "    INNER JOIN user_roles ON user_roles.resource_id = projects.id AND user_roles.role = 'user' " +
  "    WHERE " +
  "      (projects.qualifier = 'TRK' " +
  "       or  projects.qualifier = 'VW' " +
  "       or  projects.qualifier = 'APP') " +
  "      AND projects.copy_component_uuid is NULL " +
  "      {projectsCondition} " +
  "    UNION " +
```

// groups

```
"     SELECT '" + RowKind.GROUP + "' as kind," +
"     projects.uuid AS project, " +
"     projects.qualifier AS qualifier, " +
"     NULL  AS user_id, " +
"     groups.id  AS group_id " +
"     FROM projects " +
"     INNER JOIN group_roles ON group_roles.resource_id = projects.id AND group_roles.role = 'user' " +
"     INNER JOIN groups ON groups.id = group_roles.group_id " +
"     WHERE " +
"       (projects.qualifier = 'TRK' " +
"        or  projects.qualifier = 'VW' " +
"        or  projects.qualifier = 'APP') " +
"       AND projects.copy_component_uuid is NULL " +
"       {projectsCondition} " +
"       AND group_id IS NOT NULL " +
"     UNION " +
```

// public projects are accessible to any one

```
"     SELECT '" + RowKind.ANYONE + "' as kind," +
"     projects.uuid AS project, " +
"     projects.qualifier AS qualifier, " +
"     NULL       AS user_id, " +
"     NULL    AS group_id " +
"     FROM projects " +
"     WHERE " +
"       (projects.qualifier = 'TRK' " +
"        or  projects.qualifier = 'VW' " +
"        or  projects.qualifier = 'APP') " +
"       AND projects.copy_component_uuid is NULL " +
"       AND projects.private = ? " +
"       {projectsCondition} " +
"     UNION " +
```

// private project is returned when no authorization
```
"     SELECT '" + RowKind.NONE + "' as kind," +
"     projects.uuid AS project, " +
"     projects.qualifier AS qualifier, " +
"     NULL AS user_id, " +
"     NULL  AS group_id " +
"     FROM projects " +
"     WHERE " +
"       (projects.qualifier = 'TRK' " +
"        or  projects.qualifier = 'VW' " +
"        or  projects.qualifier = 'APP') " +
"       AND projects.copy_component_uuid is NULL " +
```

```
"    AND projects.private = ? " +
"    {projectsCondition} " +

"  ) project_authorization";

List<Dto> selectAll(DbClient dbClient, DbSession session) {
  return doSelectByProjects(dbClient, session, Collections.emptyList());
}

List<Dto> selectByUuids(DbClient dbClient, DbSession session, Collection<String> projectOrViewUuids) {
  return executeLargeInputs(projectOrViewUuids, subProjectOrViewUuids -> doSelectByProjects(dbClient,
session, subProjectOrViewUuids));
}

private static List<Dto> doSelectByProjects(DbClient dbClient, DbSession session, List<String> projectUuids) {
  try {
    Map<String, Dto> dtosByProjectUuid = new HashMap<>();
    try (PreparedStatement stmt = createStatement(dbClient, session, projectUuids);
      ResultSet rs = stmt.executeQuery()) {
      while (rs.next()) {
        processRow(rs, dtosByProjectUuid);
      }
      return ImmutableList.copyOf(dtosByProjectUuid.values());
    }
  } catch (SQLException e) {
    throw new IllegalStateException("Fail to select authorizations", e);
  }
}

private static PreparedStatement createStatement(DbClient dbClient, DbSession session, List<String>
projectUuids) throws SQLException {
  String sql;
  if (projectUuids.isEmpty()) {
    sql = StringUtils.replace(SQL_TEMPLATE, "{projectsCondition}", "");
  } else {
    sql = StringUtils.replace(SQL_TEMPLATE, "{projectsCondition}", " AND projects.uuid in (" + repeat("?", ", ",
projectUuids.size()) + ")");
  }
  PreparedStatement stmt = dbClient.getMyBatis().newScrollingSelectStatement(session, sql);
  int index = 1;
  // query for RowKind.USER
  index = populateProjectUuidPlaceholders(stmt, projectUuids, index);
  // query for RowKind.GROUP
  index = populateProjectUuidPlaceholders(stmt, projectUuids, index);
  // query for RowKind.ANYONE
  index = setPrivateProjectPlaceHolder(stmt, index, false);
  index = populateProjectUuidPlaceholders(stmt, projectUuids, index);
  // query for RowKind.NONE
```

```java
    index = setPrivateProjectPlaceHolder(stmt, index, true);
    populateProjectUuidPlaceholders(stmt, projectUuids, index);
    return stmt;
  }

  private static int populateProjectUuidPlaceholders(PreparedStatement stmt, List<String> projectUuids, int index)
throws SQLException {
    int newIndex = index;
    for (String projectUuid : projectUuids) {
      stmt.setString(newIndex, projectUuid);
      newIndex++;
    }
    return newIndex;
  }

  private static int setPrivateProjectPlaceHolder(PreparedStatement stmt, int index, boolean isPrivate) throws
SQLException {
    int newIndex = index;
    stmt.setBoolean(newIndex, isPrivate);
    newIndex++;
    return newIndex;
  }

  private static void processRow(ResultSet rs, Map<String, Dto> dtosByProjectUuid) throws SQLException {
    RowKind rowKind = RowKind.valueOf(rs.getString(1));
    String projectUuid = rs.getString(2);

    Dto dto = dtosByProjectUuid.get(projectUuid);
    if (dto == null) {
      String qualifier = rs.getString(5);
      dto = new Dto(projectUuid, qualifier);
      dtosByProjectUuid.put(projectUuid, dto);
    }
    switch (rowKind) {
      case NONE:
        break;
      case USER:
        dto.addUserId(rs.getInt(3));
        break;
      case GROUP:
        dto.addGroupId(rs.getInt(4));
        break;
      case ANYONE:
        dto.allowAnyone();
        break;
    }
  }
}
```

```java
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;

import com.google.common.collect.ImmutableMap;
import java.util.Optional;
import org.elasticsearch.index.query.BoolQueryBuilder;
import org.elasticsearch.index.query.QueryBuilder;
import org.elasticsearch.index.query.QueryBuilders;
import org.elasticsearch.join.query.JoinQueryBuilders;
import org.sonar.api.ce.ComputeEngineSide;
import org.sonar.api.server.ServerSide;
import org.sonar.db.user.GroupDto;
import org.sonar.server.es.IndexType;
import org.sonar.server.es.NewIndex;
import org.sonar.server.user.UserSession;

import static com.google.common.base.Preconditions.checkArgument;
import static java.util.Objects.requireNonNull;
import static org.elasticsearch.index.query.QueryBuilders.boolQuery;
import static org.elasticsearch.index.query.QueryBuilders.termQuery;

@ServerSide
@ComputeEngineSide
public class AuthorizationTypeSupport {

  public static final String TYPE_AUTHORIZATION = "authorization";
  public static final String FIELD_GROUP_IDS = "groupIds";
  public static final String FIELD_USER_IDS = "userIds";

  /**
```

```
 * When true, then anybody can access to the project. In that case
 * it's useless to store granted groups and users. The related
 * fields are empty.
 */
public static final String FIELD_ALLOW_ANYONE = "allowAnyone";

private final UserSession userSession;

public AuthorizationTypeSupport(UserSession userSession) {
  this.userSession = userSession;
}

/**
 * @return the identifier of the ElasticSearch type (including it's index name), that corresponds to a certain
document type
 */
public static IndexType getAuthorizationIndexType(IndexType indexType) {
  requireNonNull(indexType);
  requireNonNull(indexType.getIndex());
  checkArgument(!AuthorizationTypeSupport.TYPE_AUTHORIZATION.equals(indexType.getType()),
"Authorization types do not have authorization on their own.");
  return new IndexType(indexType.getIndex(), AuthorizationTypeSupport.TYPE_AUTHORIZATION);
}

/**
 * Creates a type that requires to verify that user has the read permission
 * when searching for documents.
 * It relies on a parent type named "authorization" that is automatically
 * populated by {@link org.sonar.server.permission.index.PermissionIndexer}.
 *
 * Both types {@code typeName} and "authorization" are created. Documents
 * must be created with _parent and _routing having the parent uuid as values.
 *
 * @see NewIndex.NewIndexType#requireProjectAuthorization()
 */
public static NewIndex.NewIndexType enableProjectAuthorization(NewIndex.NewIndexType type) {
  type.setAttribute("_parent", ImmutableMap.of("type", TYPE_AUTHORIZATION));
  type.setAttribute("_routing", ImmutableMap.of("required", true));

  NewIndex.NewIndexType authType = type.getIndex().createType(TYPE_AUTHORIZATION);
  authType.setAttribute("_routing", ImmutableMap.of("required", true));
  authType.createLongField(FIELD_GROUP_IDS);
  authType.createLongField(FIELD_USER_IDS);
  authType.createBooleanField(FIELD_ALLOW_ANYONE);
  authType.setEnableSource(false);
  return type;
}
```

```java
/**
 * Build a filter to restrict query to the documents on which
 * user has read access.
 */
public QueryBuilder createQueryFilter() {
  if (userSession.isRoot()) {
    return QueryBuilders.matchAllQuery();
  }

  Integer userId = userSession.getUserId();
  BoolQueryBuilder filter = boolQuery();

  // anyone
  filter.should(QueryBuilders.termQuery(FIELD_ALLOW_ANYONE, true));

  // users
  Optional.ofNullable(userId)
    .map(Integer::longValue)
    .ifPresent(id -> filter.should(termQuery(FIELD_USER_IDS, id)));

  // groups
  userSession.getGroups()
    .stream()
    .map(GroupDto::getId)
    .forEach(groupId -> filter.should(termQuery(FIELD_GROUP_IDS, groupId)));

  return JoinQueryBuilders.hasParentQuery(
    TYPE_AUTHORIZATION,
    QueryBuilders.boolQuery().filter(filter),
    false);
}
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
```

```
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;

import com.google.common.annotations.VisibleForTesting;
import java.util.Arrays;
import java.util.Collection;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Set;
import java.util.stream.Collectors;
import java.util.stream.Stream;
import org.elasticsearch.action.index.IndexRequest;
import org.sonar.core.util.stream.MoreCollectors;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.es.EsQueueDto;
import org.sonar.server.es.BulkIndexer;
import org.sonar.server.es.BulkIndexer.Size;
import org.sonar.server.es.EsClient;
import org.sonar.server.es.IndexType;
import org.sonar.server.es.IndexingResult;
import org.sonar.server.es.OneToOneResilientIndexingListener;
import org.sonar.server.es.ProjectIndexer;
import org.sonar.server.permission.index.PermissionIndexerDao.Dto;

import static java.util.Collections.emptyList;
import static org.sonar.core.util.stream.MoreCollectors.toArrayList;
import static org.sonar.core.util.stream.MoreCollectors.toSet;

/**
 * Populates the types "authorization" of each index requiring project
 * authorization.
 */
public class PermissionIndexer implements ProjectIndexer {

  private final DbClient dbClient;
  private final EsClient esClient;
  private final Collection<AuthorizationScope> authorizationScopes;
  private final Set<IndexType> indexTypes;

  public PermissionIndexer(DbClient dbClient, EsClient esClient, NeedAuthorizationIndexer...
needAuthorizationIndexers) {
    this(dbClient, esClient, Arrays.stream(needAuthorizationIndexers)
      .map(NeedAuthorizationIndexer::getAuthorizationScope)
      .collect(MoreCollectors.toList(needAuthorizationIndexers.length)));
```

```java
  }

  @VisibleForTesting
  public PermissionIndexer(DbClient dbClient, EsClient esClient, Collection<AuthorizationScope>
authorizationScopes) {
    this.dbClient = dbClient;
    this.esClient = esClient;
    this.authorizationScopes = authorizationScopes;
    this.indexTypes = authorizationScopes.stream()
      .map(AuthorizationScope::getIndexType)
      .collect(toSet(authorizationScopes.size()));
  }

  @Override
  public Set<IndexType> getIndexTypes() {
    return indexTypes;
  }

  @Override
  public void indexOnStartup(Set<IndexType> uninitializedIndexTypes) {
    // TODO do not load everything in memory. Db rows should be scrolled.
    List<Dto> authorizations = getAllAuthorizations();
    Stream<AuthorizationScope> scopes = getScopes(uninitializedIndexTypes);
    index(authorizations, scopes, Size.LARGE);
  }

  @VisibleForTesting
  void index(List<Dto> authorizations) {
    index(authorizations, authorizationScopes.stream(), Size.REGULAR);
  }

  @Override
  public void indexOnAnalysis(String branchUuid) {
    // nothing to do, permissions don't change during an analysis
  }

  @Override
  public Collection<EsQueueDto> prepareForRecovery(DbSession dbSession, Collection<String> projectUuids,
ProjectIndexer.Cause cause) {
    switch (cause) {
      case MEASURE_CHANGE:
      case PROJECT_KEY_UPDATE:
      case PROJECT_TAGS_UPDATE:
        // nothing to change. Measures, project key and tags are not part of this index
        return emptyList();

      case PROJECT_CREATION:
      case PROJECT_DELETION:
```

```java
      case PERMISSION_CHANGE:
        return insertIntoEsQueue(dbSession, projectUuids);

      default:
        // defensive case
        throw new IllegalStateException("Unsupported cause: " + cause);
    }
  }

  private Collection<EsQueueDto> insertIntoEsQueue(DbSession dbSession, Collection<String> projectUuids) {
    List<EsQueueDto> items = indexTypes.stream()
      .flatMap(indexType -> projectUuids.stream().map(projectUuid -> EsQueueDto.create(indexType.format(),
projectUuid, null, projectUuid)))
      .collect(toArrayList());

    dbClient.esQueueDao().insert(dbSession, items);
    return items;
  }

  private void index(Collection<PermissionIndexerDao.Dto> authorizations, Stream<AuthorizationScope> scopes,
Size bulkSize) {
    if (authorizations.isEmpty()) {
      return;
    }

    // index each authorization in each scope
    scopes.forEach(scope -> {
      IndexType indexType = scope.getIndexType();

      BulkIndexer bulkIndexer = new BulkIndexer(esClient, indexType, bulkSize);
      bulkIndexer.start();

      authorizations.stream()
        .filter(scope.getProjectPredicate())
        .map(dto -> newIndexRequest(dto, indexType))
        .forEach(bulkIndexer::add);

      bulkIndexer.stop();
    });
  }

  @Override
  public IndexingResult index(DbSession dbSession, Collection<EsQueueDto> items) {
    IndexingResult result = new IndexingResult();

    List<BulkIndexer> bulkIndexers = items.stream()
      .map(EsQueueDto::getDocType)
      .distinct()
```

```
      .map(IndexType::parse)
      .filter(indexTypes::contains)
      .map(indexType -> new BulkIndexer(esClient, indexType, Size.REGULAR, new
OneToOneResilientIndexingListener(dbClient, dbSession, items)))
      .collect(Collectors.toList());

    if (bulkIndexers.isEmpty()) {
      return result;
    }

    bulkIndexers.forEach(BulkIndexer::start);

    PermissionIndexerDao permissionIndexerDao = new PermissionIndexerDao();
    Set<String> remainingProjectUuids =
items.stream().map(EsQueueDto::getDocId).collect(MoreCollectors.toHashSet());
    permissionIndexerDao.selectByUuids(dbClient, dbSession, remainingProjectUuids).forEach(p -> {
      remainingProjectUuids.remove(p.getProjectUuid());
      bulkIndexers.forEach(bi -> bi.add(newIndexRequest(p, bi.getIndexType())));
    });

    // the remaining references on projects that don't exist in db. They must
    // be deleted from index.
    remainingProjectUuids.forEach(projectUuid -> bulkIndexers.forEach(bi -> bi.addDeletion(bi.getIndexType(),
projectUuid, projectUuid)));

    bulkIndexers.forEach(b -> result.add(b.stop()));

    return result;
  }

  private static IndexRequest newIndexRequest(PermissionIndexerDao.Dto dto, IndexType indexType) {
    Map<String, Object> doc = new HashMap<>();
    if (dto.isAllowAnyone()) {
      doc.put(AuthorizationTypeSupport.FIELD_ALLOW_ANYONE, true);
      // no need to feed users and groups
    } else {
      doc.put(AuthorizationTypeSupport.FIELD_ALLOW_ANYONE, false);
      doc.put(AuthorizationTypeSupport.FIELD_GROUP_IDS, dto.getGroupIds());
      doc.put(AuthorizationTypeSupport.FIELD_USER_IDS, dto.getUserIds());
    }
    return new IndexRequest(indexType.getIndex(), indexType.getType())
      .id(dto.getProjectUuid())
      .routing(dto.getProjectUuid())
      .source(doc);
  }

  private Stream<AuthorizationScope> getScopes(Set<IndexType> indexTypes) {
    return authorizationScopes.stream()
```

```java
      .filter(scope -> indexTypes.contains(scope.getIndexType())));
  }

  private List<Dto> getAllAuthorizations() {
    try (DbSession dbSession = dbClient.openSession(false)) {
      return new PermissionIndexerDao().selectAll(dbClient, dbSession);
    }
  }
}
```
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
*/
package org.sonar.server.permission;

import javax.annotation.concurrent.Immutable;
import org.sonar.db.component.ComponentDto;
import org.sonar.server.permission.ws.ProjectWsRef;

import static java.util.Objects.requireNonNull;

/**
* Reference to a project by its db id or uuid. The field "id" should
* be removed as soon as backend is fully based on uuids.
*
* @see ProjectWsRef
*/
@Immutable
public class ProjectId {

  private final long id;
  private final String uuid;
  private final boolean isPrivate;

```java
  public ProjectId(ComponentDto project) {
    this.id = requireNonNull(project.getId());
    this.uuid = requireNonNull(project.uuid());
    this.isPrivate = project.isPrivate();
  }

  public long getId() {
    return id;
  }

  public String getUuid() {
    return uuid;
  }

  public boolean isPrivate() {
    return isPrivate;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import java.util.List;

import static org.apache.commons.lang.StringUtils.isNotBlank;
import static org.sonar.server.ws.WsUtils.checkRequest;

public class ApplyPermissionTemplateQuery {

  private final String templateUuid;
```

```java
  private List<String> componentKeys;

  private ApplyPermissionTemplateQuery(String templateUuid, List<String> componentKeys) {
    this.templateUuid = templateUuid;
    this.componentKeys = componentKeys;
    validate();
  }

  public static ApplyPermissionTemplateQuery create(String templateUuid, List<String> componentKeys) {
    return new ApplyPermissionTemplateQuery(templateUuid, componentKeys);
  }

  public String getTemplateUuid() {
    return templateUuid;
  }

  public List<String> getComponentKeys() {
    return componentKeys;
  }

  private void validate() {
    checkRequest(isNotBlank(templateUuid), "Permission template is mandatory");
    checkRequest(componentKeys != null && !componentKeys.isEmpty(), "No project provided. Please provide at
least one project.");
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import java.util.Optional;
```

```java
import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.core.permission.ProjectPermissions;

import static java.util.Objects.requireNonNull;
import static org.sonar.server.ws.WsUtils.checkRequest;

public abstract class PermissionChange {

 public enum Operation {
   ADD, REMOVE
 }

 private final Operation operation;
 private final String organizationUuid;
 private final String permission;
 private final ProjectId projectId;

 public PermissionChange(Operation operation, String organizationUuid, String permission, @Nullable ProjectId projectId) {
   this.operation = requireNonNull(operation);
   this.organizationUuid = requireNonNull(organizationUuid);
   this.permission = requireNonNull(permission);
   this.projectId = projectId;
   if (projectId == null) {
     checkRequest(GlobalPermissions.ALL.contains(permission), "Invalid global permission '%s'. Valid values are %s", permission, GlobalPermissions.ALL);
   } else {
     checkRequest(ProjectPermissions.ALL.contains(permission), "Invalid project permission '%s'. Valid values are %s", permission, ProjectPermissions.ALL);
   }
 }

 public Operation getOperation() {
   return operation;
 }

 public String getOrganizationUuid() {
   return organizationUuid;
 }

 public String getPermission() {
   return permission;
 }

 public Optional<ProjectId> getProjectId() {
   return Optional.ofNullable(projectId);
```

```java
  }

  /**
   * Shortcut based on {@link #getProjectId()}
   */
  @CheckForNull
  public String getProjectUuid() {
    return projectId == null ? null : projectId.getUuid();
  }

  /**
   * Shortcut based on {@link #getProjectId()}
   */
  @CheckForNull
  public Long getNullableProjectId() {
    return projectId == null ? null : projectId.getId();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import java.util.Optional;
import org.sonar.api.web.UserRole;
import org.sonar.db.permission.OrganizationPermission;
import org.sonar.server.user.UserSession;

import static org.sonar.server.user.AbstractUserSession.insufficientPrivilegesException;

public class PermissionPrivilegeChecker {
  private PermissionPrivilegeChecker() {
```

```java
    // static methods only
  }

  public static void checkGlobalAdmin(UserSession userSession, String organizationUuid) {
    userSession
      .checkLoggedIn()
      .checkPermission(OrganizationPermission.ADMINISTER, organizationUuid);
  }

  /**
   * Checks that user is administrator of the specified project, or of the specified organization if project is not
   * defined.
   * @throws org.sonar.server.exceptions.ForbiddenException if user is not administrator
   */
  public static void checkProjectAdmin(UserSession userSession, String organizationUuid, Optional<ProjectId>
projectId) {
    userSession.checkLoggedIn();

    if (userSession.hasPermission(OrganizationPermission.ADMINISTER, organizationUuid)) {
      return;
    }

    if (projectId.isPresent()) {
      userSession.checkComponentUuidPermission(UserRole.ADMIN, projectId.get().getUuid());
    } else {
      throw insufficientPrivilegesException();
    }
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
```

```java
package org.sonar.server.permission;

import javax.annotation.Nullable;

import static java.util.Objects.requireNonNull;

public class UserPermissionChange extends PermissionChange {

 private final UserId userId;

 public UserPermissionChange(Operation operation, String organizationUuid, String permission, @Nullable ProjectId projectId,
   UserId userId) {
   super(operation, organizationUuid, permission, projectId);
   this.userId = requireNonNull(userId);
 }

 public UserId getUserId() {
   return userId;
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import java.text.MessageFormat;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;
import java.util.List;
import java.util.Set;
```

```java
import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
import org.apache.commons.lang.StringUtils;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.server.ServerSide;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.DefaultTemplates;
import org.sonar.db.permission.GroupPermissionDto;
import org.sonar.db.permission.OrganizationPermission;
import org.sonar.db.permission.UserPermissionDto;
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.permission.template.PermissionTemplateGroupDto;
import org.sonar.db.permission.template.PermissionTemplateUserDto;
import org.sonar.server.es.ProjectIndexer;
import org.sonar.server.es.ProjectIndexers;
import org.sonar.server.permission.ws.template.DefaultTemplatesResolver;
import org.sonar.server.permission.ws.template.DefaultTemplatesResolverImpl;
import org.sonar.server.user.UserSession;

import static com.google.common.base.Preconditions.checkArgument;
import static java.lang.String.format;
import static java.util.Arrays.asList;
import static java.util.Collections.singletonList;
import static org.sonar.api.security.DefaultGroups.isAnyone;

@ServerSide
public class PermissionTemplateService {

 private final DbClient dbClient;
 private final ProjectIndexers projectIndexers;
 private final UserSession userSession;
 private final DefaultTemplatesResolver defaultTemplatesResolver;

 public PermissionTemplateService(DbClient dbClient, ProjectIndexers projectIndexers, UserSession userSession,
   DefaultTemplatesResolver defaultTemplatesResolver) {
  this.dbClient = dbClient;
  this.projectIndexers = projectIndexers;
  this.userSession = userSession;
  this.defaultTemplatesResolver = defaultTemplatesResolver;
 }

 public boolean wouldUserHaveScanPermissionWithDefaultTemplate(DbSession dbSession,
   String organizationUuid, @Nullable Integer userId,
   String projectKey, String qualifier) {
```

```
  if (userSession.hasPermission(OrganizationPermission.SCAN, organizationUuid)) {
    return true;
  }

  ComponentDto dto = new
ComponentDto().setOrganizationUuid(organizationUuid).setDbKey(projectKey).setQualifier(qualifier);
  PermissionTemplateDto template = findTemplate(dbSession, organizationUuid, dto);
  if (template == null) {
    return false;
  }

  List<String> potentialPermissions =
dbClient.permissionTemplateDao().selectPotentialPermissionsByUserIdAndTemplateId(dbSession, userId,
template.getId());
  return potentialPermissions.contains(OrganizationPermission.SCAN.getKey());
 }

 /**
  * Apply a permission template to a set of projects. Authorization to administrate these projects
  * is not verified. The projects must exist, so the "project creator" permissions defined in the
  * template are ignored.
  */
 public void applyAndCommit(DbSession dbSession, PermissionTemplateDto template,
Collection<ComponentDto> projects) {
  if (projects.isEmpty()) {
    return;
  }

  for (ComponentDto project : projects) {
    copyPermissions(dbSession, template, project, null);
  }
  projectIndexers.commitAndIndex(dbSession, projects, ProjectIndexer.Cause.PERMISSION_CHANGE);
 }

 /**
  * Apply the default permission template to project. The project can already exist (so it has permissions) or
  * can be provisioned (so has no permissions yet).
  * @param projectCreatorUserId id of the user who creates the project, only if project is provisioned. He will
  */
 public void applyDefault(DbSession dbSession, String organizationUuid, ComponentDto component, @Nullable
Integer projectCreatorUserId) {
  PermissionTemplateDto template = findTemplate(dbSession, organizationUuid, component);
  checkArgument(template != null, "Cannot retrieve default permission template");
  copyPermissions(dbSession, template, component, projectCreatorUserId);
 }

 public boolean hasDefaultTemplateWithPermissionOnProjectCreator(DbSession dbSession, String
organizationUuid, ComponentDto component) {
```

```java
    PermissionTemplateDto template = findTemplate(dbSession, organizationUuid, component);
    return hasProjectCreatorPermission(dbSession, template);
  }

  private boolean hasProjectCreatorPermission(DbSession dbSession, @Nullable PermissionTemplateDto template)
{
    return template != null && dbClient.permissionTemplateCharacteristicDao().selectByTemplateIds(dbSession,
singletonList(template.getId())).stream()
      .anyMatch(PermissionTemplateCharacteristicDto::getWithProjectCreator);
  }

  private void copyPermissions(DbSession dbSession, PermissionTemplateDto template, ComponentDto project,
@Nullable Integer projectCreatorUserId) {
    dbClient.groupPermissionDao().deleteByRootComponentId(dbSession, project.getId());
    dbClient.userPermissionDao().deleteProjectPermissions(dbSession, project.getId());

    List<PermissionTemplateUserDto> usersPermissions =
dbClient.permissionTemplateDao().selectUserPermissionsByTemplateId(dbSession, template.getId());
    String organizationUuid = template.getOrganizationUuid();
    usersPermissions
      .stream()
      .filter(up -> permissionValidForProject(project, up.getPermission()))
      .forEach(up -> {
        UserPermissionDto dto = new UserPermissionDto(organizationUuid, up.getPermission(), up.getUserId(),
project.getId());
        dbClient.userPermissionDao().insert(dbSession, dto);
      });

    List<PermissionTemplateGroupDto> groupsPermissions =
dbClient.permissionTemplateDao().selectGroupPermissionsByTemplateId(dbSession, template.getId());
    groupsPermissions
      .stream()
      .filter(gp -> groupNameValidForProject(project, gp.getGroupName()))
      .filter(gp -> permissionValidForProject(project, gp.getPermission()))
      .forEach(gp -> {
        GroupPermissionDto dto = new GroupPermissionDto()
          .setOrganizationUuid(organizationUuid)
          .setGroupId(isAnyone(gp.getGroupName()) ? null : gp.getGroupId())
          .setRole(gp.getPermission())
          .setResourceId(project.getId());
        dbClient.groupPermissionDao().insert(dbSession, dto);
      });

    List<PermissionTemplateCharacteristicDto> characteristics =
dbClient.permissionTemplateCharacteristicDao().selectByTemplateIds(dbSession, asList(template.getId()));
    if (projectCreatorUserId != null) {
      Set<String> permissionsForCurrentUserAlreadyInDb = usersPermissions.stream()
        .filter(userPermission -> projectCreatorUserId.equals(userPermission.getUserId()))
```

```java
      .map(PermissionTemplateUserDto::getPermission)
      .collect(java.util.stream.Collectors.toSet());
    characteristics.stream()
      .filter(PermissionTemplateCharacteristicDto::getWithProjectCreator)
      .filter(up -> permissionValidForProject(project, up.getPermission()))
      .filter(characteristic -> !permissionsForCurrentUserAlreadyInDb.contains(characteristic.getPermission()))
      .forEach(c -> {
        UserPermissionDto dto = new UserPermissionDto(organizationUuid, c.getPermission(), projectCreatorUserId,
project.getId());
        dbClient.userPermissionDao().insert(dbSession, dto);
      });
  }
}


private static boolean permissionValidForProject(ComponentDto project, String permission) {
  return project.isPrivate() || !ProjectPermissions.PUBLIC_PERMISSIONS.contains(permission);
}


private static boolean groupNameValidForProject(ComponentDto project, String groupName) {
  return !project.isPrivate() || !isAnyone(groupName);
}


/**
 * Return the permission template for the given component. If no template key pattern match then consider default
 * template for the component qualifier.
 */
@CheckForNull
private PermissionTemplateDto findTemplate(DbSession dbSession, String organizationUuid, ComponentDto
component) {
  List<PermissionTemplateDto> allPermissionTemplates = dbClient.permissionTemplateDao().selectAll(dbSession,
organizationUuid, null);
  List<PermissionTemplateDto> matchingTemplates = new ArrayList<>();
  for (PermissionTemplateDto permissionTemplateDto : allPermissionTemplates) {
    String keyPattern = permissionTemplateDto.getKeyPattern();
    if (StringUtils.isNotBlank(keyPattern) && component.getDbKey().matches(keyPattern)) {
      matchingTemplates.add(permissionTemplateDto);
    }
  }
  checkAtMostOneMatchForComponentKey(component.getDbKey(), matchingTemplates);
  if (matchingTemplates.size() == 1) {
    return matchingTemplates.get(0);
  }

  DefaultTemplates defaultTemplates = dbClient.organizationDao().getDefaultTemplates(dbSession,
organizationUuid)
    .orElseThrow(() -> new IllegalStateException(
      format("No Default templates defined for organization with uuid '%s'", organizationUuid)));
```

```java
    String qualifier = component.qualifier();
    DefaultTemplatesResolverImpl.ResolvedDefaultTemplates resolvedDefaultTemplates =
defaultTemplatesResolver.resolve(defaultTemplates);
    switch (qualifier) {
      case Qualifiers.PROJECT:
        return dbClient.permissionTemplateDao().selectByUuid(dbSession, resolvedDefaultTemplates.getProject());
      case Qualifiers.VIEW:
      case Qualifiers.APP:
        String viewDefaultTemplateUuid = resolvedDefaultTemplates.getView().orElseThrow(
          () -> new IllegalStateException("Attempt to create a view when Governance plugin is not installed"));
        return dbClient.permissionTemplateDao().selectByUuid(dbSession, viewDefaultTemplateUuid);
      default:
        throw new IllegalArgumentException(format("Qualifier '%s' is not supported", qualifier));
    }
  }


  private static void checkAtMostOneMatchForComponentKey(String componentKey,
List<PermissionTemplateDto> matchingTemplates) {
    if (matchingTemplates.size() > 1) {
      StringBuilder templatesNames = new StringBuilder();
      for (Iterator<PermissionTemplateDto> it = matchingTemplates.iterator(); it.hasNext();) {
        templatesNames.append("\"").append(it.next().getName()).append("\"");
        if (it.hasNext()) {
          templatesNames.append(", ");
        }
      }
      throw new IllegalStateException(MessageFormat.format(
        "The \"{0}\" key matches multiple permission templates: {1}."
          + " A system administrator must update these templates so that only one of them matches the key.",
        componentKey,
        templatesNames.toString()));
    }
  }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
```

```java
package org.sonar.server.permission;

import java.util.List;
import java.util.Optional;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.UserPermissionDto;

import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.server.permission.PermissionChange.Operation.ADD;
import static org.sonar.server.permission.PermissionChange.Operation.REMOVE;
import static org.sonar.server.ws.WsUtils.checkRequest;

/**
* Adds and removes user permissions. Both global and project scopes are supported.
*/
public class UserPermissionChanger {

 private final DbClient dbClient;

 public UserPermissionChanger(DbClient dbClient) {
   this.dbClient = dbClient;
 }

 public boolean apply(DbSession dbSession, UserPermissionChange change) {
   ensureConsistencyWithVisibility(change);
   if (isImplicitlyAlreadyDone(change)) {
    return false;
   }
   switch (change.getOperation()) {
    case ADD:
      return addPermission(dbSession, change);
    case REMOVE:
      return removePermission(dbSession, change);
    default:
      throw new UnsupportedOperationException("Unsupported permission change: " + change.getOperation());
   }
 }

 private static boolean isImplicitlyAlreadyDone(UserPermissionChange change) {
   return change.getProjectId()
```

```java
        .map(projectId -> isImplicitlyAlreadyDone(projectId, change))
        .orElse(false);
    }

    private static boolean isImplicitlyAlreadyDone(ProjectId projectId, UserPermissionChange change) {
      return isAttemptToAddPublicPermissionToPublicComponent(change, projectId);
    }

    private static boolean isAttemptToAddPublicPermissionToPublicComponent(UserPermissionChange change,
    ProjectId projectId) {
      return !projectId.isPrivate()
        && change.getOperation() == ADD
        && ProjectPermissions.PUBLIC_PERMISSIONS.contains(change.getPermission());
    }

    private static void ensureConsistencyWithVisibility(UserPermissionChange change) {
      change.getProjectId()
        .ifPresent(projectId -> checkRequest(
          !isAttemptToRemovePublicPermissionFromPublicComponent(change, projectId),
          "Permission %s can't be removed from a public component", change.getPermission()));
    }

    private static boolean isAttemptToRemovePublicPermissionFromPublicComponent(UserPermissionChange
    change, ProjectId projectId) {
      return !projectId.isPrivate()
        && change.getOperation() == REMOVE
        && ProjectPermissions.PUBLIC_PERMISSIONS.contains(change.getPermission());
    }

    private boolean addPermission(DbSession dbSession, UserPermissionChange change) {
      if (loadExistingPermissions(dbSession, change).contains(change.getPermission())) {
        return false;
      }
      UserPermissionDto dto = new UserPermissionDto(change.getOrganizationUuid(), change.getPermission(),
    change.getUserId().getId(), change.getNullableProjectId());
      dbClient.userPermissionDao().insert(dbSession, dto);
      return true;
    }

    private boolean removePermission(DbSession dbSession, UserPermissionChange change) {
      if (!loadExistingPermissions(dbSession, change).contains(change.getPermission())) {
        return false;
      }
      checkOtherAdminsExist(dbSession, change);
      Optional<ProjectId> projectId = change.getProjectId();
      if (projectId.isPresent()) {
        dbClient.userPermissionDao().deleteProjectPermission(dbSession, change.getUserId().getId(),
    change.getPermission(), projectId.get().getId());
```

```
    } else {
      dbClient.userPermissionDao().deleteGlobalPermission(dbSession, change.getUserId().getId(),
change.getPermission(), change.getOrganizationUuid());
    }
    return true;
  }

  private List<String> loadExistingPermissions(DbSession dbSession, UserPermissionChange change) {
    Optional<ProjectId> projectId = change.getProjectId();
    if (projectId.isPresent()) {
      return dbClient.userPermissionDao().selectProjectPermissionsOfUser(dbSession,
        change.getUserId().getId(),
        projectId.get().getId());
    }
    return dbClient.userPermissionDao().selectGlobalPermissionsOfUser(dbSession,
      change.getUserId().getId(),
      change.getOrganizationUuid());
  }

  private void checkOtherAdminsExist(DbSession dbSession, UserPermissionChange change) {
    if (SYSTEM_ADMIN.equals(change.getPermission()) && !change.getProjectId().isPresent()) {
      int remaining =
dbClient.authorizationDao().countUsersWithGlobalPermissionExcludingUserPermission(dbSession,
        change.getOrganizationUuid(), change.getPermission(), change.getUserId().getId());
      checkRequest(remaining > 0, "Last user with permission '%s'. Permission cannot be removed.",
SYSTEM_ADMIN);
    }
  }
}
```

```java
package org.sonar.server.permission;

import javax.annotation.concurrent.Immutable;
import org.sonar.db.user.UserDto;

import static java.util.Objects.requireNonNull;

/**
 * Reference a user by his technical (db) id or functional login.
 * This is temporary class as long as services and DAOs do not
 * use only technical id.
 */
@Immutable
public class UserId {

  private final int id;
  private final String login;

  public UserId(int userId, String login) {
    this.id = userId;
    this.login = requireNonNull(login);
  }

  public int getId() {
    return id;
  }

  public String getLogin() {
    return login;
  }

  public static UserId from(UserDto dto) {
    return new UserId(dto.getId(), dto.getLogin());
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
```

```
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import java.util.List;
import java.util.Optional;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.permission.GroupPermissionDto;

import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.server.permission.PermissionChange.Operation.ADD;
import static org.sonar.server.permission.PermissionChange.Operation.REMOVE;
import static org.sonar.server.permission.ws.PermissionRequestValidator.validateNotAnyoneAndAdminPermission;
import static org.sonar.server.ws.WsUtils.checkRequest;

public class GroupPermissionChanger {

  private final DbClient dbClient;

  public GroupPermissionChanger(DbClient dbClient) {
    this.dbClient = dbClient;
  }

  public boolean apply(DbSession dbSession, GroupPermissionChange change) {
    ensureConsistencyWithVisibility(change);
    if (isImplicitlyAlreadyDone(change)) {
      return false;
    }
    switch (change.getOperation()) {
      case ADD:
        return addPermission(dbSession, change);
      case REMOVE:
        return removePermission(dbSession, change);
      default:
        throw new UnsupportedOperationException("Unsupported permission change: " + change.getOperation());
    }
  }

  private static boolean isImplicitlyAlreadyDone(GroupPermissionChange change) {
    return change.getProjectId()
      .map(projectId -> isImplicitlyAlreadyDone(projectId, change))
      .orElse(false);
```

```java
    }

    private static boolean isImplicitlyAlreadyDone(ProjectId projectId, GroupPermissionChange change) {
      return isAttemptToAddPublicPermissionToPublicComponent(change, projectId)
        || isAttemptToRemovePermissionFromAnyoneOnPrivateComponent(change, projectId);
    }

    private static boolean isAttemptToAddPublicPermissionToPublicComponent(GroupPermissionChange change,
    ProjectId projectId) {
      return !projectId.isPrivate()
        && change.getOperation() == ADD
        && ProjectPermissions.PUBLIC_PERMISSIONS.contains(change.getPermission());
    }

    private static boolean isAttemptToRemovePermissionFromAnyoneOnPrivateComponent(GroupPermissionChange
    change, ProjectId projectId) {
      return projectId.isPrivate()
        && change.getOperation() == REMOVE
        && change.getGroupIdOrAnyone().isAnyone();
    }

    private static void ensureConsistencyWithVisibility(GroupPermissionChange change) {
      change.getProjectId()
        .ifPresent(projectId -> {
         checkRequest(
           !isAttemptToAddPermissionToAnyoneOnPrivateComponent(change, projectId),
           "No permission can be granted to Anyone on a private component");
         checkRequest(
           !isAttemptToRemovePublicPermissionFromPublicComponent(change, projectId),
           "Permission %s can't be removed from a public component", change.getPermission());
        });
    }

    private static boolean isAttemptToAddPermissionToAnyoneOnPrivateComponent(GroupPermissionChange
    change, ProjectId projectId) {
      return projectId.isPrivate()
        && change.getOperation() == ADD
        && change.getGroupIdOrAnyone().isAnyone();
    }

    private static boolean isAttemptToRemovePublicPermissionFromPublicComponent(GroupPermissionChange
    change, ProjectId projectId) {
      return !projectId.isPrivate()
        && change.getOperation() == REMOVE
        && ProjectPermissions.PUBLIC_PERMISSIONS.contains(change.getPermission());
    }

    private boolean addPermission(DbSession dbSession, GroupPermissionChange change) {
```

```
    if (loadExistingPermissions(dbSession, change).contains(change.getPermission())) {
      return false;
    }

    validateNotAnyoneAndAdminPermission(change.getPermission(), change.getGroupIdOrAnyone());
    GroupPermissionDto addedDto = new GroupPermissionDto()
      .setRole(change.getPermission())
      .setOrganizationUuid(change.getOrganizationUuid())
      .setGroupId(change.getGroupIdOrAnyone().getId())
      .setResourceId(change.getNullableProjectId());
    dbClient.groupPermissionDao().insert(dbSession, addedDto);
    return true;
  }

  private boolean removePermission(DbSession dbSession, GroupPermissionChange change) {
    if (!loadExistingPermissions(dbSession, change).contains(change.getPermission())) {
      return false;
    }
    checkIfRemainingGlobalAdministrators(dbSession, change);
    dbClient.groupPermissionDao().delete(dbSession,
      change.getPermission(),
      change.getOrganizationUuid(),
      change.getGroupIdOrAnyone().getId(),
      change.getNullableProjectId());
    return true;
  }

  private List<String> loadExistingPermissions(DbSession dbSession, GroupPermissionChange change) {
    Optional<ProjectId> projectId = change.getProjectId();
    if (projectId.isPresent()) {
      return dbClient.groupPermissionDao().selectProjectPermissionsOfGroup(dbSession,
        change.getOrganizationUuid(),
        change.getGroupIdOrAnyone().getId(),
        projectId.get().getId());
    }
    return dbClient.groupPermissionDao().selectGlobalPermissionsOfGroup(dbSession,
      change.getOrganizationUuid(),
      change.getGroupIdOrAnyone().getId());
  }

  private void checkIfRemainingGlobalAdministrators(DbSession dbSession, GroupPermissionChange change) {
    if (SYSTEM_ADMIN.equals(change.getPermission()) &&
      !change.getGroupIdOrAnyone().isAnyone() &&
      !change.getProjectId().isPresent()) {
      // removing global admin permission from group
      int remaining = dbClient.authorizationDao().countUsersWithGlobalPermissionExcludingGroup(dbSession,
        change.getOrganizationUuid(), SYSTEM_ADMIN, change.getGroupIdOrAnyone().getId());
      checkRequest(remaining > 0, "Last group with permission '%s'. Permission cannot be removed.",
```

SYSTEM_ADMIN);
    }
  }

}
{
 "paging": {
  "pageIndex": 1,
  "pageSize": 20,
  "total": 3
 },
 "groups": [
  {
   "name": "Anyone",
   "permissions": [
    "scan"
   ]
  },
  {
   "name": "group-1-name",
   "description": "group-1-description",
   "permissions": [
    "scan"
   ]
  },
  {
   "name": "group-2-name",
   "description": "group-2-description",
   "permissions": [
    "scan"
   ]
  }
 ]
}
{
 "projects": [],
 "permissions": [
  {
   "key": "user",
   "name": "Browse",
   "description": "Ability to access a project, browse its measures, and create/edit issues for it."
  },
  {
   "key": "admin",
   "name": "Administer",
   "description": "Ability to access project settings and perform administration tasks. (Users will also need
\"Browse\" permission)"
  },

```
      {
        "key": "issueadmin",
        "name": "Administer Issues",
        "description": "Grants the permission to perform advanced editing on issues: marking an issue False Positive /
Won't Fix or changing an Issue's severity. (Users will also need \"Browse\" permission)"
      },
      {
        "key": "codeviewer",
        "name": "See Source Code",
        "description": "Ability to view the project's source code. (Users will also need \"Browse\" permission)"
      }
    ],
    "paging": {
      "pageIndex": 1,
      "pageSize": 25,
      "total": 0
    }
  }
}
{
  "projects": [],
  "paging": {
    "pageIndex": 1,
    "pageSize": 25,
    "total": 0
  }
}
{
  "paging": {
    "pageIndex": 1,
    "pageSize": 20,
    "total": 2
  },
  "users": [
    {
      "login": "login-1",
      "name": "name-1",
      "email": "email-1",
      "permissions": [
        "scan"
      ]
    },
    {
      "login": "login-2",
      "name": "name-2",
      "email": "email-2",
      "permissions": [
        "scan"
      ]
```

```json
    }
  ]
}
{
 "paging": {
  "pageIndex": 1,
  "pageSize": 20,
  "total": 2
 },
 "users": [
   {
    "login": "admin",
    "name": "Administrator",
    "email": "admin@admin.com",
    "permissions": [
      "admin",
      "gateadmin",
      "profileadmin"
    ]
   },
   {
    "login": "george.orwell",
    "name": "George Orwell",
    "email": "george.orwell@1984.net",
    "permissions": [
      "scan"
    ]
   }
 ]
}
[
 {
  "key": "secret.secured",
  "value": "password"
 },
 {
  "key": "plugin.licenseHash.secured",
  "value": "987654321"
 },
 {
  "key": "foo",
  "value": "one"
 },
 {
  "key": "plugin.license.secured",
  "value": "ABCD"
 }
]
```

```
[
 {
  "key": "plugin.licenseHash.secured",
  "value": "987654321"
 },
 {
  "key": "foo",
  "value": "one"
 },
 {
  "key": "plugin.license.secured",
  "value": "ABCD"
 },
 {
  "key": "commercial.plugin",
  "value": "ABCD"
 }
]
[
 {
  "key": "foo",
  "value": "1",
  "values": [
   "1"
  ]
 },
 {
  "key": "foo.1.key",
  "value": "key1"
 },
 {
  "key": "foo.1.plugin.license.secured",
  "value": "ABCD"
 },
 {
  "key": "foo.1.secret.secured",
  "value": "123456"
 }
]
[
 {
  "key": "foo",
  "value": "1",
  "values": [
   "1"
  ]
 },
 {
```

```
    "key": "foo.1.key",
    "value": "key1"
  }
]
[
 {
   "key": "foo",
   "value": "one"
 }
]
```

```
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import java.util.List;
import javax.annotation.Nullable;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.utils.internal.AlwaysIncreasingSystem2;
import org.sonar.api.web.UserRole;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateDbTester;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.GroupDto;
```

```java
import org.sonar.db.user.UserDto;
import org.sonar.server.es.ProjectIndexers;
import org.sonar.server.es.TestProjectIndexers;
import org.sonar.server.permission.ws.template.DefaultTemplatesResolverRule;
import org.sonar.server.tester.UserSessionRule;

import static java.util.Collections.singletonList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.core.permission.GlobalPermissions.SCAN_EXECUTION;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;

public class PermissionTemplateServiceTest {

 @Rule
 public ExpectedException throwable = ExpectedException.none();
 @Rule
 public DbTester dbTester = DbTester.create(new AlwaysIncreasingSystem2());
 @Rule
 public DefaultTemplatesResolverRule defaultTemplatesResolver =
DefaultTemplatesResolverRule.withGovernance();

 private UserSessionRule userSession = UserSessionRule.standalone();
 private PermissionTemplateDbTester templateDb = dbTester.permissionTemplates();
 private DbSession session = dbTester.getSession();
 private ProjectIndexers projectIndexers = new TestProjectIndexers();

 private PermissionTemplateService underTest = new PermissionTemplateService(dbTester.getDbClient(),
projectIndexers, userSession, defaultTemplatesResolver);

 @Test
 public void apply_does_not_insert_permission_to_group_AnyOne_when_applying_template_on_private_project()
{
   OrganizationDto organization = dbTester.organizations().insert();
   ComponentDto privateProject = dbTester.components().insertPrivateProject(organization);
   PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
   dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, "p1");

   underTest.applyAndCommit(session, permissionTemplate, singletonList(privateProject));

   assertThat(selectProjectPermissionsOfGroup(organization, null, privateProject)).isEmpty();
 }

 @Test
 public void
apply_default_does_not_insert_permission_to_group_AnyOne_when_applying_template_on_private_project() {
   OrganizationDto organization = dbTester.organizations().insert();
   ComponentDto privateProject = dbTester.components().insertPrivateProject(organization);
```

```java
    UserDto creator = dbTester.users().insertUser();
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, "p1");
    dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), privateProject, creator.getId());

    assertThat(selectProjectPermissionsOfGroup(organization, null, privateProject)).isEmpty();
  }

  @Test
  public void
apply_inserts_permissions_to_group_AnyOne_but_USER_and_CODEVIEWER_when_applying_template_on_public_project() {
    OrganizationDto organization = dbTester.organizations().insert();
    ComponentDto publicProject = dbTester.components().insertPublicProject(organization);
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, perm));
    dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, "p1");

    underTest.applyAndCommit(session, permissionTemplate, singletonList(publicProject));

    assertThat(selectProjectPermissionsOfGroup(organization, null, publicProject))
      .containsOnly("p1", UserRole.ADMIN, UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void
applyDefault_inserts_permissions_to_group_AnyOne_but_USER_and_CODEVIEWER_when_applying_template_on_public_project() {
    OrganizationDto organization = dbTester.organizations().insert();
    ComponentDto publicProject = dbTester.components().insertPublicProject(organization);
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, perm));
    dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, "p1");
    dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), publicProject, null);

    assertThat(selectProjectPermissionsOfGroup(organization, null, publicProject))
      .containsOnly("p1", UserRole.ADMIN, UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void apply_inserts_any_permissions_to_group_when_applying_template_on_private_project() {
    OrganizationDto organization = dbTester.organizations().insert();
```

```java
    ComponentDto privateProject = dbTester.components().insertPrivateProject(organization);
    GroupDto group = dbTester.users().insertGroup(organization);
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, perm));
    dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, "p1");

    underTest.applyAndCommit(session, permissionTemplate, singletonList(privateProject));

    assertThat(selectProjectPermissionsOfGroup(organization, group, privateProject))
      .containsOnly("p1", UserRole.USER, UserRole.CODEVIEWER, UserRole.ADMIN, UserRole.ISSUE_ADMIN,
GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void applyDefault_inserts_any_permissions_to_group_when_applying_template_on_private_project() {
    OrganizationDto organization = dbTester.organizations().insert();
    GroupDto group = dbTester.users().insertGroup(organization);
    ComponentDto privateProject = dbTester.components().insertPrivateProject(organization);
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, perm));
    dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, "p1");
    dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), privateProject, null);

    assertThat(selectProjectPermissionsOfGroup(organization, group, privateProject))
      .containsOnly("p1", UserRole.USER, UserRole.CODEVIEWER, UserRole.ADMIN, UserRole.ISSUE_ADMIN,
GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void
apply_inserts_permissions_to_group_but_USER_and_CODEVIEWER_when_applying_template_on_public_projec
t() {
    OrganizationDto organization = dbTester.organizations().insert();
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ComponentDto publicProject = dbTester.components().insertPublicProject(organization);
    GroupDto group = dbTester.users().insertGroup(organization);
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, perm));
    dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, "p1");

    underTest.applyAndCommit(session, permissionTemplate, singletonList(publicProject));

    assertThat(selectProjectPermissionsOfGroup(organization, group, publicProject))
      .containsOnly("p1", UserRole.ADMIN, UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION);
```

```java
  }

  @Test
  public void
applyDefault_inserts_permissions_to_group_but_USER_and_CODEVIEWER_when_applying_template_on_public
_project() {
    OrganizationDto organization = dbTester.organizations().insert();
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ComponentDto publicProject = dbTester.components().insertPublicProject(organization);
    GroupDto group = dbTester.users().insertGroup(organization);
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, perm));
    dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, "p1");
    dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), publicProject, null);

    assertThat(selectProjectPermissionsOfGroup(organization, group, publicProject))
      .containsOnly("p1", UserRole.ADMIN, UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void
apply_inserts_permissions_to_user_but_USER_and_CODEVIEWER_when_applying_template_on_public_project(
) {
    OrganizationDto organization = dbTester.organizations().insert();
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ComponentDto publicProject = dbTester.components().insertPublicProject(organization);
    UserDto user = dbTester.users().insertUser();
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, perm));
    dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, "p1");

    underTest.applyAndCommit(session, permissionTemplate, singletonList(publicProject));

    assertThat(selectProjectPermissionsOfUser(user, publicProject))
      .containsOnly("p1", UserRole.ADMIN, UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void
applyDefault_inserts_permissions_to_user_but_USER_and_CODEVIEWER_when_applying_template_on_public_
project() {
    OrganizationDto organization = dbTester.organizations().insert();
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ComponentDto publicProject = dbTester.components().insertPublicProject(organization);
    UserDto user = dbTester.users().insertUser();
    ProjectPermissions.ALL
```

```java
      .forEach(perm -> dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, perm));
    dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, "p1");
    dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), publicProject, null);

    assertThat(selectProjectPermissionsOfUser(user, publicProject))
      .containsOnly("p1", UserRole.ADMIN, UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void apply_inserts_any_permissions_to_user_when_applying_template_on_private_project() {
    OrganizationDto organization = dbTester.organizations().insert();
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ComponentDto privateProject = dbTester.components().insertPrivateProject(organization);
    UserDto user = dbTester.users().insertUser();
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, perm));
    dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, "p1");

    underTest.applyAndCommit(session, permissionTemplate, singletonList(privateProject));

    assertThat(selectProjectPermissionsOfUser(user, privateProject))
      .containsOnly("p1", UserRole.USER, UserRole.CODEVIEWER, UserRole.ADMIN, UserRole.ISSUE_ADMIN,
GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void applyDefault_inserts_any_permissions_to_user_when_applying_template_on_private_project() {
    OrganizationDto organization = dbTester.organizations().insert();
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
    ComponentDto privateProject = dbTester.components().insertPrivateProject(organization);
    UserDto user = dbTester.users().insertUser();
    ProjectPermissions.ALL
      .forEach(perm -> dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, perm));
    dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, "p1");
    dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), privateProject, null);

    assertThat(selectProjectPermissionsOfUser(user, privateProject))
      .containsOnly("p1", UserRole.USER, UserRole.CODEVIEWER, UserRole.ADMIN, UserRole.ISSUE_ADMIN,
GlobalPermissions.SCAN_EXECUTION);
  }

  @Test
  public void
applyDefault_inserts_permissions_to_ProjectCreator_but_USER_and_CODEVIEWER_when_applying_template_o
```

```java
n_public_project() {
  OrganizationDto organization = dbTester.organizations().insert();
  PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
  ComponentDto publicProject = dbTester.components().insertPublicProject(organization);
  UserDto user = dbTester.users().insertUser();
  ProjectPermissions.ALL
    .forEach(perm -> dbTester.permissionTemplates().addProjectCreatorToTemplate(permissionTemplate, perm));
  dbTester.permissionTemplates().addProjectCreatorToTemplate(permissionTemplate, "p1");
  dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

  underTest.applyDefault(session, organization.getUuid(), publicProject, user.getId());

  assertThat(selectProjectPermissionsOfUser(user, publicProject))
    .containsOnly("p1", UserRole.ADMIN, UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION);
}

@Test
public void
applyDefault_inserts_any_permissions_to_ProjectCreator_when_applying_template_on_private_project() {
  OrganizationDto organization = dbTester.organizations().insert();
  PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
  ComponentDto privateProject = dbTester.components().insertPrivateProject(organization);
  UserDto user = dbTester.users().insertUser();
  ProjectPermissions.ALL
    .forEach(perm -> dbTester.permissionTemplates().addProjectCreatorToTemplate(permissionTemplate, perm));
  dbTester.permissionTemplates().addProjectCreatorToTemplate(permissionTemplate, "p1");
  dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

  underTest.applyDefault(session, organization.getUuid(), privateProject, user.getId());

  assertThat(selectProjectPermissionsOfUser(user, privateProject))
    .containsOnly("p1", UserRole.USER, UserRole.CODEVIEWER, UserRole.ADMIN, UserRole.ISSUE_ADMIN,
GlobalPermissions.SCAN_EXECUTION);
}

@Test
public void apply_template_on_view() {
  OrganizationDto organization = dbTester.organizations().insert();
  ComponentDto view = dbTester.components().insertView(organization);
  PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
  GroupDto group = dbTester.users().insertGroup(organization);
  dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, ADMINISTER.getKey());
  dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group,
PROVISION_PROJECTS.getKey());
  dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

  underTest.applyDefault(session, organization.getUuid(), view, null);
```

```java
    assertThat(selectProjectPermissionsOfGroup(organization, group, view))
      .containsOnly(ADMINISTER.getKey(), PROVISION_PROJECTS.getKey());
  }

  @Test
  public void apply_default_template_on_view() {
    OrganizationDto organization = dbTester.organizations().insert();
    ComponentDto view = dbTester.components().insertView(organization);
    PermissionTemplateDto projectPermissionTemplate =
dbTester.permissionTemplates().insertTemplate(organization);
    PermissionTemplateDto viewPermissionTemplate =
dbTester.permissionTemplates().insertTemplate(organization);
    GroupDto group = dbTester.users().insertGroup(organization);
    dbTester.permissionTemplates().addGroupToTemplate(viewPermissionTemplate, group,
ADMINISTER.getKey());
    dbTester.permissionTemplates().addGroupToTemplate(viewPermissionTemplate, group,
PROVISION_PROJECTS.getKey());
    dbTester.organizations().setDefaultTemplates(organization, projectPermissionTemplate.getUuid(),
viewPermissionTemplate.getUuid());

    underTest.applyDefault(session, organization.getUuid(), view, null);

    assertThat(selectProjectPermissionsOfGroup(organization, group, view))
      .containsOnly(ADMINISTER.getKey(), PROVISION_PROJECTS.getKey());
  }

  @Test
  public void apply_project_default_template_on_view_when_no_view_default_template() {
    OrganizationDto organization = dbTester.organizations().insert();
    ComponentDto view = dbTester.components().insertView(organization);
    PermissionTemplateDto projectPermissionTemplate =
dbTester.permissionTemplates().insertTemplate(organization);
    GroupDto group = dbTester.users().insertGroup(organization);
    dbTester.permissionTemplates().addGroupToTemplate(projectPermissionTemplate, group,
PROVISION_PROJECTS.getKey());
    dbTester.organizations().setDefaultTemplates(organization, projectPermissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), view, null);

    assertThat(selectProjectPermissionsOfGroup(organization, group,
view)).containsOnly(PROVISION_PROJECTS.getKey());
  }

  @Test
  public void apply_template_on_applications() {
    OrganizationDto organization = dbTester.organizations().insert();
    ComponentDto application = dbTester.components().insertApplication(organization);
    PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
```

```java
    GroupDto group = dbTester.users().insertGroup(organization);
    dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group, ADMINISTER.getKey());
    dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, group,
PROVISION_PROJECTS.getKey());
    dbTester.organizations().setDefaultTemplates(organization, permissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), application, null);

    assertThat(selectProjectPermissionsOfGroup(organization, group, application))
      .containsOnly(ADMINISTER.getKey(), PROVISION_PROJECTS.getKey());
  }

  @Test
  public void apply_default_view_template_on_application() {
    OrganizationDto organization = dbTester.organizations().insert();
    ComponentDto application = dbTester.components().insertApplication(organization);
    PermissionTemplateDto projectPermissionTemplate =
dbTester.permissionTemplates().insertTemplate(organization);
    PermissionTemplateDto viewPermissionTemplate =
dbTester.permissionTemplates().insertTemplate(organization);
    GroupDto group = dbTester.users().insertGroup(organization);
    dbTester.permissionTemplates().addGroupToTemplate(viewPermissionTemplate, group,
ADMINISTER.getKey());
    dbTester.permissionTemplates().addGroupToTemplate(viewPermissionTemplate, group,
PROVISION_PROJECTS.getKey());
    dbTester.organizations().setDefaultTemplates(organization, projectPermissionTemplate.getUuid(),
viewPermissionTemplate.getUuid());

    underTest.applyDefault(session, organization.getUuid(), application, null);

    assertThat(selectProjectPermissionsOfGroup(organization, group, application))
      .containsOnly(ADMINISTER.getKey(), PROVISION_PROJECTS.getKey());
  }

  @Test
  public void apply_project_default_template_on_application_when_no_application_default_template() {
    OrganizationDto organization = dbTester.organizations().insert();
    ComponentDto application = dbTester.components().insertApplication(organization);
    PermissionTemplateDto projectPermissionTemplate =
dbTester.permissionTemplates().insertTemplate(organization);
    GroupDto group = dbTester.users().insertGroup(organization);
    dbTester.permissionTemplates().addGroupToTemplate(projectPermissionTemplate, group,
PROVISION_PROJECTS.getKey());
    dbTester.organizations().setDefaultTemplates(organization, projectPermissionTemplate.getUuid(), null);

    underTest.applyDefault(session, organization.getUuid(), application, null);

    assertThat(selectProjectPermissionsOfGroup(organization, group,
```

```java
  application)).containsOnly(PROVISION_PROJECTS.getKey());
 }

 @Test
 public void apply_permission_template() {
   OrganizationDto organization = dbTester.organizations().insert();
   UserDto user = dbTester.users().insertUser();
   ComponentDto project = dbTester.components().insertPrivateProject(organization);
   GroupDto adminGroup = dbTester.users().insertGroup(organization);
   GroupDto userGroup = dbTester.users().insertGroup(organization);
   dbTester.users().insertPermissionOnGroup(adminGroup, "admin");
   dbTester.users().insertPermissionOnGroup(userGroup, "user");
   dbTester.users().insertPermissionOnUser(organization, user, "admin");
   PermissionTemplateDto permissionTemplate = dbTester.permissionTemplates().insertTemplate(organization);
   dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, adminGroup, "admin");
   dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, adminGroup, "issueadmin");
   dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, userGroup, "user");
   dbTester.permissionTemplates().addGroupToTemplate(permissionTemplate, userGroup, "codeviewer");
   dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, "user");
   dbTester.permissionTemplates().addAnyoneToTemplate(permissionTemplate, "codeviewer");
   dbTester.permissionTemplates().addUserToTemplate(permissionTemplate, user, "admin");

   assertThat(selectProjectPermissionsOfGroup(organization, adminGroup, project)).isEmpty();
   assertThat(selectProjectPermissionsOfGroup(organization, userGroup, project)).isEmpty();
   assertThat(selectProjectPermissionsOfGroup(organization, null, project)).isEmpty();
   assertThat(selectProjectPermissionsOfUser(user, project)).isEmpty();

   underTest.applyAndCommit(session, permissionTemplate, singletonList(project));

   assertThat(selectProjectPermissionsOfGroup(organization, adminGroup, project)).containsOnly("admin",
 "issueadmin");
   assertThat(selectProjectPermissionsOfGroup(organization, userGroup, project)).containsOnly("user",
 "codeviewer");
   assertThat(selectProjectPermissionsOfGroup(organization, null, project)).isEmpty();
   assertThat(selectProjectPermissionsOfUser(user, project)).containsOnly("admin");
 }

 private List<String> selectProjectPermissionsOfGroup(OrganizationDto organizationDto, @Nullable GroupDto
 groupDto, ComponentDto project) {
   return dbTester.getDbClient().groupPermissionDao().selectProjectPermissionsOfGroup(session,
     organizationDto.getUuid(), groupDto != null ? groupDto.getId() : null, project.getId());
 }

 private List<String> selectProjectPermissionsOfUser(UserDto userDto, ComponentDto project) {
   return dbTester.getDbClient().userPermissionDao().selectProjectPermissionsOfUser(session,
     userDto.getId(), project.getId());
 }
```

```
@Test
public void would_user_have_scan_permission_with_default_permission_template() {
  OrganizationDto organization = dbTester.organizations().insert();
  GroupDto group = dbTester.users().insertGroup(organization);
  UserDto user = dbTester.users().insertUser();
  dbTester.users().insertMember(group, user);
  PermissionTemplateDto template = templateDb.insertTemplate(organization);
  dbTester.organizations().setDefaultTemplates(template, null);
  templateDb.addProjectCreatorToTemplate(template.getId(), SCAN_EXECUTION);
  templateDb.addUserToTemplate(template.getId(), user.getId(), UserRole.USER);
  templateDb.addGroupToTemplate(template.getId(), group.getId(), UserRole.CODEVIEWER);
  templateDb.addGroupToTemplate(template.getId(), null, UserRole.ISSUE_ADMIN);

  // authenticated user
  checkWouldUserHaveScanPermission(organization, user.getId(), true);

  // anonymous user
  checkWouldUserHaveScanPermission(organization, null, false);
}

@Test
public void would_user_have_scan_permission_with_unknown_default_permission_template() {
  dbTester.organizations().setDefaultTemplates(dbTester.getDefaultOrganization(),
"UNKNOWN_TEMPLATE_UUID", null);

  checkWouldUserHaveScanPermission(dbTester.getDefaultOrganization(), null, false);
}

@Test
public void would_user_have_scan_permission_with_empty_template() {
  PermissionTemplateDto template = templateDb.insertTemplate(dbTester.getDefaultOrganization());
  dbTester.organizations().setDefaultTemplates(template, null);

  checkWouldUserHaveScanPermission(dbTester.getDefaultOrganization(), null, false);
}

private void checkWouldUserHaveScanPermission(OrganizationDto organization, @Nullable Integer userId,
boolean expectedResult) {
  assertThat(underTest.wouldUserHaveScanPermissionWithDefaultTemplate(session, organization.getUuid(),
userId, "PROJECT_KEY", Qualifiers.PROJECT))
    .isEqualTo(expectedResult);
}

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
```

```
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Test;
import org.sonar.api.security.DefaultGroups;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;

import static java.lang.String.format;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.server.ws.WebService.Param.PAGE;
import static org.sonar.api.server.ws.WebService.Param.PAGE_SIZE;
import static org.sonar.api.server.ws.WebService.Param.TEXT_QUERY;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.core.permission.GlobalPermissions.SCAN_EXECUTION;
import static org.sonar.db.component.ComponentTesting.newPrivateProjectDto;
import static org.sonar.db.component.ComponentTesting.newView;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.SCAN;
import static org.sonar.test.JsonAssert.assertJson;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;

public class GroupsActionTest extends BasePermissionWsTest<GroupsAction> {
```

```java
private GroupDto group1;
private GroupDto group2;
private GroupDto group3;

@Override
protected GroupsAction buildWsAction() {
  return new GroupsAction(
    db.getDbClient(),
    userSession,
    newPermissionWsSupport());
}

@Before
public void setUp() {
  OrganizationDto defOrg = db.getDefaultOrganization();
  group1 = db.users().insertGroup(defOrg, "group-1-name");
  group2 = db.users().insertGroup(defOrg, "group-2-name");
  group3 = db.users().insertGroup(defOrg, "group-3-name");
  db.users().insertPermissionOnGroup(group1, SCAN);
  db.users().insertPermissionOnGroup(group2, SCAN);
  db.users().insertPermissionOnGroup(group3, ADMINISTER);
  db.users().insertPermissionOnAnyone(defOrg, SCAN);
  db.commit();
}

@Test
public void search_for_groups_with_one_permission() {
  loginAsAdmin(db.getDefaultOrganization());

  String json = newRequest()
    .setParam(PARAM_PERMISSION, SCAN.getKey())
    .execute()
    .getInput();
  assertJson(json).isSimilarTo("{\n" +
    " \"paging\": {\n" +
    "   \"pageIndex\": 1,\n" +
    "   \"pageSize\": 20,\n" +
    "   \"total\": 3\n" +
    " },\n" +
    " \"groups\": [\n" +
    "   {\n" +
    "     \"name\": \"Anyone\",\n" +
    "     \"permissions\": [\n" +
    "       \"scan\"\n" +
    "     ]\n" +
    "   },\n" +
    "   {\n" +
    "     \"name\": \"group-1-name\",\n" +
```

```java
      "    \"description\": \"" + group1.getDescription() + "\",\n" +
      "    \"permissions\": [\n" +
      "      \"scan\"\n" +
      "    ]\n" +
      "  },\n" +
      "  {\n" +
      "    \"name\": \"group-2-name\",\n" +
      "    \"description\": \"" + group2.getDescription() + "\",\n" +
      "    \"permissions\": [\n" +
      "      \"scan\"\n" +
      "    ]\n" +
      "  }\n" +
      "  ]\n" +
      "}\n");
  }

  @Test
  public void search_with_selection() {
    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
      .setParam(PARAM_PERMISSION, SCAN.getKey())
      .execute()
      .getInput();

    assertThat(result).containsSubsequence(DefaultGroups.ANYONE, "group-1", "group-2");
  }

  @Test
  public void search_groups_with_pagination() {
    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
      .setParam(PARAM_PERMISSION, SCAN.getKey())
      .setParam(PAGE_SIZE, "1")
      .setParam(PAGE, "3")
      .execute()
      .getInput();

    assertThat(result).contains("group-2")
      .doesNotContain("group-1")
      .doesNotContain("group-3");
  }

  @Test
  public void search_groups_with_query() {
    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
      .setParam(PARAM_PERMISSION, SCAN.getKey())
      .setParam(TEXT_QUERY, "group-")
```

```java
      .execute()
      .getInput();

    assertThat(result)
      .contains("group-1", "group-2")
      .doesNotContain(DefaultGroups.ANYONE);
  }

  @Test
  public void search_groups_with_project_permissions() {
    OrganizationDto organizationDto = db.getDefaultOrganization();
    ComponentDto project = db.components().insertComponent(newPrivateProjectDto(organizationDto, "project-uuid"));
    GroupDto group = db.users().insertGroup(organizationDto, "project-group-name");
    db.users().insertProjectPermissionOnGroup(group, ISSUE_ADMIN, project);

    ComponentDto anotherProject =
db.components().insertComponent(ComponentTesting.newPrivateProjectDto(organizationDto));
    GroupDto anotherGroup = db.users().insertGroup(organizationDto, "another-project-group-name");
    db.users().insertProjectPermissionOnGroup(anotherGroup, ISSUE_ADMIN, anotherProject);

    GroupDto groupWithoutPermission = db.users().insertGroup(organizationDto, "group-without-permission");

    userSession.logIn().addProjectPermission(ADMIN, project);
    String result = newRequest()
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .setParam(PARAM_PROJECT_ID, "project-uuid")
      .execute()
      .getInput();

    assertThat(result).contains(group.getName())
      .doesNotContain(anotherGroup.getName())
      .doesNotContain(groupWithoutPermission.getName());
  }

  @Test
  public void return_also_groups_without_permission_when_search_query() {
    OrganizationDto organizationDto = db.getDefaultOrganization();
    ComponentDto project = db.components().insertComponent(newPrivateProjectDto(organizationDto, "project-uuid"));
    GroupDto group = db.users().insertGroup(organizationDto, "group-with-permission");
    db.users().insertProjectPermissionOnGroup(group, ISSUE_ADMIN, project);

    GroupDto groupWithoutPermission = db.users().insertGroup(organizationDto, "group-without-permission");
    GroupDto anotherGroup = db.users().insertGroup(organizationDto, "another-group");

    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
```

```
    .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
    .setParam(PARAM_PROJECT_ID, "project-uuid")
    .setParam(TEXT_QUERY, "group-with")
    .execute()
    .getInput();

  assertThat(result).contains(group.getName())
    .doesNotContain(groupWithoutPermission.getName())
    .doesNotContain(anotherGroup.getName());
}

@Test
public void return_only_groups_with_permission_when_no_search_query() {
  ComponentDto project = db.components().insertComponent(newPrivateProjectDto(db.getDefaultOrganization(),
"project-uuid"));
  GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "project-group-name");
  db.users().insertProjectPermissionOnGroup(group, ISSUE_ADMIN, project);

  GroupDto groupWithoutPermission = db.users().insertGroup(db.getDefaultOrganization(), "group-without-
permission");

  loginAsAdmin(db.getDefaultOrganization());
  String result = newRequest()
    .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
    .setParam(PARAM_PROJECT_ID, project.uuid())
    .execute()
    .getInput();

  assertThat(result).contains(group.getName())
    .doesNotContain(groupWithoutPermission.getName());
}

@Test
public void return_anyone_group_when_search_query_and_no_param_permission() {
  OrganizationDto organizationDto = db.organizations().insert();
  ComponentDto project = db.components().insertComponent(newPrivateProjectDto(organizationDto, "project-
uuid"));
  GroupDto group = db.users().insertGroup(organizationDto, "group-with-permission");
  db.users().insertProjectPermissionOnGroup(group, ISSUE_ADMIN, project);

  loginAsAdmin(db.getDefaultOrganization());
  String result = newRequest()
    .setParam(PARAM_PROJECT_ID, project.uuid())
    .setParam(TEXT_QUERY, "nyo")
    .execute()
    .getInput();

  assertThat(result).contains("Anyone");
```

```
  }

  @Test
  public void search_groups_on_views() {
    ComponentDto view = db.components().insertComponent(newView(db.getDefaultOrganization(), "view-
uuid").setDbKey("view-key"));
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "project-group-name");
    db.users().insertProjectPermissionOnGroup(group, ISSUE_ADMIN, view);

    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .setParam(PARAM_PROJECT_ID, "view-uuid")
      .execute()
      .getInput();

    assertThat(result).contains("project-group-name")
      .doesNotContain("group-1")
      .doesNotContain("group-2")
      .doesNotContain("group-3");
  }

  @Test
  public void fail_if_not_logged_in() {
    expectedException.expect(UnauthorizedException.class);
    userSession.anonymous();

    newRequest()
      .setParam(PARAM_PERMISSION, SCAN.getKey())
      .execute();
  }

  @Test
  public void fail_if_insufficient_privileges() {
    expectedException.expect(ForbiddenException.class);

    userSession.logIn("login");
    newRequest()
      .setParam(PARAM_PERMISSION, SCAN.getKey())
      .execute();
  }

  @Test
  public void fail_if_project_uuid_and_project_key_are_provided() {
    db.components().insertComponent(newPrivateProjectDto(db.organizations().insert(), "project-
uuid").setDbKey("project-key"));

    expectedException.expect(BadRequestException.class);
```

```java
    loginAsAdmin(db.getDefaultOrganization());
    newRequest()
      .setParam(PARAM_PERMISSION, SCAN_EXECUTION)
      .setParam(PARAM_PROJECT_ID, "project-uuid")
      .setParam(PARAM_PROJECT_KEY, "project-key")
      .execute();
  }

  @Test
  public void fail_when_using_branch_uuid() {
    ComponentDto project = db.components().insertMainBranch();
    ComponentDto branch = db.components().insertProjectBranch(project);
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization());
    db.users().insertProjectPermissionOnGroup(group, ISSUE_ADMIN, project);
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage(format("Project id '%s' not found", branch.uuid()));

    newRequest()
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .setParam(PARAM_PROJECT_ID, branch.uuid())
      .execute();
  }

  @Test
  public void fail_when_using_branch_db_key() {
    ComponentDto project = db.components().insertMainBranch();
    ComponentDto branch = db.components().insertProjectBranch(project);
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization());
    db.users().insertProjectPermissionOnGroup(group, ISSUE_ADMIN, project);
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage(format("Project key '%s' not found", branch.getDbKey()));

    newRequest()
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .setParam(PARAM_PROJECT_KEY, branch.getDbKey())
      .execute();
  }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
```

```
package org.sonar.server.permission.ws;

import org.junit.Test;
import org.sonar.core.platform.ComponentContainer;

import static org.assertj.core.api.Assertions.assertThat;
import static
org.sonar.core.platform.ComponentContainer.COMPONENTS_IN_EMPTY_COMPONENT_CONTAINER;

public class PermissionsWsModuleTest {
 @Test
 public void verify_count_of_added_components() {
   ComponentContainer container = new ComponentContainer();
   new PermissionsWsModule().configure(container);
   assertThat(container.size()).isEqualTo(COMPONENTS_IN_EMPTY_COMPONENT_CONTAINER + 25);
 }
}
```

```java
package org.sonar.server.permission.ws;

import com.google.common.collect.HashBasedTable;
import java.util.Collections;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;

public class SearchProjectPermissionsDataTest {
 @Rule
 public ExpectedException expectedException = ExpectedException.none();

 @Test
 public void fail_if_no_projects() {
   expectedException.expect(IllegalStateException.class);

   SearchProjectPermissionsData.newBuilder()
     .groupCountByProjectIdAndPermission(HashBasedTable.create())
     .userCountByProjectIdAndPermission(HashBasedTable.create())
     .build();
 }

 @Test
 public void fail_if_no_group_count() {
   expectedException.expect(IllegalStateException.class);

   SearchProjectPermissionsData.newBuilder()
     .rootComponents(Collections.emptyList())
     .userCountByProjectIdAndPermission(HashBasedTable.create())
     .build();
 }

 @Test
 public void fail_if_no_user_count() {
   expectedException.expect(IllegalStateException.class);

   SearchProjectPermissionsData.newBuilder()
     .rootComponents(Collections.emptyList())
     .groupCountByProjectIdAndPermission(HashBasedTable.create())
     .build();
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
```

```
package org.sonar.server.permission.ws.template;

import java.util.List;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Test;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class RemoveUserFromTemplateActionTest extends
BasePermissionWsTest<RemoveUserFromTemplateAction> {

 private static final String DEFAULT_PERMISSION = CODEVIEWER;

 private UserDto user;
 private PermissionTemplateDto template;
```

```java
@Override
protected RemoveUserFromTemplateAction buildWsAction() {
  return new RemoveUserFromTemplateAction(db.getDbClient(), newPermissionWsSupport(), userSession);
}

@Before
public void setUp() {
  user = db.users().insertUser("user-login");
  db.organizations().addMember(db.getDefaultOrganization(), user);
  template = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
  addUserToTemplate(user, template, DEFAULT_PERMISSION);
}

@Test
public void remove_user_from_template() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());
  newRequest(user.getLogin(), template.getUuid(), DEFAULT_PERMISSION);

  assertThat(getLoginsInTemplateAndPermission(template, DEFAULT_PERMISSION)).isEmpty();
}

@Test
public void remove_user_from_template_by_name_case_insensitive() {
  loginAsAdmin(db.getDefaultOrganization());
  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, DEFAULT_PERMISSION)
    .setParam(PARAM_TEMPLATE_NAME, template.getName().toUpperCase())
    .execute();

  assertThat(getLoginsInTemplateAndPermission(template, DEFAULT_PERMISSION)).isEmpty();
}

@Test
public void remove_user_from_template_twice_without_failing() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());
  newRequest(user.getLogin(), template.getUuid(), DEFAULT_PERMISSION);
  newRequest(user.getLogin(), template.getUuid(), DEFAULT_PERMISSION);

  assertThat(getLoginsInTemplateAndPermission(template, DEFAULT_PERMISSION)).isEmpty();
}

@Test
public void keep_user_permission_not_removed() throws Exception {
  addUserToTemplate(user, template, ISSUE_ADMIN);

  loginAsAdmin(db.getDefaultOrganization());
  newRequest(user.getLogin(), template.getUuid(), DEFAULT_PERMISSION);
```

```java
    assertThat(getLoginsInTemplateAndPermission(template, DEFAULT_PERMISSION)).isEmpty();
    assertThat(getLoginsInTemplateAndPermission(template, ISSUE_ADMIN)).containsExactly(user.getLogin());
  }

  @Test
  public void keep_other_users_when_one_user_removed() throws Exception {
    UserDto newUser = db.users().insertUser("new-login");
    db.organizations().addMember(db.getDefaultOrganization(), newUser);
    addUserToTemplate(newUser, template, DEFAULT_PERMISSION);

    loginAsAdmin(db.getDefaultOrganization());
    newRequest(user.getLogin(), template.getUuid(), DEFAULT_PERMISSION);

    assertThat(getLoginsInTemplateAndPermission(template, DEFAULT_PERMISSION)).containsExactly("new-
login");
  }

  @Test
  public void fail_if_not_a_project_permission() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(user.getLogin(), template.getUuid(), GlobalPermissions.PROVISIONING);
  }

  @Test
  public void fail_if_insufficient_privileges() throws Exception {
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest(user.getLogin(), template.getUuid(), DEFAULT_PERMISSION);
  }

  @Test
  public void fail_if_not_logged_in() throws Exception {
    userSession.anonymous();

    expectedException.expect(UnauthorizedException.class);

    newRequest(user.getLogin(), template.getUuid(), DEFAULT_PERMISSION);
  }

  @Test
  public void fail_if_user_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());
```

```java
    expectedException.expect(IllegalArgumentException.class);

    newRequest(null, template.getUuid(), DEFAULT_PERMISSION);
  }

  @Test
  public void fail_if_permission_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(user.getLogin(), template.getUuid(), null);
  }

  @Test
  public void fail_if_template_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    newRequest(user.getLogin(), null, DEFAULT_PERMISSION);
  }

  @Test
  public void fail_if_user_does_not_exist() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("User with login 'unknown-login' is not found");

    newRequest("unknown-login", template.getUuid(), DEFAULT_PERMISSION);
  }

  @Test
  public void fail_if_template_key_does_not_exist() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("Permission template with id 'unknown-key' is not found");

    newRequest(user.getLogin(), "unknown-key", DEFAULT_PERMISSION);
  }

  private void newRequest(@Nullable String userLogin, @Nullable String templateKey, @Nullable String
permission) {
    TestRequest request = newRequest();
    if (userLogin != null) {
```

```
      request.setParam(PARAM_USER_LOGIN, userLogin);
    }
    if (templateKey != null) {
      request.setParam(org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID,
templateKey);
    }
    if (permission != null) {
      request.setParam(PARAM_PERMISSION, permission);
    }

    request.execute();
  }

  private List<String> getLoginsInTemplateAndPermission(PermissionTemplateDto template, String permission) {
    PermissionQuery permissionQuery =
PermissionQuery.builder().setOrganizationUuid(template.getOrganizationUuid()).setPermission(permission).build()
;
    return db.getDbClient().permissionTemplateDao()
      .selectUserLoginsByQueryAndTemplate(db.getSession(), permissionQuery, template.getId());
  }

  private void addUserToTemplate(UserDto user, PermissionTemplateDto template, String permission) {
    db.getDbClient().permissionTemplateDao().insertUserPermission(db.getSession(), template.getId(), user.getId(),
permission);
    db.commit();
  }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;
```

```java
import java.util.List;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.api.web.UserRole;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.es.TestProjectIndexers;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.permission.PermissionTemplateService;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;
import org.sonar.server.ws.TestResponse;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class ApplyTemplateActionTest extends BasePermissionWsTest<ApplyTemplateAction> {

  @Rule
  public DefaultTemplatesResolverRule defaultTemplatesResolver =
DefaultTemplatesResolverRule.withoutGovernance();
  private UserDto user1;
  private UserDto user2;
  private GroupDto group1;
  private GroupDto group2;
  private ComponentDto project;
  private PermissionTemplateDto template1;
  private PermissionTemplateDto template2;

  private PermissionTemplateService permissionTemplateService = new
PermissionTemplateService(db.getDbClient(),
    new TestProjectIndexers(), userSession, defaultTemplatesResolver);

  @Override
  protected ApplyTemplateAction buildWsAction() {
    return new ApplyTemplateAction(db.getDbClient(), userSession, permissionTemplateService,
```

```
newPermissionWsSupport());
}

@Before
public void setUp() {
  user1 = db.users().insertUser();
  db.organizations().addMember(db.getDefaultOrganization(), user1);
  user2 = db.users().insertUser();
  db.organizations().addMember(db.getDefaultOrganization(), user2);
  group1 = db.users().insertGroup();
  group2 = db.users().insertGroup();

  // template 1
  template1 = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
  addUserToTemplate(user1, template1, UserRole.CODEVIEWER);
  addUserToTemplate(user2, template1, UserRole.ISSUE_ADMIN);
  addGroupToTemplate(group1, template1, UserRole.ADMIN);
  addGroupToTemplate(group2, template1, UserRole.USER);
  // template 2
  template2 = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
  addUserToTemplate(user1, template2, UserRole.USER);
  addUserToTemplate(user2, template2, UserRole.USER);
  addGroupToTemplate(group1, template2, UserRole.USER);
  addGroupToTemplate(group2, template2, UserRole.USER);

  project = db.components().insertPrivateProject();
  db.users().insertProjectPermissionOnUser(user1, UserRole.ADMIN, project);
  db.users().insertProjectPermissionOnUser(user2, UserRole.ADMIN, project);
  db.users().insertProjectPermissionOnGroup(group1, UserRole.ADMIN, project);
  db.users().insertProjectPermissionOnGroup(group2, UserRole.ADMIN, project);
}

@Test
public void apply_template_with_project_uuid() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest(template1.getUuid(), project.uuid(), null);

  assertTemplate1AppliedToProject();
}

@Test
public void apply_template_with_project_uuid_by_template_name() {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
    .setParam(PARAM_TEMPLATE_NAME, template1.getName().toUpperCase())
    .setParam(PARAM_PROJECT_ID, project.uuid())
```

```java
      .execute();

    assertTemplate1AppliedToProject();
  }

  @Test
  public void apply_template_with_project_key() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    newRequest(template1.getUuid(), null, project.getDbKey());

    assertTemplate1AppliedToProject();
  }

  @Test
  public void fail_when_unknown_template() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("Permission template with id 'unknown-template-uuid' is not found");

    newRequest("unknown-template-uuid", project.uuid(), null);
  }

  @Test
  public void fail_when_unknown_project_uuid() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("Project id 'unknown-project-uuid' not found");

    newRequest(template1.getUuid(), "unknown-project-uuid", null);
  }

  @Test
  public void fail_when_unknown_project_key() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("Project key 'unknown-project-key' not found");

    newRequest(template1.getUuid(), null, "unknown-project-key");
  }

  @Test
  public void fail_when_template_is_not_provided() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());
```

```java
    expectedException.expect(BadRequestException.class);

    newRequest(null, project.uuid(), null);
  }

  @Test
  public void fail_when_project_uuid_and_key_not_provided() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Project id or project key can be provided, not both.");

    newRequest(template1.getUuid(), null, null);
  }

  @Test
  public void fail_when_not_admin_of_organization() throws Exception {
    userSession.logIn().addPermission(ADMINISTER, "otherOrg");

    expectedException.expect(ForbiddenException.class);

    newRequest(template1.getUuid(), project.uuid(), null);
  }

  private void assertTemplate1AppliedToProject() {
    assertThat(selectProjectPermissionGroups(project, UserRole.ADMIN)).containsExactly(group1.getName());
    assertThat(selectProjectPermissionGroups(project, UserRole.USER)).containsExactly(group2.getName());
    assertThat(selectProjectPermissionUsers(project, UserRole.ADMIN)).isEmpty();
    assertThat(selectProjectPermissionUsers(project, UserRole.CODEVIEWER)).containsExactly(user1.getId());
    assertThat(selectProjectPermissionUsers(project, UserRole.ISSUE_ADMIN)).containsExactly(user2.getId());
  }

  private TestResponse newRequest(@Nullable String templateUuid, @Nullable String projectUuid, @Nullable
String projectKey) {
    TestRequest request = newRequest();
    if (templateUuid != null) {
      request.setParam(PARAM_TEMPLATE_ID, templateUuid);
    }
    if (projectUuid != null) {
      request.setParam(PARAM_PROJECT_ID, projectUuid);
    }
    if (projectKey != null) {
      request.setParam(PARAM_PROJECT_KEY, projectKey);
    }

    return request.execute();
  }
```

```java
 private void addUserToTemplate(UserDto user, PermissionTemplateDto permissionTemplate, String permission) {
    db.getDbClient().permissionTemplateDao().insertUserPermission(db.getSession(), permissionTemplate.getId(),
user.getId(), permission);
    db.commit();
 }

 private void addGroupToTemplate(GroupDto group, PermissionTemplateDto permissionTemplate, String
permission) {
    db.getDbClient().permissionTemplateDao().insertGroupPermission(db.getSession(), permissionTemplate.getId(),
group.getId(), permission);
    db.commit();
 }

 private List<String> selectProjectPermissionGroups(ComponentDto project, String permission) {
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(project.getOrganizationUuid()).setPermission(permission).setComp
onentUuid(project.uuid()).build();
    return db.getDbClient().groupPermissionDao().selectGroupNamesByQuery(db.getSession(), query);
 }

 private List<Integer> selectProjectPermissionUsers(ComponentDto project, String permission) {
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(project.getOrganizationUuid()).setPermission(permission).setComp
onentUuid(project.uuid()).build();
    return db.getDbClient().userPermissionDao().selectUserIdsByQuery(db.getSession(), query);
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;
```

```java
import java.util.Date;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.mockito.Mockito.spy;
import static org.mockito.Mockito.when;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateDto;
import static org.sonar.db.permission.OrganizationPermission.SCAN;
import static org.sonar.test.JsonAssert.assertJson;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_DESCRIPTION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY_PATTERN;

public class UpdateTemplateActionTest extends BasePermissionWsTest<UpdateTemplateAction> {

  private System2 system = spy(System2.INSTANCE);
  private PermissionTemplateDto template;

  @Override
  protected UpdateTemplateAction buildWsAction() {
    return new UpdateTemplateAction(db.getDbClient(), userSession, system, newPermissionWsSupport());
  }

  @Before
  public void setUp() {
    when(system.now()).thenReturn(1_440_512_328_743L);
    template = db.getDbClient().permissionTemplateDao().insert(db.getSession(), newPermissionTemplateDto()
      .setOrganizationUuid(db.getDefaultOrganization().getUuid())
      .setName("Permission Template Name")
      .setDescription("Permission Template Description")
      .setKeyPattern(".*\\.pattern\\..*")
      .setCreatedAt(new Date(1_000_000_000_000L))
      .setUpdatedAt(new Date(1_000_000_000_000L)));
    db.commit();
  }

  @Test
```

```java
  public void update_all_permission_template_fields() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    String result = call(template.getUuid(), "Finance", "Permissions for financially related projects",
".*\\.finance\\..*");

    assertJson(result)
      .ignoreFields("id")
      .isSimilarTo(getClass().getResource("update_template-example.json"));
    PermissionTemplateDto finance = selectTemplateInDefaultOrganization("Finance");
    assertThat(finance.getName()).isEqualTo("Finance");
    assertThat(finance.getDescription()).isEqualTo("Permissions for financially related projects");
    assertThat(finance.getKeyPattern()).isEqualTo(".*\\.finance\\..*");
    assertThat(finance.getUuid()).isEqualTo(template.getUuid());
    assertThat(finance.getCreatedAt()).isEqualTo(template.getCreatedAt());
    assertThat(finance.getUpdatedAt().getTime()).isEqualTo(1440512328743L);
  }

  @Test
  public void update_with_the_same_values() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    call(template.getUuid(), template.getName(), template.getDescription(), template.getKeyPattern());

    PermissionTemplateDto reloaded = db.getDbClient().permissionTemplateDao().selectByUuid(db.getSession(),
template.getUuid());
    assertThat(reloaded.getName()).isEqualTo(template.getName());
    assertThat(reloaded.getDescription()).isEqualTo(template.getDescription());
    assertThat(reloaded.getKeyPattern()).isEqualTo(template.getKeyPattern());
  }

  @Test
  public void update_name_only() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    call(template.getUuid(), "Finance", null, null);

    PermissionTemplateDto finance = selectTemplateInDefaultOrganization("Finance");
    assertThat(finance.getName()).isEqualTo("Finance");
    assertThat(finance.getDescription()).isEqualTo(template.getDescription());
    assertThat(finance.getKeyPattern()).isEqualTo(template.getKeyPattern());
  }

  @Test
  public void fail_if_key_is_not_found() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
```

```java
    expectedException.expectMessage("Permission template with id 'unknown-key' is not found");

    call("unknown-key", null, null, null);
  }

  @Test
  public void fail_if_name_already_exists_in_another_template() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());
    PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("A template with the name '" + anotherTemplate.getName() + "' already exists
(case insensitive).");

    call(this.template.getUuid(), anotherTemplate.getName(), null, null);
  }

  @Test
  public void fail_if_key_is_not_provided() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    call(null, "Finance", null, null);
  }

  @Test
  public void fail_if_name_empty() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("The template name must not be blank");

    call(template.getUuid(), "", null, null);
  }

  @Test
  public void fail_if_name_has_just_whitespaces() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("The template name must not be blank");

    call(template.getUuid(), "  \r\n", null, null);
  }

  @Test
  public void fail_if_regexp_if_not_valid() throws Exception {
```

```java
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("The 'projectKeyPattern' parameter must be a valid Java regular expression.
'[azerty' was passed");

    call(template.getUuid(), "Finance", null, "[azerty");
  }

  @Test
  public void fail_if_name_already_exists_in_database_case_insensitive() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());
    PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();

    String nameCaseInsensitive = anotherTemplate.getName().toUpperCase();
    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("A template with the name '" + nameCaseInsensitive + "' already exists (case
insensitive).");

    call(this.template.getUuid(), nameCaseInsensitive, null, null);
  }

  @Test
  public void fail_if_not_logged_in() throws Exception {
    expectedException.expect(UnauthorizedException.class);
    userSession.anonymous();

    call(template.getUuid(), "Finance", null, null);
  }

  @Test
  public void fail_if_not_admin() throws Exception {
    userSession.logIn().addPermission(SCAN, db.getDefaultOrganization());

    expectedException.expect(ForbiddenException.class);

    call(template.getUuid(), "Finance", null, null);
  }

  private String call(@Nullable String key, @Nullable String name, @Nullable String description, @Nullable String
projectPattern) {
    TestRequest request = newRequest();
    if (key != null) {
      request.setParam(PARAM_ID, key);
    }
    if (name != null) {
      request.setParam(PARAM_NAME, name);
    }
```

```java
    if (description != null) {
      request.setParam(PARAM_DESCRIPTION, description);
    }
    if (projectPattern != null) {
      request.setParam(PARAM_PROJECT_KEY_PATTERN, projectPattern);
    }

    return request.execute().getInput();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import java.util.List;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Test;
import org.sonar.api.web.UserRole;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.GroupDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.security.DefaultGroups.ANYONE;
```

```java
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class AddGroupToTemplateActionTest extends BasePermissionWsTest<AddGroupToTemplateAction> {

  private PermissionTemplateDto template;
  private GroupDto group;

  @Override
  protected AddGroupToTemplateAction buildWsAction() {
    return new AddGroupToTemplateAction(db.getDbClient(), newPermissionWsSupport(), userSession);
  }

  @Before
  public void setUp() {
    template = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
    group = db.users().insertGroup(db.getDefaultOrganization(), "group-name");
  }

  @Test
  public void add_group_to_template() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    newRequest(group.getName(), template.getUuid(), CODEVIEWER);

    assertThat(getGroupNamesInTemplateAndPermission(template,
CODEVIEWER)).containsExactly(group.getName());
  }

  @Test
  public void add_group_to_template_by_name() {
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, CODEVIEWER)
      .setParam(PARAM_TEMPLATE_NAME, template.getName().toUpperCase())
      .execute();

    assertThat(getGroupNamesInTemplateAndPermission(template,
CODEVIEWER)).containsExactly(group.getName());
  }
```

```java
@Test
public void add_with_group_id() {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
    .setParam(PARAM_TEMPLATE_ID, template.getUuid())
    .setParam(PARAM_PERMISSION, CODEVIEWER)
    .setParam(PARAM_GROUP_ID, String.valueOf(group.getId()))
    .execute();

  assertThat(getGroupNamesInTemplateAndPermission(template,
CODEVIEWER)).containsExactly(group.getName());
}

@Test
public void does_not_add_a_group_twice() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest(group.getName(), template.getUuid(), ISSUE_ADMIN);
  newRequest(group.getName(), template.getUuid(), ISSUE_ADMIN);

  assertThat(getGroupNamesInTemplateAndPermission(template,
ISSUE_ADMIN)).containsExactly(group.getName());
}

@Test
public void add_anyone_group_to_template() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest(ANYONE, template.getUuid(), CODEVIEWER);

  assertThat(getGroupNamesInTemplateAndPermission(template, CODEVIEWER)).containsExactly(ANYONE);
}

@Test
public void fail_if_add_anyone_group_to_admin_permission() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage(String.format("It is not possible to add the '%s' permission to the group
'Anyone'", UserRole.ADMIN));

  newRequest(ANYONE, template.getUuid(), ADMIN);
}

@Test
public void fail_if_not_a_project_permission() throws Exception {
```

```java
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(group.getName(), template.getUuid(), GlobalPermissions.PROVISIONING);
  }

  @Test
  public void fail_if_not_admin_of_default_organization() throws Exception {
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest(group.getName(), template.getUuid(), CODEVIEWER);
  }

  @Test
  public void fail_if_group_params_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    newRequest(null, template.getUuid(), CODEVIEWER);
  }

  @Test
  public void fail_if_permission_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(group.getName(), template.getUuid(), null);
  }

  @Test
  public void fail_if_template_uuid_and_name_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    newRequest(group.getName(), null, CODEVIEWER);
  }

  @Test
  public void fail_if_group_does_not_exist() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
```

```java
    expectedException.expectMessage("No group with name 'unknown-group-name'");

    newRequest("unknown-group-name", template.getUuid(), CODEVIEWER);
  }

  @Test
  public void fail_if_template_key_does_not_exist() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("Permission template with id 'unknown-key' is not found");

    newRequest(group.getName(), "unknown-key", CODEVIEWER);
  }

  private void newRequest(@Nullable String groupName, @Nullable String templateKey, @Nullable String
permission) {
    TestRequest request = newRequest();
    if (groupName != null) {
      request.setParam(PARAM_GROUP_NAME, groupName);
    }
    if (templateKey != null) {
      request.setParam(PARAM_TEMPLATE_ID, templateKey);
    }
    if (permission != null) {
      request.setParam(PARAM_PERMISSION, permission);
    }

    request.execute();
  }

  private List<String> getGroupNamesInTemplateAndPermission(PermissionTemplateDto template, String
permission) {
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(template.getOrganizationUuid()).setPermission(permission).build()
;
    return db.getDbClient().permissionTemplateDao()
      .selectGroupNamesByQueryAndTemplate(db.getSession(), query, template.getId());
  }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
```

```java
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
package org.sonar.server.permission.ws.template;

import java.util.Optional;
import org.junit.Before;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.permission.ws.BasePermissionWsTest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.mockito.Mockito.spy;
import static org.mockito.Mockito.when;
import static org.sonar.core.permission.GlobalPermissions.QUALITY_GATE_ADMIN;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class AddProjectCreatorToTemplateActionTest extends
BasePermissionWsTest<AddProjectCreatorToTemplateAction> {

 private System2 system = spy(System2.INSTANCE);
 private PermissionTemplateDto template;

 @Override
 protected AddProjectCreatorToTemplateAction buildWsAction() {
   return new AddProjectCreatorToTemplateAction(db.getDbClient(), newPermissionWsSupport(), userSession,
system);
 }

 @Before
 public void setUp() {
   template = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
```

```java
    when(system.now()).thenReturn(2_000_000_000L);
  }

  @Test
  public void insert_row_when_no_template_permission() {
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_PERMISSION, UserRole.ADMIN)
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .execute();

    assertThatProjectCreatorIsPresentFor(UserRole.ADMIN, template.getId());
  }

  @Test
  public void update_row_when_existing_template_permission() {
    loginAsAdmin(db.getDefaultOrganization());
    PermissionTemplateCharacteristicDto characteristic =
  db.getDbClient().permissionTemplateCharacteristicDao().insert(db.getSession(),
      new PermissionTemplateCharacteristicDto()
        .setTemplateId(template.getId())
        .setPermission(UserRole.USER)
        .setWithProjectCreator(false)
        .setCreatedAt(1_000_000_000L)
        .setUpdatedAt(1_000_000_000L));
    db.commit();
    when(system.now()).thenReturn(3_000_000_000L);

    newRequest()
      .setParam(PARAM_PERMISSION, UserRole.USER)
      .setParam(PARAM_TEMPLATE_NAME, template.getName())
      .execute();

    assertThatProjectCreatorIsPresentFor(UserRole.USER, template.getId());
    PermissionTemplateCharacteristicDto reloaded = reload(characteristic);
    assertThat(reloaded.getCreatedAt()).isEqualTo(1_000_000_000L);
    assertThat(reloaded.getUpdatedAt()).isEqualTo(3_000_000_000L);
  }

  @Test
  public void fail_when_template_does_not_exist() {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, UserRole.ADMIN)
```

```java
        .setParam(PARAM_TEMPLATE_ID, "42")
        .execute();
    }

    @Test
    public void fail_if_permission_is_not_a_project_permission() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(IllegalArgumentException.class);

      newRequest()
        .setParam(PARAM_PERMISSION, QUALITY_GATE_ADMIN)
        .setParam(PARAM_TEMPLATE_ID, template.getUuid())
        .execute();
    }

    @Test
    public void fail_if_not_admin_of_default_organization() {
      userSession.logIn().addPermission(ADMINISTER_QUALITY_GATES, db.getDefaultOrganization());

      expectedException.expect(ForbiddenException.class);

      newRequest()
        .setParam(PARAM_PERMISSION, UserRole.ADMIN)
        .setParam(PARAM_TEMPLATE_ID, template.getUuid())
        .execute();
    }

    private void assertThatProjectCreatorIsPresentFor(String permission, long templateId) {
      Optional<PermissionTemplateCharacteristicDto> templatePermission =
    db.getDbClient().permissionTemplateCharacteristicDao().selectByPermissionAndTemplateId(db.getSession(),
        permission,
        templateId);
      assertThat(templatePermission).isPresent();
      assertThat(templatePermission.get().getWithProjectCreator()).isTrue();
    }

    private PermissionTemplateCharacteristicDto reload(PermissionTemplateCharacteristicDto characteristic) {
      return
    db.getDbClient().permissionTemplateCharacteristicDao().selectByPermissionAndTemplateId(db.getSession(),
    characteristic.getPermission(), characteristic.getTemplateId()).get();
    }
    }
    /*
     * SonarQube
     * Copyright (C) 2009-2018 SonarSource SA
     * mailto:info AT sonarsource DOT com
     *
```

```
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import com.google.common.collect.HashBasedTable;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;

import static java.util.Collections.singletonList;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateDto;

public class SearchTemplatesDataTest {
 @Rule
 public ExpectedException expectedException = ExpectedException.none();

 SearchTemplatesData.Builder underTest = SearchTemplatesData.builder()
   .defaultTemplates(new DefaultTemplatesResolverImpl.ResolvedDefaultTemplates("template_uuid", null))
   .templates(singletonList(newPermissionTemplateDto()))
   .userCountByTemplateIdAndPermission(HashBasedTable.create())
   .groupCountByTemplateIdAndPermission(HashBasedTable.create())
   .withProjectCreatorByTemplateIdAndPermission(HashBasedTable.create());

 @Test
 public void fail_if_templates_is_null() {
   expectedException.expect(IllegalStateException.class);
   underTest.templates(null);

   underTest.build();
 }

 @Test
 public void fail_if_default_templates_are_null() {
   expectedException.expect(IllegalStateException.class);
   underTest.defaultTemplates(null);
```

```java
    underTest.build();
  }

  @Test
  public void fail_if_user_count_is_null() {
    expectedException.expect(IllegalStateException.class);
    underTest.userCountByTemplateIdAndPermission(null);

    underTest.build();
  }

  @Test
  public void fail_if_group_count_is_null() {
    expectedException.expect(IllegalStateException.class);
    underTest.groupCountByTemplateIdAndPermission(null);

    underTest.build();
  }

  @Test
  public void fail_if_with_project_creators_is_null() {
    expectedException.expect(IllegalStateException.class);
    underTest.withProjectCreatorByTemplateIdAndPermission(null);

    underTest.build();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;
```

```java
import java.util.Date;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Test;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.web.UserRole;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.component.ResourceTypesRule;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.i18n.I18nRule;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;
import org.sonar.server.ws.WsActionTester;
import org.sonarqube.ws.Permissions;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.server.ws.WebService.Param.TEXT_QUERY;
import static org.sonar.core.util.Uuids.UUID_EXAMPLE_01;
import static org.sonar.core.util.Uuids.UUID_EXAMPLE_02;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateDto;
import static org.sonar.test.JsonAssert.assertJson;

public class SearchTemplatesActionTest extends BasePermissionWsTest<SearchTemplatesAction> {

  private I18nRule i18n = new I18nRule();
  private DbClient dbClient = db.getDbClient();
  private DbSession dbSession = db.getSession();
  private ResourceTypesRule resourceTypesWithViews = new
ResourceTypesRule().setRootQualifiers(Qualifiers.PROJECT, Qualifiers.VIEW);
  private ResourceTypesRule resourceTypesWithoutViews = new
ResourceTypesRule().setRootQualifiers(Qualifiers.PROJECT);

  private WsActionTester underTestWithoutViews;

  @Override
  protected SearchTemplatesAction buildWsAction() {
    DefaultTemplatesResolver defaultTemplatesResolverWithViews = new
DefaultTemplatesResolverImpl(resourceTypesWithViews);
    SearchTemplatesAction searchTemplatesAction = new SearchTemplatesAction(dbClient, userSession, i18n,
newPermissionWsSupport(), defaultTemplatesResolverWithViews);
    return searchTemplatesAction;
```

```
  }

  @Before
  public void setUp() {
    DefaultTemplatesResolver defaultTemplatesResolverWithViews = new
DefaultTemplatesResolverImpl(resourceTypesWithoutViews);
    underTestWithoutViews = new WsActionTester(new SearchTemplatesAction(dbClient, userSession, i18n,
newPermissionWsSupport(), defaultTemplatesResolverWithViews));
    i18n.setProjectPermissions();
    userSession.logIn().addPermission(ADMINISTER, db.getDefaultOrganization());
  }

  @Test
  public void search_project_permissions() {
    OrganizationDto organization = db.getDefaultOrganization();
    PermissionTemplateDto projectTemplate = insertProjectTemplate(organization);
    PermissionTemplateDto viewsTemplate = insertViewsTemplate(organization);

    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    UserDto user3 = db.users().insertUser();

    GroupDto group1 = db.users().insertGroup(organization);
    GroupDto group2 = db.users().insertGroup(organization);
    GroupDto group3 = db.users().insertGroup(organization);

    addUserToTemplate(projectTemplate.getId(), user1.getId(), UserRole.ISSUE_ADMIN);
    addUserToTemplate(projectTemplate.getId(), user2.getId(), UserRole.ISSUE_ADMIN);
    addUserToTemplate(projectTemplate.getId(), user3.getId(), UserRole.ISSUE_ADMIN);
    addUserToTemplate(projectTemplate.getId(), user1.getId(), UserRole.CODEVIEWER);
    addGroupToTemplate(projectTemplate.getId(), group1.getId(), UserRole.ADMIN);
    addPermissionTemplateWithProjectCreator(projectTemplate.getId(), UserRole.ADMIN);

    addUserToTemplate(viewsTemplate.getId(), user1.getId(), UserRole.USER);
    addUserToTemplate(viewsTemplate.getId(), user2.getId(), UserRole.USER);
    addGroupToTemplate(viewsTemplate.getId(), group1.getId(), UserRole.ISSUE_ADMIN);
    addGroupToTemplate(viewsTemplate.getId(), group2.getId(), UserRole.ISSUE_ADMIN);
    addGroupToTemplate(viewsTemplate.getId(), group3.getId(), UserRole.ISSUE_ADMIN);

    db.organizations().setDefaultTemplates(projectTemplate, viewsTemplate);

    String result = newRequest().execute().getInput();

    assertJson(result)
      .withStrictArrayOrder()
      .isSimilarTo(getClass().getResource("search_templates-example.json"));
  }
```

```
@Test
public void empty_result_with_views() {
  db.organizations().setDefaultTemplates(db.getDefaultOrganization(), "AU-Tpxb--iU5OvuD2FLy", "AU-TpxcA-
iU5OvuD2FLz");
  String result = newRequest(wsTester).execute().getInput();

  assertJson(result)
    .withStrictArrayOrder()
    .ignoreFields("permissions")
    .isSimilarTo("{" +
    " \"permissionTemplates\": []," +
    " \"defaultTemplates\": [" +
    "   {" +
    "     \"templateId\": \"AU-Tpxb--iU5OvuD2FLy\"," +
    "     \"qualifier\": \"TRK\"" +
    "   }," +
    "   {" +
    "     \"templateId\": \"AU-TpxcA-iU5OvuD2FLz\"," +
    "     \"qualifier\": \"VW\"" +
    "   }" +
    " ]" +
    "}");
}

@Test
public void empty_result_without_views() {
  db.organizations().setDefaultTemplates(db.getDefaultOrganization(), "AU-Tpxb--iU5OvuD2FLy", "AU-TpxcA-
iU5OvuD2FLz");
  String result = newRequest(underTestWithoutViews).execute().getInput();

  assertJson(result)
    .withStrictArrayOrder()
    .ignoreFields("permissions")
    .isSimilarTo("{" +
    " \"permissionTemplates\": []," +
    " \"defaultTemplates\": [" +
    "   {" +
    "     \"templateId\": \"AU-Tpxb--iU5OvuD2FLy\"," +
    "     \"qualifier\": \"TRK\"" +
    "   }" +
    " ]" +
    "}");
}

@Test
public void search_by_name_in_default_organization() {
  db.organizations().setDefaultTemplates(db.permissionTemplates().insertTemplate(db.getDefaultOrganization()),
null);
```

```java
    insertProjectTemplate(db.getDefaultOrganization());
    insertViewsTemplate(db.getDefaultOrganization());

    String result = newRequest(wsTester)
      .setParam(TEXT_QUERY, "views")
      .execute()
      .getInput();

    assertThat(result).contains("Default template for Views")
      .doesNotContain("projects")
      .doesNotContain("developers");
  }

  @Test
  public void search_in_organization() {
    OrganizationDto org = db.organizations().insert();
    PermissionTemplateDto projectDefaultTemplate = db.permissionTemplates().insertTemplate(org);
    db.organizations().setDefaultTemplates(projectDefaultTemplate, null);
    PermissionTemplateDto templateInOrg = insertProjectTemplate(org);
    insertProjectTemplate(db.getDefaultOrganization());
    db.commit();
    userSession.addPermission(ADMINISTER, org);

    Permissions.SearchTemplatesWsResponse result = newRequest(underTestWithoutViews)
      .setParam("organization", org.getKey())
      .executeProtobuf(Permissions.SearchTemplatesWsResponse.class);

    assertThat(result.getPermissionTemplatesCount()).isEqualTo(2);
    assertThat(result.getPermissionTemplatesList())
      .extracting(Permissions.PermissionTemplate::getId)
      .containsOnly(projectDefaultTemplate.getUuid(), templateInOrg.getUuid());
  }

  @Test
  public void fail_if_not_logged_in() {
    expectedException.expect(UnauthorizedException.class);
    userSession.anonymous();

    newRequest().execute();
  }

  @Test
  public void display_all_project_permissions() {
    db.organizations().setDefaultTemplates(db.permissionTemplates().insertTemplate(db.getDefaultOrganization()),
null);

    String result = newRequest().execute().getInput();
```

```java
    assertJson(result)
      .withStrictArrayOrder()
      .ignoreFields("defaultTemplates", "permissionTemplates")
      .isSimilarTo(
        "{" +
        "  \"permissions\": [" +
        "    {" +
        "      \"key\": \"admin\"," +
        "      \"name\": \"Administer\"," +
        "      \"description\": \"Ability to access project settings and perform administration tasks. (Users will also need
\\\"Browse\\\" permission)\"" +
        "    }," +
        "    {" +
        "      \"key\": \"codeviewer\"," +
        "      \"name\": \"See Source Code\"," +
        "      \"description\": \"Ability to view the project\\u0027s source code. (Users will also need \\\"Browse\\\"
permission)\"" +
        "    }," +
        "    {" +
        "      \"key\": \"issueadmin\"," +
        "      \"name\": \"Administer Issues\"," +
        "      \"description\": \"Grants the permission to perform advanced editing on issues: marking an issue False
Positive / Won\\u0027t Fix or changing an Issue\\u0027s severity. (Users will also need \\\"Browse\\\"
permission)\""
        +
        "    }," +
        "    {" +
        "      \"key\": \"scan\"," +
        "      \"name\": \"Execute Analysis\"," +
        "      \"description\": \"Ability to execute analyses, and to get all settings required to perform the analysis, even
the secured ones like the scm account password, the jira account password, and so on.\""
        +
        "    }," +
        "    {" +
        "      \"key\": \"user\"," +
        "      \"name\": \"Browse\"," +
        "      \"description\": \"Ability to access a project, browse its measures, and create/edit issues for it.\"" +
        "    }" +
        "  ]" +
        "}");
  }

  private PermissionTemplateDto insertProjectTemplate(OrganizationDto org) {
    return insertTemplate(newPermissionTemplateDto()
      .setOrganizationUuid(org.getUuid())
      .setUuid(UUID_EXAMPLE_01)
      .setName("Default template for Projects")
      .setDescription("Template for new projects")
```

```java
      .setKeyPattern(null)
      .setCreatedAt(new Date(1_000_000_000_000L))
      .setUpdatedAt(new Date(1_000_000_000_000L)));
  }

  private PermissionTemplateDto insertViewsTemplate(OrganizationDto organization) {
    return insertTemplate(newPermissionTemplateDto()
      .setOrganizationUuid(organization.getUuid())
      .setUuid(UUID_EXAMPLE_02)
      .setName("Default template for Views")
      .setDescription("Template for new views")
      .setKeyPattern(".*sonar.views.*")
      .setCreatedAt(new Date(1_000_000_000_000L))
      .setUpdatedAt(new Date(1_100_000_000_000L)));
  }

  private PermissionTemplateDto insertTemplate(PermissionTemplateDto template) {
    PermissionTemplateDto insert = dbClient.permissionTemplateDao().insert(db.getSession(), template);
    db.getSession().commit();
    return insert;
  }

  private void addGroupToTemplate(long templateId, @Nullable Integer groupId, String permission) {
    dbClient.permissionTemplateDao().insertGroupPermission(db.getSession(), templateId, groupId, permission);
    db.getSession().commit();
  }

  private void addUserToTemplate(long templateId, int userId, String permission) {
    dbClient.permissionTemplateDao().insertUserPermission(db.getSession(), templateId, userId, permission);
    db.getSession().commit();
  }

  private void addPermissionTemplateWithProjectCreator(long templateId, String permission) {
    dbClient.permissionTemplateCharacteristicDao().insert(dbSession, new PermissionTemplateCharacteristicDto()
      .setWithProjectCreator(true)
      .setTemplateId(templateId)
      .setPermission(permission)
      .setCreatedAt(1_000_000_000L)
      .setUpdatedAt(2_000_000_000L));
    db.commit();
  }

  private TestRequest newRequest(WsActionTester underTest) {
    return underTest.newRequest().setMethod("POST");
  }
}
/*
 * SonarQube
```

```
package org.sonar.server.permission.ws.template;

import java.util.stream.Stream;
import org.junit.Test;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.resources.ResourceType;
import org.sonar.api.resources.ResourceTypeTree;
import org.sonar.api.resources.ResourceTypes;
import org.sonar.db.organization.DefaultTemplates;

import static org.assertj.core.api.Assertions.assertThat;

public class DefaultTemplatesResolverImplTest {

 private static final ResourceTypes RESOURCE_TYPES_WITHOUT_VIEWS = new ResourceTypes(new
ResourceTypeTree[] {
  ResourceTypeTree.builder().addType(ResourceType.builder(Qualifiers.PROJECT).build()).build()
 });
 private static final ResourceTypes RESOURCE_TYPES_WITH_VIEWS = new ResourceTypes(new
ResourceTypeTree[] {
  ResourceTypeTree.builder().addType(ResourceType.builder(Qualifiers.PROJECT).build()).build(),
  ResourceTypeTree.builder().addType(ResourceType.builder(Qualifiers.VIEW).build()).build()
 });
 private DefaultTemplatesResolverImpl underTestWithoutViews = new
DefaultTemplatesResolverImpl(RESOURCE_TYPES_WITHOUT_VIEWS);
 private DefaultTemplatesResolverImpl underTestWithViews = new
DefaultTemplatesResolverImpl(RESOURCE_TYPES_WITH_VIEWS);

 @Test
 public void project_is_project_of_DefaultTemplates_no_matter_if_views_is_installed() {
  Stream.of(
```

```java
    new DefaultTemplates().setProjectUuid("foo").setViewUuid(null),
    new DefaultTemplates().setProjectUuid("foo").setViewUuid("bar")).forEach(
      defaultTemplates -> {
        assertThat(underTestWithoutViews.resolve(defaultTemplates).getProject()).isEqualTo("foo");
        assertThat(underTestWithViews.resolve(defaultTemplates).getProject()).isEqualTo("foo");
      });
  }

  @Test
  public void view_is_empty_no_matter_view_in_DefaultTemplates_if_views_is_not_installed() {
    DefaultTemplates defaultTemplatesNoView = new DefaultTemplates().setProjectUuid("foo").setViewUuid(null);
    DefaultTemplates defaultTemplatesView = new DefaultTemplates().setProjectUuid("foo").setViewUuid("bar");

    assertThat(underTestWithoutViews.resolve(defaultTemplatesNoView).getView()).isEmpty();
    assertThat(underTestWithoutViews.resolve(defaultTemplatesView).getView()).isEmpty();
  }

  @Test
  public void view_is_project_of_DefaultTemplates_if_view_in_DefaultTemplates_is_null_and_views_is_installed()
  {
    DefaultTemplates defaultTemplates = new DefaultTemplates().setProjectUuid("foo").setViewUuid(null);

    assertThat(underTestWithViews.resolve(defaultTemplates).getView()).contains("foo");
  }

  @Test
  public void
  view_is_view_of_DefaultTemplates_if_view_in_DefaultTemplates_is_not_null_and_views_is_installed() {
    DefaultTemplates defaultTemplates = new DefaultTemplates().setProjectUuid("foo").setViewUuid("bar");

    assertThat(underTestWithViews.resolve(defaultTemplates).getView()).contains("bar");
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
```

```java
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
package org.sonar.server.permission.ws.template;

import java.util.List;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Test;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_PROFILES;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class AddUserToTemplateActionTest extends BasePermissionWsTest<AddUserToTemplateAction> {

  private UserDto user;
  private PermissionTemplateDto permissionTemplate;

  @Override
  protected AddUserToTemplateAction buildWsAction() {
    return new AddUserToTemplateAction(db.getDbClient(), newPermissionWsSupport(), userSession);
  }

  @Before
  public void setUp() {
    user = db.users().insertUser("user-login");
    db.organizations().addMember(db.getDefaultOrganization(), user);
    permissionTemplate = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
  }

  @Test
```

```java
public void add_user_to_template() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest(user.getLogin(), permissionTemplate.getUuid(), CODEVIEWER);

  assertThat(getLoginsInTemplateAndPermission(permissionTemplate,
CODEVIEWER)).containsExactly(user.getLogin());
}

@Test
public void add_user_to_template_by_name() {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, CODEVIEWER)
    .setParam(PARAM_TEMPLATE_NAME, permissionTemplate.getName().toUpperCase())
    .execute();

  assertThat(getLoginsInTemplateAndPermission(permissionTemplate,
CODEVIEWER)).containsExactly(user.getLogin());
}

@Test
public void add_user_to_template_by_name_and_organization() {
  OrganizationDto organizationDto = db.organizations().insert();
  PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate(organizationDto);
  addUserAsMemberOfOrganization(organizationDto);
  loginAsAdmin(organizationDto);

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, CODEVIEWER)
    .setParam(PARAM_TEMPLATE_NAME, permissionTemplate.getName().toUpperCase())
    .setParam(PARAM_ORGANIZATION, organizationDto.getKey())
    .execute();

  assertThat(getLoginsInTemplateAndPermission(permissionTemplate,
CODEVIEWER)).containsExactly(user.getLogin());
}

@Test
public void does_not_add_a_user_twice() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  newRequest(user.getLogin(), permissionTemplate.getUuid(), ISSUE_ADMIN);
  newRequest(user.getLogin(), permissionTemplate.getUuid(), ISSUE_ADMIN);
```

```java
    assertThat(getLoginsInTemplateAndPermission(permissionTemplate,
  ISSUE_ADMIN)).containsExactly(user.getLogin());
  }

  @Test
  public void fail_if_not_a_project_permission() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(user.getLogin(), permissionTemplate.getUuid(), GlobalPermissions.PROVISIONING);
  }

  @Test
  public void fail_if_not_admin_of_default_organization() throws Exception {
    userSession.logIn().addPermission(ADMINISTER_QUALITY_PROFILES, db.getDefaultOrganization());

    expectedException.expect(ForbiddenException.class);

    newRequest(user.getLogin(), permissionTemplate.getUuid(), CODEVIEWER);
  }

  @Test
  public void fail_if_user_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(null, permissionTemplate.getUuid(), CODEVIEWER);
  }

  @Test
  public void fail_if_permission_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(user.getLogin(), permissionTemplate.getUuid(), null);
  }

  @Test
  public void fail_if_template_uuid_and_name_are_missing() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    newRequest(user.getLogin(), null, CODEVIEWER);
  }
```

```java
@Test
public void fail_if_user_does_not_exist() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage("User with login 'unknown-login' is not found");

  newRequest("unknown-login", permissionTemplate.getUuid(), CODEVIEWER);
}

@Test
public void fail_if_template_key_does_not_exist() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage("Permission template with id 'unknown-key' is not found");

  newRequest(user.getLogin(), "unknown-key", CODEVIEWER);
}

@Test
public void fail_if_organization_does_not_exist() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage("No organization with key 'Unknown'");

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, CODEVIEWER)
    .setParam(PARAM_TEMPLATE_NAME, permissionTemplate.getName().toUpperCase())
    .setParam(PARAM_ORGANIZATION, "Unknown")
    .execute();  }

@Test
public void fail_to_add_permission_when_user_is_not_member_of_given_organization() {
  // User is not member of given organization
  OrganizationDto otherOrganization = db.organizations().insert();
  addUserAsMemberOfOrganization(otherOrganization);
  OrganizationDto organization = db.organizations().insert(organizationDto ->
organizationDto.setKey("Organization key"));
  PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate(organization);
  loginAsAdmin(organization);

  expectedException.expect(IllegalArgumentException.class);
  expectedException.expectMessage("User 'user-login' is not member of organization 'Organization key'");
```

```java
  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, CODEVIEWER)
    .setParam(PARAM_TEMPLATE_NAME, permissionTemplate.getName().toUpperCase())
    .setParam(PARAM_ORGANIZATION, organization.getKey())
    .execute();
}

private void newRequest(@Nullable String userLogin, @Nullable String templateKey, @Nullable String
permission) {
  TestRequest request = newRequest();
  if (userLogin != null) {
    request.setParam(PARAM_USER_LOGIN, userLogin);
  }
  if (templateKey != null) {
    request.setParam(org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID,
templateKey);
  }
  if (permission != null) {
    request.setParam(PARAM_PERMISSION, permission);
  }

  request.execute();
}

private List<String> getLoginsInTemplateAndPermission(PermissionTemplateDto template, String permission) {
  PermissionQuery permissionQuery =
PermissionQuery.builder().setOrganizationUuid(template.getOrganizationUuid()).setPermission(permission).build()
;
  return db.getDbClient().permissionTemplateDao()
    .selectUserLoginsByQueryAndTemplate(db.getSession(), permissionQuery, template.getId());
}

private void addUserAsMemberOfOrganization(OrganizationDto organization) {
  db.organizations().addMember(organization, user);
}
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
```

package org.sonar.server.permission.ws.template;

import java.util.Arrays;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.utils.internal.AlwaysIncreasingSystem2;
import org.sonar.api.web.UserRole;
import org.sonar.db.DbClient;
import org.sonar.db.DbTester;
import org.sonar.db.component.ResourceTypesRule;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.permission.template.PermissionTemplateTesting;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.GroupTesting;
import org.sonar.db.user.UserDto;
import org.sonar.db.user.UserTesting;
import org.sonar.server.component.ComponentFinder;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.organization.TestDefaultOrganizationProvider;
import org.sonar.server.permission.ws.PermissionWsSupport;
import org.sonar.server.tester.UserSessionRule;
import org.sonar.server.usergroups.DefaultGroupFinder;
import org.sonar.server.usergroups.ws.GroupWsSupport;
import org.sonar.server.ws.TestRequest;
import org.sonar.server.ws.TestResponse;
import org.sonar.server.ws.WsActionTester;

import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.fail;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;

```java
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class DeleteTemplateActionTest {

  @Rule
  public DbTester db = DbTester.create(new AlwaysIncreasingSystem2());
  @Rule
  public ExpectedException expectedException = ExpectedException.none();

  private UserSessionRule userSession = UserSessionRule.standalone();
  private DbClient dbClient = db.getDbClient();
  private final ResourceTypesRule resourceTypes = new
ResourceTypesRule().setRootQualifiers(Qualifiers.PROJECT);
  private final ResourceTypesRule resourceTypesWithViews = new
ResourceTypesRule().setRootQualifiers(Qualifiers.PROJECT, Qualifiers.VIEW);
  private DefaultTemplatesResolver defaultTemplatesResolver = new
DefaultTemplatesResolverImpl(resourceTypes);
  private DefaultTemplatesResolver defaultTemplatesResolverWithViews = new
DefaultTemplatesResolverImpl(resourceTypesWithViews);

  private WsActionTester underTestWithoutViews;
  private WsActionTester underTestWithViews;

  @Before
  public void setUp() throws Exception {
    GroupWsSupport groupWsSupport = new GroupWsSupport(dbClient, TestDefaultOrganizationProvider.from(db),
new DefaultGroupFinder(db.getDbClient()));
    this.underTestWithoutViews = new WsActionTester(new DeleteTemplateAction(dbClient, userSession,
      new PermissionWsSupport(dbClient, new ComponentFinder(dbClient, resourceTypes), groupWsSupport),
      defaultTemplatesResolver));
    this.underTestWithViews = new WsActionTester(new DeleteTemplateAction(dbClient, userSession,
      new PermissionWsSupport(dbClient, new ComponentFinder(dbClient, resourceTypes), groupWsSupport),
      defaultTemplatesResolverWithViews));
  }

  @Test
  public void delete_template_in_db() throws Exception {
    runOnAllUnderTests((underTest) -> {
      OrganizationDto organization = db.organizations().insert();
      PermissionTemplateDto template = insertTemplateAndAssociatedPermissions(organization);
      db.organizations().setDefaultTemplates(
        db.permissionTemplates().insertTemplate(organization),
        db.permissionTemplates().insertTemplate(organization));
      loginAsAdmin(organization);

      TestResponse result = newRequestByUuid(underTest, template.getUuid());

      assertThat(result.getInput()).isEmpty();
```

```java
        assertTemplateDoesNotExist(template);
      });
    }

    @Test
    public void delete_template_by_name_case_insensitive() throws Exception {
      runOnAllUnderTests((underTest) -> {
        OrganizationDto organization = db.organizations().insert();
        db.organizations().setDefaultTemplates(
          db.permissionTemplates().insertTemplate(organization),
          db.permissionTemplates().insertTemplate(organization));
        PermissionTemplateDto template = insertTemplateAndAssociatedPermissions(organization);
        loginAsAdmin(organization);
        newRequestByName(underTest, organization, template);

        assertTemplateDoesNotExist(template);
      });
    }

    @Test
    public void
    delete_template_by_name_returns_empty_when_no_organization_is_provided_and_templates_does_not_belong_to
    _default_organization() throws Exception {
      OrganizationDto organization = db.organizations().insert();
      db.organizations().setDefaultTemplates(
        db.permissionTemplates().insertTemplate(organization),
        db.permissionTemplates().insertTemplate(organization));
      PermissionTemplateDto template = insertTemplateAndAssociatedPermissions(organization);
      loginAsAdmin(organization);

      runOnAllUnderTests((underTest) -> {
        try {
          newRequestByName(underTest, null, template);
          fail("NotFoundException should have been raised");
        } catch (NotFoundException e) {
          assertThat(e).hasMessage(
            "Permission template with name '" + template.getName() + "' is not found (case insensitive) in organization
    with key '" + db.getDefaultOrganization().getKey() + "'");
        }
      });
    }

    @Test
    public void delete_template_by_name_returns_empty_when_wrong_organization_is_provided() throws Exception
    {
      OrganizationDto organization = db.organizations().insert();
      db.organizations().setDefaultTemplates(
        db.permissionTemplates().insertTemplate(organization),
```

```java
    db.permissionTemplates().insertTemplate(organization));
  PermissionTemplateDto template = insertTemplateAndAssociatedPermissions(organization);
  OrganizationDto otherOrganization = db.organizations().insert();
  loginAsAdmin(organization);

  runOnAllUnderTests((underTest) -> {
   try {
    newRequestByName(underTest, otherOrganization, template);
    fail("NotFoundException should have been raised");
   } catch (NotFoundException e) {
    assertThat(e)
      .hasMessage("Permission template with name '" + template.getName() + "' is not found (case insensitive) in
organization with key '" + otherOrganization.getKey() + "'");
   }
  });
 }

 @Test
 public void fail_if_uuid_is_not_known_without_views() throws Exception {
  userSession.logIn();

  expectedException.expect(NotFoundException.class);

  newRequestByUuid(underTestWithoutViews, "unknown-template-uuid");
 }

 @Test
 public void fail_if_uuid_is_not_known_with_views() throws Exception {
  userSession.logIn();

  expectedException.expect(NotFoundException.class);

  newRequestByUuid(underTestWithViews, "unknown-template-uuid");
 }

 @Test
 public void fail_to_delete_by_uuid_if_template_is_default_template_for_project_without_views() throws
Exception {
  fail_to_delete_by_uuid_if_template_is_default_template_for_project(this.underTestWithoutViews);
 }

 @Test
 public void fail_to_delete_by_uuid_if_template_is_default_template_for_project_with_views() throws Exception {
  fail_to_delete_by_uuid_if_template_is_default_template_for_project(this.underTestWithViews);
 }

 private void fail_to_delete_by_uuid_if_template_is_default_template_for_project(WsActionTester underTest)
throws Exception {
```

```java
  OrganizationDto organization = db.organizations().insert();
  PermissionTemplateDto projectTemplate = insertTemplateAndAssociatedPermissions(organization);
  db.organizations().setDefaultTemplates(projectTemplate,
    db.permissionTemplates().insertTemplate(organization));
  loginAsAdmin(organization);

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("It is not possible to delete the default permission template for projects");

  newRequestByUuid(underTest, projectTemplate.getUuid());
}

@Test
public void fail_to_delete_by_name_if_template_is_default_template_for_project_without_views() throws
Exception {
  fail_to_delete_by_name_if_template_is_default_template_for_project(this.underTestWithoutViews);
}

@Test
public void fail_to_delete_by_name_if_template_is_default_template_for_project_with_views() throws Exception
{
  fail_to_delete_by_name_if_template_is_default_template_for_project(this.underTestWithViews);
}

private void fail_to_delete_by_name_if_template_is_default_template_for_project(WsActionTester underTest)
throws Exception {
  OrganizationDto organization = db.organizations().insert();
  PermissionTemplateDto projectTemplate = insertTemplateAndAssociatedPermissions(organization);
  db.organizations().setDefaultTemplates(projectTemplate,
db.permissionTemplates().insertTemplate(organization));
  loginAsAdmin(organization);

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("It is not possible to delete the default permission template for projects");

  newRequestByName(underTest, organization.getKey(), projectTemplate.getName());
}

@Test
public void fail_to_delete_by_uuid_if_template_is_default_template_for_view_with_views() throws Exception {
  OrganizationDto organization = db.organizations().insert();
  PermissionTemplateDto template = insertTemplateAndAssociatedPermissions(organization);
  db.organizations().setDefaultTemplates(db.permissionTemplates().insertTemplate(organization), template);
  loginAsAdmin(organization);

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("It is not possible to delete the default permission template for views");
```

```java
    newRequestByUuid(this.underTestWithViews, template.getUuid());
  }

  @Test
  public void
default_template_for_views_can_be_deleted_by_uuid_if_views_is_not_installed_and_default_template_for_views_
is_reset() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto projectTemplate = db.permissionTemplates().insertTemplate(organization);
    PermissionTemplateDto viewTemplate = insertTemplateAndAssociatedPermissions(organization);
    db.organizations().setDefaultTemplates(projectTemplate, viewTemplate);
    loginAsAdmin(organization);

    newRequestByUuid(this.underTestWithoutViews, viewTemplate.getUuid());

    assertTemplateDoesNotExist(viewTemplate);

    assertThat(db.getDbClient().organizationDao().getDefaultTemplates(db.getSession(), organization.getUuid())
      .get().getViewUuid())
        .isNull();
  }

  @Test
  public void fail_to_delete_by_uuid_if_not_logged_in_without_views() throws Exception {
    expectedException.expect(UnauthorizedException.class);

    newRequestByUuid(underTestWithoutViews, "uuid");
  }

  @Test
  public void fail_to_delete_by_uuid_if_not_logged_in_with_views() throws Exception {
    expectedException.expect(UnauthorizedException.class);

    newRequestByUuid(underTestWithViews, "uuid");
  }

  @Test
  public void fail_to_delete_by_name_if_not_logged_in_without_views() throws Exception {
    expectedException.expect(UnauthorizedException.class);

    newRequestByName(underTestWithoutViews, "whatever", "name");
  }

  @Test
  public void fail_to_delete_by_name_if_not_logged_in_with_views() throws Exception {
    expectedException.expect(UnauthorizedException.class);

    newRequestByName(underTestWithViews, "whatever", "name");
```

```java
}

@Test
public void fail_to_delete_by_uuid_if_not_admin_without_views() throws Exception {
  OrganizationDto organization = db.organizations().insert();
  PermissionTemplateDto template = insertTemplateAndAssociatedPermissions(organization);
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequestByUuid(underTestWithoutViews, template.getUuid());
}

@Test
public void fail_to_delete_by_uuid_if_not_admin_with_views() throws Exception {
  OrganizationDto organization = db.organizations().insert();
  PermissionTemplateDto template = insertTemplateAndAssociatedPermissions(organization);
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequestByUuid(underTestWithViews, template.getUuid());
}

@Test
public void fail_to_delete_by_name_if_not_admin_without_views() throws Exception {
  OrganizationDto organization = db.organizations().insert();
  PermissionTemplateDto template = db.permissionTemplates().insertTemplate(organization);
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequestByName(underTestWithoutViews, organization.getKey(), template.getName());
}

@Test
public void fail_to_delete_by_name_if_not_admin_with_views() throws Exception {
  OrganizationDto organization = db.organizations().insert();
  PermissionTemplateDto template =
db.permissionTemplates().insertTemplate(PermissionTemplateTesting.newPermissionTemplateDto()
    .setOrganizationUuid(organization.getUuid())
    .setName("the name"));
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequestByName(underTestWithViews, organization, template);
}
```

```java
@Test
public void fail_if_neither_uuid_nor_name_is_provided_without_views() throws Exception {
  userSession.logIn();

  expectedException.expect(BadRequestException.class);

  newRequestByUuid(underTestWithoutViews, null);
}

@Test
public void fail_if_neither_uuid_nor_name_is_provided_with_views() throws Exception {
  userSession.logIn();

  expectedException.expect(BadRequestException.class);

  newRequestByUuid(underTestWithViews, null);
}

@Test
public void fail_if_both_uuid_and_name_are_provided_without_views() {
  userSession.logIn();

  expectedException.expect(BadRequestException.class);

  underTestWithoutViews.newRequest().setMethod("POST")
    .setParam(PARAM_TEMPLATE_ID, "uuid")
    .setParam(PARAM_TEMPLATE_NAME, "name")
    .execute();
}

@Test
public void fail_if_both_uuid_and_name_are_provided_with_views() {
  userSession.logIn();

  expectedException.expect(BadRequestException.class);

  underTestWithViews.newRequest().setMethod("POST")
    .setParam(PARAM_TEMPLATE_ID, "uuid")
    .setParam(PARAM_TEMPLATE_NAME, "name")
    .execute();
}

// @Test
// public void delete_perm_tpl_characteristic_when_delete_template() throws Exception {
// db.getDbClient().permissionTemplateCharacteristicDao().insert(db.getSession(), new
PermissionTemplateCharacteristicDto()
// .setPermission(UserRole.USER)
```

```java
//    .setTemplateId(template.getId())
//    .setWithProjectCreator(true)
//    .setCreatedAt(new Date().getTime())
//    .setUpdatedAt(new Date().getTime()));
// db.commit();
//
// newRequest(template.getUuid());
//
// assertThat(db.getDbClient().permissionTemplateCharacteristicDao().selectByTemplateIds(db.getSession(),
// asList(template.getId()))).isEmpty();
// }

private UserSessionRule loginAsAdmin(OrganizationDto organization) {
  return userSession.logIn().addPermission(ADMINISTER, organization);
}

private void runOnAllUnderTests(ConsumerWithException<WsActionTester> consumer) throws Exception {
  for (WsActionTester underTest : Arrays.asList(underTestWithoutViews, underTestWithViews)) {
    consumer.accept(underTest);
  }
}

private interface ConsumerWithException<T> {
  void accept(T e) throws Exception;
}

private PermissionTemplateDto insertTemplateAndAssociatedPermissions(OrganizationDto organization) {
  PermissionTemplateDto dto = db.permissionTemplates().insertTemplate(organization);
  UserDto user = db.getDbClient().userDao().insert(db.getSession(), UserTesting.newUserDto().setActive(true));
  GroupDto group = db.getDbClient().groupDao().insert(db.getSession(), GroupTesting.newGroupDto());
  db.getDbClient().permissionTemplateDao().insertUserPermission(db.getSession(), dto.getId(), user.getId(),
UserRole.ADMIN);
  db.getDbClient().permissionTemplateDao().insertGroupPermission(db.getSession(), dto.getId(), group.getId(),
UserRole.CODEVIEWER);
  db.commit();
  return dto;
}

private TestResponse newRequestByUuid(WsActionTester actionTester, @Nullable String id) {
  TestRequest request = actionTester.newRequest().setMethod("POST");
  if (id != null) {
    request.setParam(PARAM_TEMPLATE_ID, id);
  }
  return request.execute();
}

private TestResponse newRequestByName(WsActionTester actionTester, @Nullable OrganizationDto
organizationDto, @Nullable PermissionTemplateDto permissionTemplateDto)
```

```java
  throws Exception {
  return newRequestByName(
    actionTester,
    organizationDto == null ? null : organizationDto.getKey(),
    permissionTemplateDto == null ? null : permissionTemplateDto.getName());
 }

 private TestResponse newRequestByName(WsActionTester actionTester, @Nullable String organizationKey,
@Nullable String name) {
  TestRequest request = actionTester.newRequest().setMethod("POST");
  if (organizationKey != null) {
    request.setParam(PARAM_ORGANIZATION, organizationKey);
  }
  if (name != null) {
    request.setParam(PARAM_TEMPLATE_NAME, name);
  }

  return request.execute();
 }

 private void assertTemplateDoesNotExist(PermissionTemplateDto template) {
   assertThat(db.getDbClient().permissionTemplateDao().selectByUuid(db.getSession(),
template.getUuid())).isNull();
 }

}
```

/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
*/
package org.sonar.server.permission.ws.template;

import javax.annotation.Nullable;

```java
import org.junit.Test;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.permission.template.PermissionTemplateGroupDto;
import org.sonar.db.user.GroupDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonarqube.ws.Permissions.WsGroupsResponse;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.server.ws.WebService.Param.PAGE;
import static org.sonar.api.server.ws.WebService.Param.PAGE_SIZE;
import static org.sonar.api.server.ws.WebService.Param.TEXT_QUERY;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateGroupDto;
import static org.sonar.db.user.GroupTesting.newGroupDto;
import static org.sonar.test.JsonAssert.assertJson;
import static org.sonarqube.ws.MediaTypes.PROTOBUF;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class TemplateGroupsActionTest extends BasePermissionWsTest<TemplateGroupsAction> {

  @Override
  protected TemplateGroupsAction buildWsAction() {
    return new TemplateGroupsAction(db.getDbClient(), userSession, newPermissionWsSupport());
  }

  @Test
  public void template_groups_of_json_example() {
    GroupDto adminGroup = insertGroupOnDefaultOrganization("sonar-administrators", "System administrators");
    GroupDto userGroup = insertGroupOnDefaultOrganization("sonar-users", "Any new users created will
automatically join this group");

    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    addGroupToTemplate(newPermissionTemplateGroup(ISSUE_ADMIN, template.getId(), adminGroup.getId()));
    addGroupToTemplate(newPermissionTemplateGroup(ISSUE_ADMIN, template.getId(), userGroup.getId()));
    // Anyone group
    addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), null));
    addGroupToTemplate(newPermissionTemplateGroup(ISSUE_ADMIN, template.getId(), null));
```

```java
      commit();
      loginAsAdmin(db.getDefaultOrganization());

      String response = newRequest()
        .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
        .setParam(PARAM_TEMPLATE_ID, template.getUuid())
        .execute()
        .getInput();

      assertJson(response)
        .ignoreFields("id")
        .withStrictArrayOrder()
        .isSimilarTo(getClass().getResource("template_groups-example.json"));
    }

    @Test
    public void do_not_fail_when_group_name_exists_in_multiple_organizations() {
      PermissionTemplateDto template = addTemplateToDefaultOrganization();

      String groupName = "group-name";
      GroupDto group1 = db.users().insertGroup(db.getDefaultOrganization(), groupName);
      addGroupToTemplate(newPermissionTemplateGroup(CODEVIEWER, template.getId(), group1.getId()));
      addGroupToTemplate(newPermissionTemplateGroup(ADMIN, template.getId(), group1.getId()));

      OrganizationDto otherOrganization = db.organizations().insert();
      db.users().insertGroup(otherOrganization, groupName);

      loginAsAdmin(db.getDefaultOrganization());

      newRequest()
        .setMediaType(PROTOBUF)
        .setParam(PARAM_TEMPLATE_ID, template.getUuid())
        .setParam(TEXT_QUERY, "-nam")
        .execute();
    }

    @Test
    public void return_all_permissions_of_matching_groups() {
      PermissionTemplateDto template = addTemplateToDefaultOrganization();

      GroupDto group1 = db.users().insertGroup(db.getDefaultOrganization(), "group-1-name");
      addGroupToTemplate(newPermissionTemplateGroup(CODEVIEWER, template.getId(), group1.getId()));
      addGroupToTemplate(newPermissionTemplateGroup(ADMIN, template.getId(), group1.getId()));

      GroupDto group2 = db.users().insertGroup(db.getDefaultOrganization(), "group-2-name");
      addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), group2.getId()));
      addGroupToTemplate(newPermissionTemplateGroup(ADMIN, template.getId(), group2.getId()));
```

```
  GroupDto group3 = db.users().insertGroup(db.getDefaultOrganization(), "group-3-name");

  // Anyone
  addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), null));
  addGroupToTemplate(newPermissionTemplateGroup(ISSUE_ADMIN, template.getId(), null));

  PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
  addGroupToTemplate(newPermissionTemplateGroup(ADMIN, anotherTemplate.getId(), group3.getId()));
  commit();
  loginAsAdmin(db.getDefaultOrganization());

  WsGroupsResponse response = newRequest()
    .setParam(PARAM_TEMPLATE_ID, template.getUuid())
    .executeProtobuf(WsGroupsResponse.class);

  assertThat(response.getGroupsList()).extracting("name").containsExactly("Anyone", "group-1-name", "group-2-
name");
  assertThat(response.getGroups(0).getPermissionsList()).containsOnly("user", "issueadmin");
  assertThat(response.getGroups(1).getPermissionsList()).containsOnly("codeviewer", "admin");
  assertThat(response.getGroups(2).getPermissionsList()).containsOnly("user", "admin");
 }

 @Test
 public void search_by_permission() {
  PermissionTemplateDto template = addTemplateToDefaultOrganization();

  GroupDto group1 = db.users().insertGroup(db.getDefaultOrganization(), "group-1-name");
  addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), group1.getId()));
  addGroupToTemplate(newPermissionTemplateGroup(CODEVIEWER, template.getId(), group1.getId()));

  GroupDto group2 = db.users().insertGroup(db.getDefaultOrganization(), "group-2-name");
  addGroupToTemplate(newPermissionTemplateGroup(ADMIN, template.getId(), group2.getId()));

  GroupDto group3 = db.users().insertGroup(db.getDefaultOrganization(), "group-3-name");

  // Anyone
  addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), null));

  PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
  addGroupToTemplate(newPermissionTemplateGroup(ADMIN, anotherTemplate.getId(), group3.getId()));
  commit();
  loginAsAdmin(db.getDefaultOrganization());

  WsGroupsResponse response = newRequest()
    .setParam(PARAM_PERMISSION, USER)
    .setParam(PARAM_TEMPLATE_ID, template.getUuid())
    .executeProtobuf(WsGroupsResponse.class);
```

```java
    assertThat(response.getGroupsList()).extracting("name").containsExactly("Anyone", "group-1-name");
    assertThat(response.getGroups(0).getPermissionsList()).containsOnly("user");
    assertThat(response.getGroups(1).getPermissionsList()).containsOnly("user", "codeviewer");
  }

  @Test
  public void search_by_template_name() {
    OrganizationDto defaultOrg = db.getDefaultOrganization();
    GroupDto group1 = db.users().insertGroup(defaultOrg, "group-1-name");
    GroupDto group2 = db.users().insertGroup(defaultOrg, "group-2-name");
    GroupDto group3 = db.users().insertGroup(defaultOrg, "group-3-name");

    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), group1.getId()));
    addGroupToTemplate(newPermissionTemplateGroup(ADMIN, template.getId(), group2.getId()));
    addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), null));

    PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
    addGroupToTemplate(newPermissionTemplateGroup(USER, anotherTemplate.getId(), group1.getId()));
    commit();
    loginAsAdmin(db.getDefaultOrganization());

    WsGroupsResponse response = newRequest()
      .setParam(PARAM_TEMPLATE_NAME, template.getName())
      .executeProtobuf(WsGroupsResponse.class);

    assertThat(response.getGroupsList()).extracting("name").containsExactly("Anyone", "group-1-name", "group-2-name");
  }

  @Test
  public void search_with_pagination() {
    OrganizationDto defaultOrg = db.getDefaultOrganization();
    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    GroupDto group1 = db.users().insertGroup(defaultOrg, "group-1-name");
    addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), group1.getId()));
    GroupDto group2 = db.users().insertGroup(defaultOrg, "group-2-name");
    addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), group2.getId()));
    commit();
    loginAsAdmin(db.getDefaultOrganization());

    WsGroupsResponse response = newRequest()
      .setParam(PARAM_PERMISSION, USER)
      .setParam(PARAM_TEMPLATE_NAME, template.getName())
      .setParam(PAGE, "2")
      .setParam(PAGE_SIZE, "1")
      .executeProtobuf(WsGroupsResponse.class);
```

```java
    assertThat(response.getGroupsList()).extracting("name").containsExactly("group-2-name");
  }

  @Test
  public void search_with_text_query() {
    OrganizationDto defaultOrg = db.getDefaultOrganization();
    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    GroupDto group1 = db.users().insertGroup(defaultOrg, "group-1-name");
    addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), group1.getId()));
    GroupDto group2 = db.users().insertGroup(defaultOrg, "group-2-name");
    GroupDto group3 = db.users().insertGroup(defaultOrg, "group-3");
    commit();
    loginAsAdmin(db.getDefaultOrganization());

    WsGroupsResponse response = newRequest()
      .setParam(PARAM_TEMPLATE_NAME, template.getName())
      .setParam(TEXT_QUERY, "-nam")
      .executeProtobuf(WsGroupsResponse.class);

    assertThat(response.getGroupsList()).extracting("name").containsExactly("group-1-name", "group-2-name");
  }

  @Test
  public void search_with_text_query_return_all_groups_even_when_no_permission_set() {
    OrganizationDto defaultOrg = db.getDefaultOrganization();
    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    db.users().insertGroup(defaultOrg, "group-1-name");
    db.users().insertGroup(defaultOrg, "group-2-name");
    db.users().insertGroup(defaultOrg, "group-3-name");
    commit();
    loginAsAdmin(db.getDefaultOrganization());

    WsGroupsResponse response = newRequest()
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .setParam(TEXT_QUERY, "-name")
      .executeProtobuf(WsGroupsResponse.class);

    assertThat(response.getGroupsList()).extracting("name").containsExactly("group-1-name", "group-2-name",
"group-3-name");
    assertThat(response.getGroups(0).getPermissionsList()).isEmpty();
    assertThat(response.getGroups(1).getPermissionsList()).isEmpty();
    assertThat(response.getGroups(2).getPermissionsList()).isEmpty();
  }

  @Test
  public void search_with_text_query_return_anyone_group_even_when_no_permission_set() {
    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "group");
```

```java
    addGroupToTemplate(newPermissionTemplateGroup(USER, template.getId(), group.getId()));
    commit();
    loginAsAdmin(db.getDefaultOrganization());

    WsGroupsResponse response = newRequest()
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .setParam(TEXT_QUERY, "nyo")
      .executeProtobuf(WsGroupsResponse.class);

    assertThat(response.getGroupsList()).extracting("name").containsExactly("Anyone");
    assertThat(response.getGroups(0).getPermissionsList()).isEmpty();
  }

  @Test
  public void fail_if_not_logged_in() {
    PermissionTemplateDto template1 = addTemplateToDefaultOrganization();
    userSession.anonymous();

    expectedException.expect(UnauthorizedException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, USER)
      .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
      .execute();
  }

  @Test
  public void fail_if_insufficient_privileges() {
    PermissionTemplateDto template1 = addTemplateToDefaultOrganization();
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, USER)
      .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
      .execute();
  }

  @Test
  public void fail_if_template_uuid_and_name_provided() {
    PermissionTemplateDto template1 = addTemplateToDefaultOrganization();
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, USER)
```

```java
          .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
          .setParam(PARAM_TEMPLATE_NAME, template1.getName())
          .execute();
    }

    @Test
    public void fail_if_template_uuid_nor_name_provided() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(BadRequestException.class);

      newRequest()
        .setParam(PARAM_PERMISSION, USER)
        .execute();
    }

    @Test
    public void fail_if_template_is_not_found() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(NotFoundException.class);

      newRequest()
        .setParam(PARAM_PERMISSION, USER)
        .setParam(PARAM_TEMPLATE_ID, "unknown-uuid")
        .execute();
    }

    @Test
    public void fail_if_not_a_project_permission() {
      loginAsAdmin(db.getDefaultOrganization());
      PermissionTemplateDto template1 = addTemplateToDefaultOrganization();

      expectedException.expect(IllegalArgumentException.class);

      newRequest()
        .setParam(PARAM_PERMISSION, GlobalPermissions.QUALITY_GATE_ADMIN)
        .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
        .execute();
    }

    private GroupDto insertGroupOnDefaultOrganization(String name, String description) {
      return
db.users().insertGroup(newGroupDto().setName(name).setDescription(description).setOrganizationUuid(db.getDefa
ultOrganization().getUuid()));
    }

    private void addGroupToTemplate(PermissionTemplateGroupDto permissionTemplateGroup) {
```

```java
      db.getDbClient().permissionTemplateDao().insertGroupPermission(db.getSession(), permissionTemplateGroup);
    }

    private static PermissionTemplateGroupDto newPermissionTemplateGroup(String permission, long templateId,
    @Nullable Integer groupId) {
      return newPermissionTemplateGroupDto()
        .setPermission(permission)
        .setTemplateId(templateId)
        .setGroupId(groupId);
    }

    private void commit() {
      db.commit();
    }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import org.junit.Test;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.test.TestUtils.hasOnlyPrivateConstructors;

public class PermissionTemplateDtoToPermissionTemplateResponseTest {

  @Test
  public void only_private_constructors() {
    assertThat(hasOnlyPrivateConstructors(PermissionTemplateDtoToPermissionTemplateResponse.class)).isTrue();
  }
}
```

```
package org.sonar.server.permission.ws.template;

import javax.annotation.Nullable;
import org.junit.Test;
import org.sonar.api.server.ws.WebService;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.permission.template.PermissionTemplateUserDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.issue.ws.AvatarResolverImpl;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;
import org.sonarqube.ws.Permissions;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.db.permission.OrganizationPermission.SCAN;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateUserDto;
import static org.sonar.db.user.UserTesting.newUserDto;
import static org.sonar.test.JsonAssert.assertJson;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
```

```java
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class TemplateUsersActionTest extends BasePermissionWsTest<TemplateUsersAction> {

  @Override
  protected TemplateUsersAction buildWsAction() {
    return new TemplateUsersAction(db.getDbClient(), userSession, newPermissionWsSupport(), new
AvatarResolverImpl());
  }

  @Test
  public void search_for_users_with_response_example() {
    UserDto user1 =
insertUser(newUserDto().setLogin("admin").setName("Administrator").setEmail("admin@admin.com"));
    UserDto user2 = insertUser(newUserDto().setLogin("george.orwell").setName("George
Orwell").setEmail("george.orwell@1984.net"));

    PermissionTemplateDto template1 = addTemplateToDefaultOrganization();
    addUserToTemplate(newPermissionTemplateUser(CODEVIEWER, template1, user1));
    addUserToTemplate(newPermissionTemplateUser(CODEVIEWER, template1, user2));
    addUserToTemplate(newPermissionTemplateUser(ADMIN, template1, user2));
    loginAsAdmin(db.getDefaultOrganization());

    String result = newRequest(null, template1.getUuid()).execute().getInput();
    assertJson(result).isSimilarTo(getClass().getResource("template_users-example.json"));
  }

  @Test
  public void search_for_users_by_template_name() {
    loginAsAdmin(db.getDefaultOrganization());

    UserDto user1 = insertUser(newUserDto().setLogin("login-1").setName("name-1").setEmail("email-1"));
    UserDto user2 = insertUser(newUserDto().setLogin("login-2").setName("name-2").setEmail("email-2"));
    UserDto user3 = insertUser(newUserDto().setLogin("login-3").setName("name-3").setEmail("email-3"));

    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    addUserToTemplate(newPermissionTemplateUser(USER, template, user1));
    addUserToTemplate(newPermissionTemplateUser(USER, template, user2));
    addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user1));
    addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user3));

    PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
    addUserToTemplate(newPermissionTemplateUser(USER, anotherTemplate, user1));

    Permissions.UsersWsResponse response = newRequest(null, null)
      .setParam(PARAM_TEMPLATE_NAME, template.getName())
      .executeProtobuf(Permissions.UsersWsResponse.class);
```

```java
  assertThat(response.getUsersList()).extracting("login").containsExactly("login-1", "login-2", "login-3");
  assertThat(response.getUsers(0).getPermissionsList()).containsOnly("issueadmin", "user");
  assertThat(response.getUsers(1).getPermissionsList()).containsOnly("user");
  assertThat(response.getUsers(2).getPermissionsList()).containsOnly("issueadmin");
}

@Test
public void search_using_text_query() {
  loginAsAdmin(db.getDefaultOrganization());

  UserDto user1 = insertUser(newUserDto().setLogin("login-1").setName("name-1").setEmail("email-1"));
  UserDto user2 = insertUser(newUserDto().setLogin("login-2").setName("name-2").setEmail("email-2"));
  UserDto user3 = insertUser(newUserDto().setLogin("login-3").setName("name-3").setEmail("email-3"));

  PermissionTemplateDto template = addTemplateToDefaultOrganization();
  addUserToTemplate(newPermissionTemplateUser(USER, template, user1));
  addUserToTemplate(newPermissionTemplateUser(USER, template, user2));
  addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user1));
  addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user3));

  PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
  addUserToTemplate(newPermissionTemplateUser(USER, anotherTemplate, user1));

  Permissions.UsersWsResponse response = newRequest(null, null)
    .setParam(PARAM_TEMPLATE_NAME, template.getName())
    .setParam(WebService.Param.TEXT_QUERY, "ame-1")
    .executeProtobuf(Permissions.UsersWsResponse.class);

  assertThat(response.getUsersList()).extracting("login").containsOnly("login-1");
}

@Test
public void search_using_permission() {
  UserDto user1 = insertUser(newUserDto().setLogin("login-1").setName("name-1").setEmail("email-1"));
  UserDto user2 = insertUser(newUserDto().setLogin("login-2").setName("name-2").setEmail("email-2"));
  UserDto user3 = insertUser(newUserDto().setLogin("login-3").setName("name-3").setEmail("email-3"));

  PermissionTemplateDto template = addTemplateToDefaultOrganization();
  addUserToTemplate(newPermissionTemplateUser(USER, template, user1));
  addUserToTemplate(newPermissionTemplateUser(USER, template, user2));
  addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user1));
  addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user3));

  PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
  addUserToTemplate(newPermissionTemplateUser(USER, anotherTemplate, user1));

  loginAsAdmin(db.getDefaultOrganization());
  Permissions.UsersWsResponse response = newRequest(USER, template.getUuid())
```

```java
      .executeProtobuf(Permissions.UsersWsResponse.class);
  assertThat(response.getUsersList()).extracting("login").containsExactly("login-1", "login-2");
  assertThat(response.getUsers(0).getPermissionsList()).containsOnly("issueadmin", "user");
  assertThat(response.getUsers(1).getPermissionsList()).containsOnly("user");
}

@Test
public void search_with_pagination() {
  UserDto user1 = insertUser(newUserDto().setLogin("login-1").setName("name-1").setEmail("email-1"));
  UserDto user2 = insertUser(newUserDto().setLogin("login-2").setName("name-2").setEmail("email-2"));
  UserDto user3 = insertUser(newUserDto().setLogin("login-3").setName("name-3").setEmail("email-3"));

  PermissionTemplateDto template = addTemplateToDefaultOrganization();
  addUserToTemplate(newPermissionTemplateUser(USER, template, user1));
  addUserToTemplate(newPermissionTemplateUser(USER, template, user2));
  addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user1));
  addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user3));

  PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
  addUserToTemplate(newPermissionTemplateUser(USER, anotherTemplate, user1));

  loginAsAdmin(db.getDefaultOrganization());
  Permissions.UsersWsResponse response = newRequest(USER, null)
    .setParam(PARAM_TEMPLATE_NAME, template.getName())
    .setParam(WebService.Param.SELECTED, "all")
    .setParam(WebService.Param.PAGE, "2")
    .setParam(WebService.Param.PAGE_SIZE, "1")
    .executeProtobuf(Permissions.UsersWsResponse.class);

  assertThat(response.getUsersList()).extracting("login").containsOnly("login-2");
}

@Test
public void users_are_sorted_by_name() {
  UserDto user1 = insertUser(newUserDto().setLogin("login-2").setName("name-2"));
  UserDto user2 = insertUser(newUserDto().setLogin("login-3").setName("name-3"));
  UserDto user3 = insertUser(newUserDto().setLogin("login-1").setName("name-1"));

  PermissionTemplateDto template = addTemplateToDefaultOrganization();
  addUserToTemplate(newPermissionTemplateUser(USER, template, user1));
  addUserToTemplate(newPermissionTemplateUser(USER, template, user2));
  addUserToTemplate(newPermissionTemplateUser(ISSUE_ADMIN, template, user3));

  loginAsAdmin(db.getDefaultOrganization());
  Permissions.UsersWsResponse response = newRequest(null, null)
    .setParam(PARAM_TEMPLATE_NAME, template.getName())
    .executeProtobuf(Permissions.UsersWsResponse.class);
```

```java
    assertThat(response.getUsersList()).extracting("login").containsExactly("login-1", "login-2", "login-3");
  }

  @Test
  public void empty_result_when_no_user_on_template() {
    UserDto user = insertUser(newUserDto().setLogin("login-1").setName("name-1").setEmail("email-1"));
    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    PermissionTemplateDto anotherTemplate = addTemplateToDefaultOrganization();
    addUserToTemplate(newPermissionTemplateUser(USER, anotherTemplate, user));

    loginAsAdmin(db.getDefaultOrganization());
    Permissions.UsersWsResponse response = newRequest(null, null)
      .setParam(PARAM_TEMPLATE_NAME, template.getName())
      .executeProtobuf(Permissions.UsersWsResponse.class);

    assertThat(response.getUsersList()).isEmpty();
  }

  @Test
  public void fail_if_not_a_project_permission() {
    PermissionTemplateDto template = addTemplateToDefaultOrganization();
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(GlobalPermissions.PROVISIONING, template.getUuid())
      .execute();
  }

  @Test
  public void fail_if_no_template_param() {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    newRequest(null, null)
      .execute();
  }

  @Test
  public void fail_if_template_does_not_exist() {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);

    newRequest(null, "unknown-template-uuid")
      .execute();
  }
```

```java
@Test
public void fail_if_template_uuid_and_name_provided() {
  PermissionTemplateDto template = addTemplateToDefaultOrganization();
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);

  newRequest(null, template.getUuid())
    .setParam(PARAM_TEMPLATE_NAME, template.getName())
    .execute();
}

@Test
public void fail_if_not_logged_in() {
  PermissionTemplateDto template = addTemplateToDefaultOrganization();
  userSession.anonymous();

  expectedException.expect(UnauthorizedException.class);

  newRequest(null, template.getUuid()).execute();
}

@Test
public void fail_if_insufficient_privileges() {
  PermissionTemplateDto template = addTemplateToDefaultOrganization();
  userSession.logIn().addPermission(SCAN, db.getDefaultOrganization());

  expectedException.expect(ForbiddenException.class);

  newRequest(null, template.getUuid()).execute();
}

private UserDto insertUser(UserDto userDto) {
  db.users().insertUser(userDto);
  db.organizations().addMember(db.getDefaultOrganization(), userDto);
  return userDto;
}

private void addUserToTemplate(PermissionTemplateUserDto dto) {
  db.getDbClient().permissionTemplateDao().insertUserPermission(db.getSession(), dto.getTemplateId(),
dto.getUserId(), dto.getPermission());
  db.commit();
}

private static PermissionTemplateUserDto newPermissionTemplateUser(String permission,
PermissionTemplateDto template, UserDto user) {
  return newPermissionTemplateUserDto()
```

```
        .setPermission(permission)
        .setTemplateId(template.getId())
        .setUserId(user.getId());
    }

    private TestRequest newRequest(@Nullable String permission, @Nullable String templateUuid) {
      TestRequest request = newRequest();
      if (permission != null) {
        request.setParam(PARAM_PERMISSION, permission);
      }
      if (templateUuid != null) {
        request.setParam(PARAM_TEMPLATE_ID, templateUuid);
      }
      return request;
    }


}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import javax.annotation.Nullable;
import org.junit.Test;
import org.sonar.api.resources.Qualifiers;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.organization.DefaultTemplates;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.permission.template.PermissionTemplateTesting;
import org.sonar.server.exceptions.BadRequestException;
```

```java
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.i18n.I18nRule;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.resources.Qualifiers.APP;
import static org.sonar.api.resources.Qualifiers.PROJECT;
import static org.sonar.api.resources.Qualifiers.VIEW;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_QUALIFIER;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class SetDefaultTemplateActionTest extends BasePermissionWsTest<SetDefaultTemplateAction> {

  private DbClient dbClient = db.getDbClient();
  private I18nRule i18n = new I18nRule();

  @Override
  protected SetDefaultTemplateAction buildWsAction() {
    return new SetDefaultTemplateAction(db.getDbClient(), newPermissionWsSupport(), newRootResourceTypes(),
userSession, i18n);
  }

  @Test
  public void update_project_default_template() throws Exception {
    PermissionTemplateDto viewDefaultTemplate =
db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
    db.organizations().setDefaultTemplates(db.permissionTemplates().insertTemplate(db.getDefaultOrganization()),
      viewDefaultTemplate);
    PermissionTemplateDto template = insertTemplate(db.getDefaultOrganization());
    loginAsAdmin(db.getDefaultOrganization());

    newRequest(template.getUuid(), Qualifiers.PROJECT);

    assertDefaultTemplates(db.getDefaultOrganization(), template.getUuid(), viewDefaultTemplate.getUuid());
  }

  @Test
  public void update_project_default_template_without_qualifier_param() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    db.organizations().setDefaultTemplates(organization, "any-project-template-uuid", "any-view-template-uuid");
    PermissionTemplateDto template = insertTemplate(organization);
    loginAsAdmin(organization);
```

```java
    // default value is project qualifier's value
    newRequest(template.getUuid(), null);

    assertDefaultTemplates(organization, template.getUuid(), "any-view-template-uuid");
  }

  @Test
  public void update_project_default_template_by_template_name() {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto viewDefaultTemplate = db.permissionTemplates().insertTemplate(organization);
    db.organizations().setDefaultTemplates(db.permissionTemplates().insertTemplate(organization),
        viewDefaultTemplate);
    PermissionTemplateDto template = insertTemplate(organization);
    loginAsAdmin(organization);

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_TEMPLATE_NAME, template.getName().toUpperCase())
      .execute();
    db.getSession().commit();

    assertDefaultTemplates(organization, template.getUuid(), viewDefaultTemplate.getUuid());
  }

  @Test
  public void update_view_default_template() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto projectDefaultTemplate = db.permissionTemplates().insertTemplate(organization);
    db.organizations().setDefaultTemplates(projectDefaultTemplate, null);
    PermissionTemplateDto template = insertTemplate(organization);
    loginAsAdmin(organization);

    newRequest(template.getUuid(), VIEW);

    assertDefaultTemplates(organization, projectDefaultTemplate.getUuid(), template.getUuid());
  }

  @Test
  public void fail_if_update_default_template_with_app_qualifier() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto projectDefaultTemplate = db.permissionTemplates().insertTemplate(organization);
    db.organizations().setDefaultTemplates(projectDefaultTemplate, null);
    PermissionTemplateDto template = insertTemplate(organization);
    loginAsAdmin(organization);

    expectedException.expect(IllegalArgumentException.class);
    expectedException.expectMessage("Value of parameter 'qualifier' (APP) must be one of: [TRK, VW]");
```

```java
    newRequest(template.getUuid(), APP);
  }

  @Test
  public void fail_if_anonymous() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto template = insertTemplate(organization);
    userSession.anonymous();

    expectedException.expect(UnauthorizedException.class);

    newRequest(template.getUuid(), PROJECT);
  }

  @Test
  public void fail_if_not_admin() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto template = insertTemplate(organization);
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest(template.getUuid(), null);
  }

  @Test
  public void fail_if_template_not_provided() throws Exception {
    expectedException.expect(BadRequestException.class);

    newRequest(null, PROJECT);
  }

  @Test
  public void fail_if_template_does_not_exist() throws Exception {
    expectedException.expect(NotFoundException.class);

    newRequest("unknown-template-uuid", PROJECT);
  }

  @Test
  public void fail_if_qualifier_is_not_root() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto template = insertTemplate(organization);
    loginAsAdmin(organization);

    expectedException.expect(IllegalArgumentException.class);
    expectedException.expectMessage("Value of parameter 'qualifier' (FIL) must be one of: [TRK, VW]");
```

```java
    newRequest(template.getUuid(), Qualifiers.FILE);
  }

  @Test
  public void fail_if_organization_has_no_default_templates() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto template = insertTemplate(organization);
    loginAsAdmin(organization);

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("No Default templates for organization with uuid '" + organization.getUuid() +
"'");

    newRequest(template.getUuid(), null);
  }

  private String newRequest(@Nullable String templateUuid, @Nullable String qualifier) {
    TestRequest request = newRequest();
    if (templateUuid != null) {
      request.setParam(PARAM_TEMPLATE_ID, templateUuid);
    }
    if (qualifier != null) {
      request.setParam(PARAM_QUALIFIER, qualifier);
    }

    return request.execute().getInput();
  }

  private PermissionTemplateDto insertTemplate(OrganizationDto organization) {
    PermissionTemplateDto res = dbClient.permissionTemplateDao().insert(db.getSession(),
PermissionTemplateTesting.newPermissionTemplateDto()
      .setOrganizationUuid(organization.getUuid())
      .setUuid("permission-template-uuid"));
    db.commit();
    return res;
  }

  private void assertDefaultTemplates(OrganizationDto organizationDto,
    @Nullable String projectDefaultTemplateUuid, @Nullable String viewDefaultTemplateUuid) {
    DbSession dbSession = db.getSession();
    DefaultTemplates defaultTemplates = db.getDbClient().organizationDao().getDefaultTemplates(dbSession,
organizationDto.getUuid())
      .orElseThrow(() -> new IllegalStateException("No default templates for organization with uuid '" +
organizationDto.getUuid() + "'"));

    assertThat(defaultTemplates.getProjectUuid()).isEqualTo(projectDefaultTemplateUuid);
    assertThat(defaultTemplates.getViewUuid()).isEqualTo(viewDefaultTemplateUuid);
  }
```

```
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import java.util.List;
import javax.annotation.Nullable;
import org.junit.Before;
import org.junit.Test;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.GroupDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.security.DefaultGroups.ANYONE;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.db.permission.OrganizationPermission.SCAN;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class RemoveGroupFromTemplateActionTest extends
```

```
BasePermissionWsTest<RemoveGroupFromTemplateAction> {

  private static final String PERMISSION = CODEVIEWER;

  private GroupDto group;
  private PermissionTemplateDto template;

  @Override
  protected RemoveGroupFromTemplateAction buildWsAction() {
    return new RemoveGroupFromTemplateAction(db.getDbClient(), newPermissionWsSupport(), userSession);
  }

  @Before
  public void setUp() {
    loginAsAdmin(db.getDefaultOrganization());

    group = db.users().insertGroup(db.getDefaultOrganization(), "group-name");
    template = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
    addGroupToTemplate(template, group.getId(), PERMISSION);
  }

  @Test
  public void remove_group_from_template() throws Exception {
    newRequest(group.getName(), template.getUuid(), PERMISSION);

    assertThat(getGroupNamesInTemplateAndPermission(template, PERMISSION)).isEmpty();
  }

  @Test
  public void remove_group_from_template_by_name_case_insensitive() {
    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, PERMISSION)
      .setParam(PARAM_TEMPLATE_NAME, template.getName().toUpperCase())
      .execute();

    assertThat(getGroupNamesInTemplateAndPermission(template, PERMISSION)).isEmpty();
  }

  @Test
  public void remove_group_with_group_id() {
    newRequest()
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .setParam(PARAM_PERMISSION, PERMISSION)
      .setParam(PARAM_GROUP_ID, String.valueOf(group.getId()))
      .execute();

    assertThat(getGroupNamesInTemplateAndPermission(template, PERMISSION)).isEmpty();
```

```java
  }

  @Test
  public void remove_group_twice_without_error() throws Exception {
    newRequest(group.getName(), template.getUuid(), PERMISSION);
    newRequest(group.getName(), template.getUuid(), PERMISSION);

    assertThat(getGroupNamesInTemplateAndPermission(template, PERMISSION)).isEmpty();
  }

  @Test
  public void remove_anyone_group_from_template() throws Exception {
    addGroupToTemplate(template, null, PERMISSION);

    newRequest(ANYONE, template.getUuid(), PERMISSION);

    assertThat(getGroupNamesInTemplateAndPermission(template,
PERMISSION)).containsExactly(group.getName());
  }

  @Test
  public void fail_if_not_a_project_permission() throws Exception {
    expectedException.expect(IllegalArgumentException.class);

    newRequest(group.getName(), template.getUuid(), GlobalPermissions.PROVISIONING);
  }

  @Test
  public void fail_if_insufficient_privileges() throws Exception {
    userSession.logIn().addPermission(SCAN, db.getDefaultOrganization());

    expectedException.expect(ForbiddenException.class);

    newRequest(group.getName(), template.getUuid(), PERMISSION);
  }

  @Test
  public void fail_if_not_logged_in() throws Exception {
    expectedException.expect(UnauthorizedException.class);
    userSession.anonymous();

    newRequest(group.getName(), template.getUuid(), PERMISSION);
  }

  @Test
  public void fail_if_group_params_missing() throws Exception {
    expectedException.expect(BadRequestException.class);
```

```
    newRequest(null, template.getUuid(), PERMISSION);
  }

  @Test
  public void fail_if_permission_missing() throws Exception {
    expectedException.expect(IllegalArgumentException.class);

    newRequest(group.getName(), template.getUuid(), null);
  }

  @Test
  public void fail_if_template_missing() throws Exception {
    expectedException.expect(BadRequestException.class);

    newRequest(group.getName(), null, PERMISSION);
  }

  @Test
  public void fail_if_group_does_not_exist() throws Exception {
    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("No group with name 'unknown-group-name'");

    newRequest("unknown-group-name", template.getUuid(), PERMISSION);
  }

  @Test
  public void fail_if_template_key_does_not_exist() throws Exception {
    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("Permission template with id 'unknown-key' is not found");

    newRequest(group.getName(), "unknown-key", PERMISSION);
  }

  private void newRequest(@Nullable String groupName, @Nullable String templateKey, @Nullable String
permission) {
    TestRequest request = newRequest();
    if (groupName != null) {
      request.setParam(PARAM_GROUP_NAME, groupName);
    }
    if (templateKey != null) {
      request.setParam(PARAM_TEMPLATE_ID, templateKey);
    }
    if (permission != null) {
      request.setParam(PARAM_PERMISSION, permission);
    }

    request.execute();
  }
```

```java
 private void addGroupToTemplate(PermissionTemplateDto template, @Nullable Integer groupId, String
permission) {
   db.getDbClient().permissionTemplateDao().insertGroupPermission(db.getSession(), template.getId(), groupId,
permission);
   db.commit();
 }

 private List<String> getGroupNamesInTemplateAndPermission(PermissionTemplateDto template, String
permission) {
   PermissionQuery permissionQuery =
PermissionQuery.builder().setOrganizationUuid(template.getOrganizationUuid()).setPermission(permission).build()
;
   return db.getDbClient().permissionTemplateDao()
     .selectGroupNamesByQueryAndTemplate(db.getSession(), permissionQuery, template.getId());
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import javax.annotation.Nullable;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.api.utils.internal.TestSystem2;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.permission.ws.BasePermissionWsTest;
import org.sonar.server.ws.TestRequest;
import org.sonar.server.ws.TestResponse;
```

```java
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_PROFILES;
import static org.sonar.test.JsonAssert.assertJson;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_DESCRIPTION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY_PATTERN;

public class CreateTemplateActionTest extends BasePermissionWsTest<CreateTemplateAction> {

  private static final long NOW = 1_440_512_328_743L;
  private System2 system = new TestSystem2().setNow(NOW);

  @Override
  protected CreateTemplateAction buildWsAction() {
    return new CreateTemplateAction(db.getDbClient(), userSession, system, newPermissionWsSupport());
  }

  @Test
  public void create_full_permission_template() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    TestResponse result = newRequest("Finance", "Permissions for financially related projects", ".*\\.finance\\..*");

    assertJson(result.getInput())
      .ignoreFields("id")
      .isSimilarTo(getClass().getResource("create_template-example.json"));
    PermissionTemplateDto finance = selectTemplateInDefaultOrganization("Finance");
    assertThat(finance.getName()).isEqualTo("Finance");
    assertThat(finance.getDescription()).isEqualTo("Permissions for financially related projects");
    assertThat(finance.getKeyPattern()).isEqualTo(".*\\.finance\\..*");
    assertThat(finance.getUuid()).isNotEmpty();
    assertThat(finance.getCreatedAt().getTime()).isEqualTo(NOW);
    assertThat(finance.getUpdatedAt().getTime()).isEqualTo(NOW);
  }

  @Test
  public void create_minimalist_permission_template() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    newRequest("Finance", null, null);

    PermissionTemplateDto finance = selectTemplateInDefaultOrganization("Finance");
    assertThat(finance.getName()).isEqualTo("Finance");
    assertThat(finance.getDescription()).isNullOrEmpty();
    assertThat(finance.getKeyPattern()).isNullOrEmpty();
    assertThat(finance.getUuid()).isNotEmpty();
    assertThat(finance.getCreatedAt().getTime()).isEqualTo(NOW);
```

```java
    assertThat(finance.getUpdatedAt().getTime()).isEqualTo(NOW);
  }

  @Test
  public void fail_if_name_not_provided() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(IllegalArgumentException.class);

    newRequest(null, null, null);
  }

  @Test
  public void fail_if_name_empty() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("The template name must not be blank");

    newRequest("", null, null);
  }

  @Test
  public void fail_if_regexp_if_not_valid() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("The 'projectKeyPattern' parameter must be a valid Java regular expression.
'[azerty' was passed");

    newRequest("Finance", null, "[azerty");
  }

  @Test
  public void fail_if_name_already_exists_in_database_case_insensitive() throws Exception {
    loginAsAdmin(db.getDefaultOrganization());
    PermissionTemplateDto template = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("A template with the name '" + template.getName() + "' already exists (case
insensitive).");

    newRequest(template.getName(), null, null);
  }

  @Test
  public void fail_if_not_admin() throws Exception {
    userSession.logIn().addPermission(ADMINISTER_QUALITY_PROFILES, db.getDefaultOrganization());
```

```java
    expectedException.expect(ForbiddenException.class);

    newRequest("Finance", null, null);
  }

  private TestResponse newRequest(@Nullable String name, @Nullable String description, @Nullable String
projectPattern) {
    TestRequest request = newRequest();
    if (name != null) {
      request.setParam(PARAM_NAME, name);
    }
    if (description != null) {
      request.setParam(PARAM_DESCRIPTION, description);
    }
    if (projectPattern != null) {
      request.setParam(PARAM_PROJECT_KEY_PATTERN, projectPattern);
    }

    return request.execute();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import java.util.Optional;
import org.junit.Before;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
```

```java
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.permission.template.PermissionTemplateCharacteristicDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.permission.ws.BasePermissionWsTest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.when;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;

public class RemoveProjectCreatorFromTemplateActionTest extends
BasePermissionWsTest<RemoveProjectCreatorFromTemplateAction> {

  private System2 system = mock(System2.class);
  private PermissionTemplateDto template;

  @Override
  protected RemoveProjectCreatorFromTemplateAction buildWsAction() {
    return new RemoveProjectCreatorFromTemplateAction(db.getDbClient(), newPermissionWsSupport(),
userSession, system);
  }

  @Before
  public void setUp() {
    loginAsAdmin(db.getDefaultOrganization());
    when(system.now()).thenReturn(2_000_000_000L);
    template = db.permissionTemplates().insertTemplate(db.getDefaultOrganization());
  }

  @Test
  public void update_template_permission() {
    PermissionTemplateCharacteristicDto characteristic =
db.getDbClient().permissionTemplateCharacteristicDao().insert(db.getSession(),
      new PermissionTemplateCharacteristicDto()
        .setTemplateId(template.getId())
        .setPermission(UserRole.USER)
        .setWithProjectCreator(false)
        .setCreatedAt(1_000_000_000L)
        .setUpdatedAt(1_000_000_000L));
    db.commit();
    when(system.now()).thenReturn(3_000_000_000L);

    newRequest()
```

```
      .setParam(PARAM_PERMISSION, UserRole.USER)
      .setParam(PARAM_TEMPLATE_NAME, template.getName())
      .execute();

    assertWithoutProjectCreatorFor(UserRole.USER);
    PermissionTemplateCharacteristicDto reloaded = reload(characteristic);
    assertThat(reloaded.getCreatedAt()).isEqualTo(1_000_000_000L);
    assertThat(reloaded.getUpdatedAt()).isEqualTo(3_000_000_000L);
  }

  @Test
  public void do_not_fail_when_no_template_permission() {
    newRequest()
      .setParam(PARAM_PERMISSION, UserRole.ADMIN)
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .execute();

    assertNoTemplatePermissionFor(UserRole.ADMIN);
  }

  @Test
  public void fail_when_template_does_not_exist() {
    expectedException.expect(NotFoundException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, UserRole.ADMIN)
      .setParam(PARAM_TEMPLATE_ID, "42")
      .execute();
  }

  @Test
  public void fail_if_permission_is_not_a_project_permission() {
    expectedException.expect(IllegalArgumentException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, GlobalPermissions.QUALITY_GATE_ADMIN)
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .execute();
  }

  @Test
  public void fail_if_not_authenticated() {
    userSession.anonymous();

    expectedException.expect(UnauthorizedException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, UserRole.ADMIN)
```

```java
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .execute();
  }

  @Test
  public void fail_if_insufficient_privileges() {
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest()
      .setParam(PARAM_PERMISSION, UserRole.ADMIN)
      .setParam(PARAM_TEMPLATE_ID, template.getUuid())
      .execute();
  }

  private void assertWithoutProjectCreatorFor(String permission) {
    Optional<PermissionTemplateCharacteristicDto> templatePermission =
db.getDbClient().permissionTemplateCharacteristicDao().selectByPermissionAndTemplateId(db.getSession(),
      permission, template.getId());
    assertThat(templatePermission).isPresent();
    assertThat(templatePermission.get().getWithProjectCreator()).isFalse();
  }

  private void assertNoTemplatePermissionFor(String permission) {
    Optional<PermissionTemplateCharacteristicDto> templatePermission =
db.getDbClient().permissionTemplateCharacteristicDao().selectByPermissionAndTemplateId(db.getSession(),
      permission, template.getId());
    assertThat(templatePermission).isNotPresent();
  }

  private PermissionTemplateCharacteristicDto reload(PermissionTemplateCharacteristicDto characteristic) {
    return
db.getDbClient().permissionTemplateCharacteristicDao().selectByPermissionAndTemplateId(db.getSession(),
characteristic.getPermission(), characteristic.getTemplateId())
      .get();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
```

```
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;

import org.junit.rules.ExternalResource;
import org.sonar.api.resources.ResourceType;
import org.sonar.api.resources.ResourceTypeTree;
import org.sonar.api.resources.ResourceTypes;
import org.sonar.db.organization.DefaultTemplates;

import static org.sonar.api.resources.Qualifiers.APP;
import static org.sonar.api.resources.Qualifiers.PROJECT;
import static org.sonar.api.resources.Qualifiers.VIEW;

public class DefaultTemplatesResolverRule extends ExternalResource implements DefaultTemplatesResolver {
 private static final DefaultTemplatesResolver WITH_GOV = new DefaultTemplatesResolverImpl(
   new ResourceTypes(new ResourceTypeTree[] {
    ResourceTypeTree.builder()
     .addType(ResourceType.builder(PROJECT).build())
     .build(),
    ResourceTypeTree.builder()
     .addType(ResourceType.builder(VIEW).build())
     .build(),
    ResourceTypeTree.builder()
     .addType(ResourceType.builder(APP).build())
     .build()
   }));
 private static final DefaultTemplatesResolver WITHOUT_GOV = new DefaultTemplatesResolverImpl(
   new ResourceTypes(new ResourceTypeTree[] {ResourceTypeTree.builder()
    .addType(ResourceType.builder(PROJECT).build())
    .build()}));

 private final boolean governanceInitiallyInstalled;
 private boolean governanceInstalled;

 private DefaultTemplatesResolverRule(boolean governanceInitiallyInstalled) {
  this.governanceInitiallyInstalled = governanceInitiallyInstalled;
  this.governanceInstalled = governanceInitiallyInstalled;
 }

 @Override
```

```java
  protected void before() {
    this.governanceInstalled = governanceInitiallyInstalled;
  }

  public void installGovernance() {
    this.governanceInstalled = true;
  }

  public void uninstallGovernance() {
    this.governanceInstalled = false;
  }

  public static DefaultTemplatesResolverRule withoutGovernance() {
    return new DefaultTemplatesResolverRule(false);
  }

  public static DefaultTemplatesResolverRule withGovernance() {
    return new DefaultTemplatesResolverRule(true);
  }

  @Override
  public DefaultTemplatesResolverImpl.ResolvedDefaultTemplates resolve(DefaultTemplates defaultTemplates) {
    if (governanceInstalled) {
      return WITH_GOV.resolve(defaultTemplates);
    }
    return WITHOUT_GOV.resolve(defaultTemplates);
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws.template;
```

```java
import java.util.Collections;
import java.util.List;
import org.apache.commons.lang.StringUtils;
import org.junit.Before;
import org.junit.Test;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.api.web.UserRole;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.es.ProjectIndexers;
import org.sonar.server.es.TestProjectIndexers;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.i18n.I18nRule;
import org.sonar.server.permission.PermissionTemplateService;
import org.sonar.server.permission.ws.BasePermissionWsTest;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.utils.DateUtils.parseDate;
import static org.sonar.db.component.ComponentTesting.newApplication;
import static org.sonar.db.component.ComponentTesting.newView;
import static org.sonar.db.component.SnapshotTesting.newAnalysis;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_TEMPLATE_NAME;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_ANALYZED_BEFORE;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_ON_PROVISIONED_ONLY;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_PROJECTS;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_QUALIFIERS;
import static org.sonarqube.ws.client.project.ProjectsWsParameters.PARAM_VISIBILITY;

public class BulkApplyTemplateActionTest extends BasePermissionWsTest<BulkApplyTemplateAction> {

  @org.junit.Rule
  public DefaultTemplatesResolverRule defaultTemplatesResolver =
  DefaultTemplatesResolverRule.withoutGovernance();

  private UserDto user1;
  private UserDto user2;
  private GroupDto group1;
  private GroupDto group2;
```

```java
private OrganizationDto organization;
private PermissionTemplateDto template1;
private PermissionTemplateDto template2;
private ProjectIndexers projectIndexers = new TestProjectIndexers();

@Override
protected BulkApplyTemplateAction buildWsAction() {
  PermissionTemplateService permissionTemplateService = new PermissionTemplateService(db.getDbClient(),
    projectIndexers, userSession, defaultTemplatesResolver);
  return new BulkApplyTemplateAction(db.getDbClient(), userSession, permissionTemplateService,
newPermissionWsSupport(), new I18nRule(), newRootResourceTypes());
}

@Before
public void setUp() {
  organization = db.organizations().insert();

  user1 = db.users().insertUser();
  user2 = db.users().insertUser();
  group1 = db.users().insertGroup(organization);
  group2 = db.users().insertGroup(organization);

  db.organizations().addMember(organization, user1);
  db.organizations().addMember(organization, user2);

  // template 1 for org 1
  template1 = db.permissionTemplates().insertTemplate(organization);
  addUserToTemplate(user1, template1, UserRole.CODEVIEWER);
  addUserToTemplate(user2, template1, UserRole.ISSUE_ADMIN);
  addGroupToTemplate(group1, template1, UserRole.ADMIN);
  addGroupToTemplate(group2, template1, UserRole.USER);
  // template 2
  template2 = db.permissionTemplates().insertTemplate(organization);
  addUserToTemplate(user1, template2, UserRole.USER);
  addUserToTemplate(user2, template2, UserRole.USER);
  addGroupToTemplate(group1, template2, UserRole.USER);
  addGroupToTemplate(group2, template2, UserRole.USER);
}

@Test
public void bulk_apply_template_by_template_uuid() {
  // this project should not be applied the template
  OrganizationDto otherOrganization = db.organizations().insert();
  db.components().insertPrivateProject(otherOrganization);

  ComponentDto privateProject = db.components().insertPrivateProject(organization);
  ComponentDto publicProject = db.components().insertPublicProject(organization);
  loginAsAdmin(organization);
```

```java
    newRequest()
      .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
      .execute();

    assertTemplate1AppliedToPrivateProject(privateProject);
    assertTemplate1AppliedToPublicProject(publicProject);
  }

  @Test
  public void
request_throws_NotFoundException_if_template_with_specified_name_does_not_exist_in_specified_organization(
) {
    OrganizationDto otherOrganization = db.organizations().insert();
    loginAsAdmin(otherOrganization);

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage("Permission template with name '" + template1.getName()
      + "' is not found (case insensitive) in organization with key '" + otherOrganization.getKey() + "'");

    newRequest()
      .setParam(PARAM_ORGANIZATION, otherOrganization.getKey())
      .setParam(PARAM_TEMPLATE_NAME, template1.getName())
      .execute();
  }

  @Test
  public void request_throws_IAE_if_more_than_1000_projects() {
    expectedException.expect(IllegalArgumentException.class);
    expectedException.expectMessage("'projects' can contains only 1000 values, got 1001");

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_TEMPLATE_NAME, template1.getName())
      .setParam(PARAM_PROJECTS, StringUtils.join(Collections.nCopies(1_001, "foo"), ","))
      .execute();
  }

  @Test
  public void bulk_apply_template_by_template_name() {
    ComponentDto privateProject = db.components().insertPrivateProject(organization);
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    loginAsAdmin(organization);

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_TEMPLATE_NAME, template1.getName())
      .execute();
```

```java
    assertTemplate1AppliedToPrivateProject(privateProject);
    assertTemplate1AppliedToPublicProject(publicProject);
  }

  @Test
  public void apply_template_by_qualifiers() {
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    ComponentDto privateProject = db.components().insertPrivateProject(organization);
    ComponentDto view = db.components().insertComponent(newView(organization));
    ComponentDto application = db.components().insertComponent(newApplication(organization));
    loginAsAdmin(organization);

    newRequest()
      .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
      .setParam(PARAM_QUALIFIERS, String.join(",", Qualifiers.PROJECT, Qualifiers.APP))
      .execute();

    assertTemplate1AppliedToPrivateProject(privateProject);
    assertTemplate1AppliedToPublicProject(publicProject);
    assertTemplate1AppliedToPublicProject(application);
    assertNoPermissionOnProject(view);
  }

  @Test
  public void apply_template_by_query_on_name_and_key_public_project() {
    ComponentDto publicProjectFoundByKey =
ComponentTesting.newPublicProjectDto(organization).setDbKey("sonar");
    db.components().insertProjectAndSnapshot(publicProjectFoundByKey);
    ComponentDto publicProjectFoundByName =
ComponentTesting.newPublicProjectDto(organization).setName("name-sonar-name");
    db.components().insertProjectAndSnapshot(publicProjectFoundByName);
    ComponentDto projectUntouched = ComponentTesting.newPublicProjectDto(organization).setDbKey("new-
sona").setName("project-name");
    db.components().insertProjectAndSnapshot(projectUntouched);
    loginAsAdmin(organization);

    newRequest()
      .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
      .setParam(Param.TEXT_QUERY, "SONAR")
      .execute();

    assertTemplate1AppliedToPublicProject(publicProjectFoundByKey);
    assertTemplate1AppliedToPublicProject(publicProjectFoundByName);
    assertNoPermissionOnProject(projectUntouched);
  }

  @Test
```

```java
 public void apply_template_by_query_on_name_and_key() {
   // partial match on key
   ComponentDto privateProjectFoundByKey =
ComponentTesting.newPrivateProjectDto(organization).setDbKey("sonarqube");
   db.components().insertProjectAndSnapshot(privateProjectFoundByKey);
   ComponentDto privateProjectFoundByName =
ComponentTesting.newPrivateProjectDto(organization).setName("name-sonar-name");
   db.components().insertProjectAndSnapshot(privateProjectFoundByName);
   ComponentDto projectUntouched = ComponentTesting.newPublicProjectDto(organization).setDbKey("new-
sona").setName("project-name");
   db.components().insertProjectAndSnapshot(projectUntouched);
   loginAsAdmin(organization);

   newRequest()
    .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
    .setParam(Param.TEXT_QUERY, "SONAR")
    .execute();

   assertTemplate1AppliedToPrivateProject(privateProjectFoundByKey);
   assertTemplate1AppliedToPrivateProject(privateProjectFoundByName);
   assertNoPermissionOnProject(projectUntouched);
 }

 @Test
 public void apply_template_by_project_keys() {
   ComponentDto project1 = db.components().insertPrivateProject(organization);
   ComponentDto project2 = db.components().insertPrivateProject(organization);
   ComponentDto untouchedProject = db.components().insertPrivateProject(organization);
   loginAsAdmin(organization);

   newRequest()
    .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
    .setParam(PARAM_PROJECTS, String.join(",", project1.getKey(), project2.getKey()))
    .execute();

   assertTemplate1AppliedToPrivateProject(project1);
   assertTemplate1AppliedToPrivateProject(project2);
   assertNoPermissionOnProject(untouchedProject);
 }

 @Test
 public void apply_template_by_provisioned_only() {
   ComponentDto provisionedProject1 = db.components().insertPrivateProject(organization);
   ComponentDto provisionedProject2 = db.components().insertPrivateProject(organization);
   ComponentDto analyzedProject = db.components().insertPrivateProject(organization);
   db.components().insertSnapshot(newAnalysis(analyzedProject));
   loginAsAdmin(organization);
```

```
    newRequest()
     .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
     .setParam(PARAM_ON_PROVISIONED_ONLY, "true")
     .execute();

    assertTemplate1AppliedToPrivateProject(provisionedProject1);
    assertTemplate1AppliedToPrivateProject(provisionedProject2);
    assertNoPermissionOnProject(analyzedProject);
  }

  @Test
  public void apply_template_by_analyzed_before() {
    ComponentDto oldProject1 = db.components().insertPrivateProject(organization);
    ComponentDto oldProject2 = db.components().insertPrivateProject(organization);
    ComponentDto recentProject = db.components().insertPrivateProject(organization);
    db.components().insertSnapshot(oldProject1, a -> a.setCreatedAt(parseDate("2015-02-03").getTime()));
    db.components().insertSnapshot(oldProject2, a -> a.setCreatedAt(parseDate("2016-12-11").getTime()));
    db.components().insertSnapshot(recentProject, a -> a.setCreatedAt(System.currentTimeMillis()));
    loginAsAdmin(organization);

    newRequest()
     .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
     .setParam(PARAM_ANALYZED_BEFORE, "2017-09-07")
     .execute();

    assertTemplate1AppliedToPrivateProject(oldProject1);
    assertTemplate1AppliedToPrivateProject(oldProject2);
    assertNoPermissionOnProject(recentProject);
  }

  @Test
  public void apply_template_by_visibility() {
    ComponentDto privateProject1 = db.components().insertPrivateProject(organization);
    ComponentDto privateProject2 = db.components().insertPrivateProject(organization);
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    loginAsAdmin(organization);

    newRequest()
     .setParam(PARAM_TEMPLATE_ID, template1.getUuid())
     .setParam(PARAM_VISIBILITY, "private")
     .execute();

    assertTemplate1AppliedToPrivateProject(privateProject1);
    assertTemplate1AppliedToPrivateProject(privateProject2);
    assertNoPermissionOnProject(publicProject);
  }

  @Test
```

```java
public void fail_if_no_template_parameter() {
  loginAsAdmin(db.getDefaultOrganizaion());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Template name or template id must be provided, not both.");

  newRequest().execute();
}

@Test
public void fail_if_template_name_is_incorrect() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage("Permission template with id 'unknown-template-uuid' is not found");

  newRequest().setParam(PARAM_TEMPLATE_ID, "unknown-template-uuid").execute();
}

private void assertTemplate1AppliedToPublicProject(ComponentDto project) {
  assertThat(selectProjectPermissionGroups(project, UserRole.ADMIN)).containsExactly(group1.getName());
  assertThat(selectProjectPermissionGroups(project, UserRole.USER)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.ADMIN)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.CODEVIEWER)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.ISSUE_ADMIN)).containsExactly(user2.getId());
}

private void assertTemplate1AppliedToPrivateProject(ComponentDto project) {
  assertThat(selectProjectPermissionGroups(project, UserRole.ADMIN)).containsExactly(group1.getName());
  assertThat(selectProjectPermissionGroups(project, UserRole.USER)).containsExactly(group2.getName());
  assertThat(selectProjectPermissionUsers(project, UserRole.ADMIN)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.CODEVIEWER)).containsExactly(user1.getId());
  assertThat(selectProjectPermissionUsers(project, UserRole.ISSUE_ADMIN)).containsExactly(user2.getId());
}

private void assertNoPermissionOnProject(ComponentDto project) {
  assertThat(selectProjectPermissionGroups(project, UserRole.ADMIN)).isEmpty();
  assertThat(selectProjectPermissionGroups(project, UserRole.CODEVIEWER)).isEmpty();
  assertThat(selectProjectPermissionGroups(project, UserRole.ISSUE_ADMIN)).isEmpty();
  assertThat(selectProjectPermissionGroups(project, UserRole.USER)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.ADMIN)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.CODEVIEWER)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.ISSUE_ADMIN)).isEmpty();
  assertThat(selectProjectPermissionUsers(project, UserRole.USER)).isEmpty();
}

private void addUserToTemplate(UserDto user, PermissionTemplateDto permissionTemplate, String permission) {
  db.getDbClient().permissionTemplateDao().insertUserPermission(db.getSession(), permissionTemplate.getId(),
```

```java
    user.getId(), permission);
    db.commit();
  }

  private void addGroupToTemplate(GroupDto group, PermissionTemplateDto permissionTemplate, String
permission) {
    db.getDbClient().permissionTemplateDao().insertGroupPermission(db.getSession(), permissionTemplate.getId(),
group.getId(), permission);
    db.commit();
  }

  private List<String> selectProjectPermissionGroups(ComponentDto project, String permission) {
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(project.getOrganizationUuid()).setPermission(permission).setComp
onentUuid(project.uuid()).build();
    return db.getDbClient().groupPermissionDao().selectGroupNamesByQuery(db.getSession(), query);
  }

  private List<Integer> selectProjectPermissionUsers(ComponentDto project, String permission) {
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(project.getOrganizationUuid()).setPermission(permission).setComp
onentUuid(project.uuid()).build();
    return db.getDbClient().userPermissionDao().selectUserIdsByQuery(db.getSession(), query);
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.junit.Test;
import org.sonar.api.web.UserRole;
```

```java
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.ServerException;

import static java.lang.String.format;
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.fail;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.PROVISIONING;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.component.ComponentTesting.newDirectory;
import static org.sonar.db.component.ComponentTesting.newFileDto;
import static org.sonar.db.component.ComponentTesting.newModuleDto;
import static org.sonar.db.component.ComponentTesting.newPrivateProjectDto;
import static org.sonar.db.component.ComponentTesting.newSubView;
import static org.sonar.db.component.ComponentTesting.newView;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;

public class AddGroupActionTest extends BasePermissionWsTest<AddGroupAction> {

 private static final String A_PROJECT_UUID = "project-uuid";
 private static final String A_PROJECT_KEY = "project-key";

 @Override
 protected AddGroupAction buildWsAction() {
   return new AddGroupAction(db.getDbClient(), userSession, newPermissionUpdater(),
newPermissionWsSupport());
 }

 @Test
 public void add_permission_to_group_referenced_by_its_name() {
   GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
   loginAsAdmin(db.getDefaultOrganization());
```

```java
  newRequest()
    .setParam(PARAM_GROUP_NAME, "sonar-administrators")
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();

  assertThat(db.users().selectGroupPermissions(group, null)).containsOnly(SYSTEM_ADMIN);
}

@Test
public void reference_group_by_its_name_in_organization() {
  OrganizationDto org = db.organizations().insert();
  GroupDto group = db.users().insertGroup(org, "the-group");
  loginAsAdmin(org);

  newRequest()
    .setParam(PARAM_ORGANIZATION, org.getKey())
    .setParam(PARAM_GROUP_NAME, group.getName())
    .setParam(PARAM_PERMISSION, PROVISIONING)
    .execute();

  assertThat(db.users().selectGroupPermissions(group, null)).containsOnly(PROVISIONING);
}

@Test
public void add_permission_to_group_referenced_by_its_id() {
  GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
    .setParam(PARAM_GROUP_ID, group.getId().toString())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();

  assertThat(db.users().selectGroupPermissions(group, null)).containsOnly(SYSTEM_ADMIN);
}

@Test
public void add_permission_to_project_referenced_by_its_id() {
  GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
  ComponentDto project = db.components().insertComponent(newPrivateProjectDto(db.getDefaultOrganization(),
A_PROJECT_UUID).setDbKey(A_PROJECT_KEY));
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
    .setParam(PARAM_GROUP_NAME, group.getName())
    .setParam(PARAM_PROJECT_ID, A_PROJECT_UUID)
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
```

```java
    assertThat(db.users().selectGroupPermissions(group, null)).isEmpty();
    assertThat(db.users().selectGroupPermissions(group, project)).containsOnly(SYSTEM_ADMIN);
  }

  @Test
  public void add_permission_to_project_referenced_by_its_key() {
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    ComponentDto project = db.components().insertComponent(newPrivateProjectDto(db.getDefaultOrganization(),
A_PROJECT_UUID).setDbKey(A_PROJECT_KEY));
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PROJECT_KEY, A_PROJECT_KEY)
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();

    assertThat(db.users().selectGroupPermissions(group, null)).isEmpty();
    assertThat(db.users().selectGroupPermissions(group, project)).containsOnly(SYSTEM_ADMIN);
  }

  @Test
  public void add_with_view_uuid() {
    OrganizationDto organizationDto = db.getDefaultOrganization();
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    ComponentDto view = db.components().insertComponent(newView(organizationDto, "view-
uuid").setDbKey("view-key"));
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PROJECT_ID, view.uuid())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();

    assertThat(db.users().selectGroupPermissions(group, null)).isEmpty();
    assertThat(db.users().selectGroupPermissions(group, view)).containsOnly(SYSTEM_ADMIN);
  }

  @Test
  public void fail_if_project_uuid_is_not_found() {
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(NotFoundException.class);
    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
```

```java
      .setParam(PARAM_PROJECT_ID, "not-found")
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();
  }

  @Test
  public void fail_when_component_is_a_module() {
    ComponentDto module =
db.components().insertComponent(newModuleDto(ComponentTesting.newPrivateProjectDto(db.organizations().ins
ert())));

    failIfComponentIsNotAProjectOrView(module);
  }

  @Test
  public void fail_when_component_is_a_directory() {
    ComponentDto file =
db.components().insertComponent(newDirectory(ComponentTesting.newPrivateProjectDto(db.organizations().inser
t()), "A/B"));

    failIfComponentIsNotAProjectOrView(file);
  }

  @Test
  public void fail_when_component_is_a_file() {
    ComponentDto file =
db.components().insertComponent(newFileDto(ComponentTesting.newPrivateProjectDto(db.organizations().insert()
), null, "file-uuid"));

    failIfComponentIsNotAProjectOrView(file);
  }

  @Test
  public void fail_when_component_is_a_subview() {
    ComponentDto file =
db.components().insertComponent(newSubView(ComponentTesting.newView(db.organizations().insert())));

    failIfComponentIsNotAProjectOrView(file);
  }

  private void failIfComponentIsNotAProjectOrView(ComponentDto file) {
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Component '" + file.getDbKey() + "' (id: " + file.uuid() + ") must be a project
or a view.");
```

```java
      newRequest()
        .setParam(PARAM_GROUP_NAME, group.getName())
        .setParam(PARAM_PROJECT_ID, file.uuid())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();
  }

  @Test
  public void adding_a_project_permission_fails_if_project_is_not_set() throws Exception {
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    executeRequest(group, UserRole.ISSUE_ADMIN);
  }

  @Test
  public void adding_a_project_permission_fails_if_component_is_not_a_project() {
    OrganizationDto organizationDto = db.getDefaultOrganization();
    GroupDto group = db.users().insertGroup(organizationDto, "sonar-administrators");
    ComponentDto project = db.components().insertComponent(newPrivateProjectDto(organizationDto,
A_PROJECT_UUID).setDbKey(A_PROJECT_KEY));
    ComponentDto file = db.components().insertComponent(ComponentTesting.newFileDto(project, null, "file-
uuid"));
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PROJECT_ID, file.uuid())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();
  }

  @Test
  public void fail_when_get_request() {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(ServerException.class);

    newRequest()
      .setMethod("GET")
      .setParam(PARAM_GROUP_NAME, "sonar-administrators")
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();
  }
```

```java
@Test
public void fail_when_group_name_and_group_id_are_missing() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Group name or group id must be provided");

  newRequest()
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
}

@Test
public void fail_when_permission_is_missing() {
  GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(IllegalArgumentException.class);

  newRequest()
    .setParam(PARAM_GROUP_NAME, group.getName())
    .execute();
}

@Test
public void fail_if_not_administrator_of_organization() {
  GroupDto group = db.users().insertGroup();
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(IllegalArgumentException.class);

  newRequest()
    .setParam(PARAM_GROUP_NAME, group.getName())
    .execute();
}

@Test
public void fail_if_administrator_of_other_organization_only() {
  OrganizationDto org1 = db.organizations().insert();
  OrganizationDto org2 = db.organizations().insert();
  GroupDto group = db.users().insertGroup(org1, "the-group");
  loginAsAdmin(org2);

  expectedException.expect(ForbiddenException.class);

  newRequest()
    .setParam(PARAM_GROUP_ID, group.getId().toString())
```

```java
      .setParam(PARAM_PERMISSION, PROVISIONING)
      .execute();
  }

  @Test
  public void fail_when_project_uuid_and_project_key_are_provided() {
    GroupDto group = db.users().insertGroup();
    ComponentDto project =
  db.components().insertComponent(ComponentTesting.newPrivateProjectDto(db.organizations().insert()));
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Project id or project key can be provided, not both.");

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PROJECT_KEY, project.getDbKey())
      .execute();
  }

  @Test
  public void adding_global_permission_fails_if_not_administrator_of_organization() {
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    // user is administrator of another organization
    userSession.logIn().addPermission(ADMINISTER, "anotherOrg");

    expectedException.expect(ForbiddenException.class);

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, PROVISIONING)
      .execute();
  }

  @Test
  public void adding_project_permission_fails_if_not_administrator_of_project() {
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    ComponentDto project = db.components().insertPrivateProject();
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, PROVISIONING)
      .setParam(PARAM_PROJECT_KEY, project.getDbKey())
```

```java
      .execute();
  }

  /**
   * User is project administrator but not system administrator
   */
  @Test
  public void adding_project_permission_is_allowed_to_project_administrators() {
    GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
    ComponentDto project = db.components().insertPrivateProject();
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    newRequest()
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .execute();

    assertThat(db.users().selectGroupPermissions(group, project)).containsOnly(ISSUE_ADMIN);
  }

  @Test
  public void fails_when_adding_any_permission_to_group_AnyOne_on_a_private_project() {
    ComponentDto project = db.components().insertPrivateProject();
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    ProjectPermissions.ALL
      .forEach(permission -> {
        try {
          newRequest()
            .setParam(PARAM_GROUP_NAME, "anyone")
            .setParam(PARAM_PROJECT_ID, project.uuid())
            .setParam(PARAM_PERMISSION, permission)
            .execute();
          fail("a BadRequestException should have been raised for " + permission);
        } catch (BadRequestException e) {
          assertThat(e).hasMessage("No permission can be granted to Anyone on a private component");
        }
      });
  }

  @Test
  public void no_effect_when_adding_USER_permission_to_group_AnyOne_on_a_public_project() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    newRequest()
```

```java
    .setParam(PARAM_GROUP_NAME, "anyone")
    .setParam(PARAM_PROJECT_ID, project.uuid())
    .setParam(PARAM_PERMISSION, USER)
    .execute();

  assertThat(db.users().selectAnyonePermissions(organization, project)).isEmpty();
}

@Test
public void no_effect_when_adding_CODEVIEWER_permission_to_group_AnyOne_on_a_public_project() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertPublicProject(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

  newRequest()
   .setParam(PARAM_GROUP_NAME, "anyone")
   .setParam(PARAM_PROJECT_ID, project.uuid())
   .setParam(PARAM_PERMISSION, CODEVIEWER)
   .execute();

  assertThat(db.users().selectAnyonePermissions(organization, project)).isEmpty();
}

@Test
public void no_effect_when_adding_USER_permission_to_group_on_a_public_project() {
  OrganizationDto organization = db.organizations().insert();
  GroupDto group = db.users().insertGroup(organization);
  ComponentDto project = db.components().insertPublicProject(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

  newRequest()
   .setParam(PARAM_ORGANIZATION, organization.getKey())
   .setParam(PARAM_GROUP_NAME, group.getName())
   .setParam(PARAM_PROJECT_ID, project.uuid())
   .setParam(PARAM_PERMISSION, USER)
   .execute();

  assertThat(db.users().selectAnyonePermissions(organization, project)).isEmpty();
}

@Test
public void no_effect_when_adding_CODEVIEWER_permission_to_group_on_a_public_project() {
  OrganizationDto organization = db.organizations().insert();
  GroupDto group = db.users().insertGroup(organization);
  ComponentDto project = db.components().insertPublicProject(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

  newRequest()
```

```java
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, CODEVIEWER)
      .execute();

    assertThat(db.users().selectAnyonePermissions(organization, project)).isEmpty();
  }

  @Test
  public void fail_when_using_branch_db_key() throws Exception {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = db.components().insertMainBranch(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
    ComponentDto branch = db.components().insertProjectBranch(project);

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage(format("Project key '%s' not found", branch.getDbKey()));

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_PROJECT_KEY, branch.getDbKey())
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .execute();
  }

  @Test
  public void fail_when_using_branch_uuid() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = db.components().insertMainBranch(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
    ComponentDto branch = db.components().insertProjectBranch(project);

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage(format("Project id '%s' not found", branch.uuid()));

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_PROJECT_ID, branch.uuid())
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .execute();
  }

  private void executeRequest(GroupDto groupDto, String permission) {
```

```
    newRequest()
      .setParam(PARAM_GROUP_NAME, groupDto.getName())
      .setParam(PARAM_PERMISSION, permission)
      .execute();
  }


}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Test;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.GroupTesting;
import org.sonar.db.user.UserDto;
import org.sonar.db.user.UserTesting;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.i18n.I18nRule;
import org.sonarqube.ws.Permissions;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.core.permission.GlobalPermissions.PROVISIONING;
import static org.sonar.core.permission.GlobalPermissions.SCAN_EXECUTION;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_PROFILES;
```

```java
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;
import static org.sonar.db.permission.OrganizationPermission.SCAN;
import static org.sonar.test.JsonAssert.assertJson;

public class SearchGlobalPermissionsActionTest extends BasePermissionWsTest<SearchGlobalPermissionsAction>
{

  private I18nRule i18n = new I18nRule();

  @Override
  protected SearchGlobalPermissionsAction buildWsAction() {
    return new SearchGlobalPermissionsAction(db.getDbClient(), userSession, i18n, newPermissionWsSupport());
  }

  @Before
  public void setUp() {
    initI18nMessages();
  }

  @Test
  public void search_in_organization() {
    OrganizationDto org = db.organizations().insert();
    loginAsAdmin(org);
    GroupDto adminGroup = db.users().insertGroup(newGroup(org, "sonar-admins", "Administrators"));
    GroupDto userGroup = db.users().insertGroup(newGroup(org, "sonar-users", "Users"));
    db.users().insertPermissionOnAnyone(org, SCAN);
    db.users().insertPermissionOnGroup(userGroup, SCAN);
    db.users().insertPermissionOnGroup(userGroup, PROVISIONING);
    db.users().insertPermissionOnGroup(adminGroup, ADMINISTER);
    UserDto user = db.users().insertUser(newUserDto("user", "user-name"));
    UserDto adminUser = db.users().insertUser(newUserDto("admin", "admin-name"));
    db.organizations().addMember(org, user);
    db.organizations().addMember(org, adminUser);
    db.users().insertPermissionOnUser(org, user, PROVISION_PROJECTS);
    db.users().insertPermissionOnUser(org, user, ADMINISTER_QUALITY_PROFILES);
    db.users().insertPermissionOnUser(org, adminUser, ADMINISTER_QUALITY_PROFILES);
    db.users().insertPermissionOnUser(org, user, ADMINISTER_QUALITY_GATES);
    db.users().insertPermissionOnUser(org, adminUser, ADMINISTER_QUALITY_GATES);

    // to be excluded, permission on another organization (the default one)
    db.users().insertPermissionOnUser(db.getDefaultOrganization(), adminUser,
ADMINISTER_QUALITY_GATES);

    String result = newRequest()
      .setParam("organization", org.getKey())
      .execute()
      .getInput();
```

```java
    assertJson(result).isSimilarTo(getClass().getResource("search_global_permissions-example.json"));
  }

  @Test
  public void search_in_default_organization_by_default() {
    OrganizationDto org = db.organizations().insert();
    loginAsAdmin(org, db.getDefaultOrganization());

    UserDto user = db.users().insertUser();
    db.users().insertPermissionOnUser(db.getDefaultOrganization(), user, SCAN);
    db.organizations().addMember(db.getDefaultOrganization(), user);

    // to be ignored, by default organization is used when searching for permissions
    db.users().insertPermissionOnUser(org, user, ADMINISTER_QUALITY_GATES);
    db.organizations().addMember(org, user);

    Permissions.WsSearchGlobalPermissionsResponse result = newRequest()
      .executeProtobuf(Permissions.WsSearchGlobalPermissionsResponse.class);

    assertThat(result.getPermissionsCount()).isEqualTo(GlobalPermissions.ALL.size());
    for (Permissions.Permission permission : result.getPermissionsList()) {
      if (permission.getKey().equals(SCAN_EXECUTION)) {
        assertThat(permission.getUsersCount()).isEqualTo(1);
      } else {
        assertThat(permission.getUsersCount()).isEqualTo(0);
      }
    }
  }

  @Test
  public void supports_protobuf_response() {
    loginAsAdmin(db.getDefaultOrganization());

    Permissions.WsSearchGlobalPermissionsResponse result = newRequest()
      .executeProtobuf(Permissions.WsSearchGlobalPermissionsResponse.class);

    assertThat(result).isNotNull();
  }

  @Test
  public void fail_if_not_admin_of_default_organization() {
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest()
      .execute();
  }
```

```
@Test
public void fail_if_not_admin_of_specified_organization() {
  OrganizationDto org = db.organizations().insert();
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(ForbiddenException.class);

  newRequest()
    .setParam("organization", org.getKey())
    .execute();
}

@Test
public void fail_if_not_logged_in() {
  userSession.anonymous();

  expectedException.expect(UnauthorizedException.class);

  newRequest().execute();
}

@Test
public void fail_if_organization_does_not_exist() {
  expectedException.expect(NotFoundException.class);

  newRequest()
    .setParam("organization", "does_not_exist")
    .execute();
}

private void initI18nMessages() {
  i18n.put("global_permissions.admin", "Administer System");
  i18n.put("global_permissions.admin.desc", "Ability to perform all administration functions for the instance: " +
    "global configuration and personalization of default dashboards.");
  i18n.put("global_permissions.profileadmin", "Administer Quality Profiles");
  i18n.put("global_permissions.profileadmin.desc", "Ability to perform any action on the quality profiles.");
  i18n.put("global_permissions.gateadmin", "Administer Quality Gates");
  i18n.put("global_permissions.gateadmin.desc", "Ability to perform any action on the quality gates.");
  i18n.put("global_permissions.scan", "Execute Analysis");
  i18n.put("global_permissions.scan.desc", "Ability to execute analyses, and to get all settings required to perform
the analysis, " +
    "even the secured ones like the scm account password, the jira account password, and so on.");
  i18n.put("global_permissions.provisioning", "Create Projects");
  i18n.put("global_permissions.provisioning.desc", "Ability to initialize project structure before first analysis.");
}

private static UserDto newUserDto(String login, String name) {
```

```
      return UserTesting.newUserDto().setLogin(login).setName(name).setActive(true);
    }

  private static GroupDto newGroup(OrganizationDto org, String name, String description) {
      return
  GroupTesting.newGroupDto().setName(name).setDescription(description).setOrganizationUuid(org.getUuid());
    }
  }
```

```
package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Rule;
import org.junit.rules.ExpectedException;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.utils.internal.AlwaysIncreasingSystem2;
import org.sonar.db.DbClient;
import org.sonar.db.DbTester;
import org.sonar.db.component.ResourceTypesRule;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.template.PermissionTemplateDto;
import org.sonar.server.component.ComponentFinder;
import org.sonar.server.es.EsTester;
import org.sonar.server.es.ProjectIndexersImpl;
import org.sonar.server.organization.TestDefaultOrganizationProvider;
import org.sonar.server.permission.GroupPermissionChanger;
import org.sonar.server.permission.PermissionUpdater;
import org.sonar.server.permission.UserPermissionChanger;
import org.sonar.server.permission.index.FooIndexDefinition;
import org.sonar.server.permission.index.PermissionIndexer;
```

```java
import org.sonar.server.tester.UserSessionRule;
import org.sonar.server.usergroups.DefaultGroupFinder;
import org.sonar.server.usergroups.ws.GroupWsSupport;
import org.sonar.server.ws.TestRequest;
import org.sonar.server.ws.WsActionTester;

import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateDto;

public abstract class BasePermissionWsTest<A extends PermissionsWsAction> {

  @Rule
  public DbTester db = DbTester.create(new AlwaysIncreasingSystem2());

  @Rule
  public EsTester es = EsTester.createCustom(new FooIndexDefinition());

  @Rule
  public ExpectedException expectedException = ExpectedException.none();

  private TestDefaultOrganizationProvider defaultOrganizationProvider =
TestDefaultOrganizationProvider.from(db);
  protected UserSessionRule userSession = UserSessionRule.standalone();
  protected WsActionTester wsTester;

  @Before
  public void initWsTester() {
    wsTester = new WsActionTester(buildWsAction());
  }

  protected abstract A buildWsAction();

  protected GroupWsSupport newGroupWsSupport() {
    return new GroupWsSupport(db.getDbClient(), defaultOrganizationProvider, new
DefaultGroupFinder(db.getDbClient()));
  }

  protected PermissionWsSupport newPermissionWsSupport() {
    DbClient dbClient = db.getDbClient();
    return new PermissionWsSupport(dbClient, new ComponentFinder(dbClient, newRootResourceTypes()),
newGroupWsSupport());
  }

  protected ResourceTypesRule newRootResourceTypes() {
    return new ResourceTypesRule().setRootQualifiers(Qualifiers.PROJECT, Qualifiers.VIEW, Qualifiers.APP);
  }

  protected PermissionUpdater newPermissionUpdater() {
```

```
    return new PermissionUpdater(
      new ProjectIndexersImpl(new PermissionIndexer(db.getDbClient(), es.client())),
      new UserPermissionChanger(db.getDbClient()),
      new GroupPermissionChanger(db.getDbClient())));
  }

  protected TestRequest newRequest() {
    return wsTester.newRequest().setMethod("POST");
  }

  protected void loginAsAdmin(OrganizationDto org, OrganizationDto... otherOrgs) {
    userSession.logIn().addPermission(ADMINISTER, org);
    for (OrganizationDto otherOrg : otherOrgs) {
      userSession.addPermission(ADMINISTER, otherOrg);
    }
  }

  protected PermissionTemplateDto selectTemplateInDefaultOrganization(String name) {
    return db.getDbClient().permissionTemplateDao().selectByName(db.getSession(),
db.getDefaultOrganization().getUuid(), name);
  }

  protected PermissionTemplateDto addTemplate(OrganizationDto organizationDto) {
    PermissionTemplateDto dto = newPermissionTemplateDto()
      .setOrganizationUuid(organizationDto.getUuid());
    db.getDbClient().permissionTemplateDao().insert(db.getSession(), dto);
    db.commit();
    return dto;
  }

  protected PermissionTemplateDto addTemplateToDefaultOrganization() {
    return addTemplate(db.getDefaultOrganization());
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
```

```java
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Test;
import org.sonar.api.web.UserRole;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.ServerException;

import static java.lang.String.format;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.component.ComponentTesting.newDirectory;
import static org.sonar.db.component.ComponentTesting.newFileDto;
import static org.sonar.db.component.ComponentTesting.newModuleDto;
import static org.sonar.db.component.ComponentTesting.newPrivateProjectDto;
import static org.sonar.db.component.ComponentTesting.newSubView;
import static org.sonar.db.component.ComponentTesting.newView;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class AddUserActionTest extends BasePermissionWsTest<AddUserAction> {

  private UserDto user;

  @Before
  public void setUp() {
    user = db.users().insertUser("ray.bradbury");
    db.organizations().addMember(db.getDefaultOrganization(), user);
  }
```

```java
  @Override
  protected AddUserAction buildWsAction() {
    return new AddUserAction(db.getDbClient(), userSession, newPermissionUpdater(),
newPermissionWsSupport());
  }

  @Test
  public void add_permission_to_user_on_default_organization_if_organization_is_not_specified() {
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();

    assertThat(db.users().selectPermissionsOfUser(user, db.getDefaultOrganization())).containsOnly(ADMINISTER);
  }

  @Test
  public void add_permission_to_user_on_specified_organization() {
    OrganizationDto organization = db.organizations().insert();
    addUserAsMemberOfOrganization(organization);
    loginAsAdmin(organization);

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();

    assertThat(db.users().selectPermissionsOfUser(user, organization)).containsOnly(ADMINISTER);
  }

  @Test
  public void add_permission_to_project_referenced_by_its_id() {
    OrganizationDto organization = db.organizations().insert();
    addUserAsMemberOfOrganization(organization);
    ComponentDto project = db.components().insertPrivateProject(organization);
    loginAsAdmin(organization);

    newRequest()
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();

    assertThat(db.users().selectPermissionsOfUser(user, organization)).isEmpty();
    assertThat(db.users().selectProjectPermissionsOfUser(user, project)).containsOnly(SYSTEM_ADMIN);
```

```java
    }

    @Test
    public void add_permission_to_project_referenced_by_its_key() {
      ComponentDto project = db.components().insertPrivateProject();
      loginAsAdmin(db.getDefaultOrganization());

      newRequest()
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PROJECT_KEY, project.getDbKey())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();

      assertThat(db.users().selectPermissionsOfUser(user, db.getDefaultOrganization())).isEmpty();
      assertThat(db.users().selectProjectPermissionsOfUser(user, project)).containsOnly(SYSTEM_ADMIN);
    }

    @Test
    public void add_permission_to_view() {
      ComponentDto view = db.components().insertComponent(newView(db.getDefaultOrganization(), "view-
    uuid").setDbKey("view-key"));
      loginAsAdmin(db.getDefaultOrganization());

      newRequest()
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PROJECT_ID, view.uuid())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();

      assertThat(db.users().selectPermissionsOfUser(user, db.getDefaultOrganization())).isEmpty();
      assertThat(db.users().selectProjectPermissionsOfUser(user, view)).containsOnly(SYSTEM_ADMIN);
    }

    @Test
    public void fail_when_project_uuid_is_unknown() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(NotFoundException.class);

      newRequest()
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PROJECT_ID, "unknown-project-uuid")
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();
    }

    @Test
    public void fail_when_component_is_a_module() {
```

```java
  ComponentDto module =
db.components().insertComponent(newModuleDto(ComponentTesting.newPrivateProjectDto(db.organizations().ins
ert())));

  failIfComponentIsNotAProjectOrView(module);
}

@Test
public void fail_when_component_is_a_directory() {
  ComponentDto file =
db.components().insertComponent(newDirectory(ComponentTesting.newPrivateProjectDto(db.organizations().inser
t()), "A/B"));

  failIfComponentIsNotAProjectOrView(file);
}

@Test
public void fail_when_component_is_a_file() {
  ComponentDto file =
db.components().insertComponent(newFileDto(ComponentTesting.newPrivateProjectDto(db.organizations().insert()
), null, "file-uuid"));

  failIfComponentIsNotAProjectOrView(file);
}

@Test
public void fail_when_component_is_a_subview() {
  ComponentDto file =
db.components().insertComponent(newSubView(ComponentTesting.newView(db.organizations().insert())));

  failIfComponentIsNotAProjectOrView(file);
}

private void failIfComponentIsNotAProjectOrView(ComponentDto file) {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Component '" + file.getDbKey() + "' (id: " + file.uuid() + ") must be a project
or a view.");

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_ID, file.uuid())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
}

@Test
```

```java
public void fail_when_project_permission_without_project() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);

  newRequest()
   .setParam(PARAM_USER_LOGIN, user.getLogin())
   .setParam(PARAM_PERMISSION, UserRole.ISSUE_ADMIN)
   .execute();
}

@Test
public void fail_when_component_is_not_a_project() {
  db.components().insertComponent(newFileDto(newPrivateProjectDto(db.organizations().insert(), "project-uuid"),
null, "file-uuid"));
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);

  newRequest()
   .setParam(PARAM_USER_LOGIN, user.getLogin())
   .setParam(PARAM_PROJECT_ID, "file-uuid")
   .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
   .execute();
}

@Test
public void fail_when_get_request() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(ServerException.class);

  newRequest()
   .setMethod("GET")
   .setParam(PARAM_USER_LOGIN, "george.orwell")
   .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
   .execute();
}

@Test
public void fail_when_user_login_is_missing() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(IllegalArgumentException.class);

  newRequest()
   .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
   .execute();
```

```
}

@Test
public void fail_when_permission_is_missing() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(NotFoundException.class);

  newRequest()
    .setParam(PARAM_USER_LOGIN, "jrr.tolkien")
    .execute();
}

@Test
public void fail_when_project_uuid_and_project_key_are_provided() {
  db.components().insertPrivateProject();
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Project id or project key can be provided, not both.");

  newRequest()
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_ID, "project-uuid")
    .setParam(PARAM_PROJECT_KEY, "project-key")
    .execute();
}

@Test
public void adding_global_permission_fails_if_not_administrator_of_organization() {
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
}

@Test
public void adding_project_permission_fails_if_not_administrator_of_project() {
  ComponentDto project = db.components().insertPrivateProject();
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);
```

```java
  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .setParam(PARAM_PROJECT_KEY, project.getDbKey())
    .execute();
}

/**
 * User is project administrator but not system administrator
 */
@Test
public void adding_project_permission_is_allowed_to_project_administrators() {
  ComponentDto project = db.components().insertPrivateProject();

  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_KEY, project.getDbKey())
    .setParam(PARAM_PERMISSION, UserRole.ISSUE_ADMIN)
    .execute();

  assertThat(db.users().selectProjectPermissionsOfUser(user, project)).containsOnly(ISSUE_ADMIN);
}

@Test
public void organization_parameter_must_not_be_set_on_project_permissions() {
  ComponentDto project = db.components().insertPrivateProject();
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(IllegalArgumentException.class);
  expectedException.expectMessage("Organization must not be set when project is set.");

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_KEY, project.getDbKey())
    .setParam(PARAM_ORGANIZATION, "an_org")
    .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
    .execute();
}

@Test
public void fail_to_add_permission_when_user_is_not_member_of_given_organization() {
  // User is not member of given organization
  OrganizationDto otherOrganization = db.organizations().insert();
  addUserAsMemberOfOrganization(otherOrganization);
  OrganizationDto organization = db.organizations().insert(organizationDto ->
organizationDto.setKey("Organization key"));
```

```
  loginAsAdmin(organization);

  expectedException.expect(IllegalArgumentException.class);
  expectedException.expectMessage("User 'ray.bradbury' is not member of organization 'Organization key'");

  newRequest()
    .setParam(PARAM_ORGANIZATION, organization.getKey())
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
}

@Test
public void no_effect_when_adding_USER_permission_on_a_public_project() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertPublicProject(organization);
  addUserAsMemberOfOrganization(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_ID, project.uuid())
    .setParam(PARAM_PERMISSION, USER)
    .execute();

  assertThat(db.users().selectAnyonePermissions(organization, project)).isEmpty();
}

@Test
public void no_effect_when_adding_CODEVIEWER_permission_on_a_public_project() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertPublicProject(organization);
  addUserAsMemberOfOrganization(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_ID, project.uuid())
    .setParam(PARAM_PERMISSION, CODEVIEWER)
    .execute();

  assertThat(db.users().selectAnyonePermissions(organization, project)).isEmpty();
}

@Test
public void fail_when_using_branch_db_key() throws Exception {
  OrganizationDto organization = db.organizations().insert();
  addUserAsMemberOfOrganization(organization);
```

```java
      ComponentDto project = db.components().insertMainBranch(organization);
      userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
      ComponentDto branch = db.components().insertProjectBranch(project);

      expectedException.expect(NotFoundException.class);
      expectedException.expectMessage(format("Project key '%s' not found", branch.getDbKey()));

      newRequest()
        .setParam(PARAM_ORGANIZATION, organization.getKey())
        .setParam(PARAM_PROJECT_KEY, branch.getDbKey())
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();
    }

    @Test
    public void fail_when_using_branch_uuid() {
      OrganizationDto organization = db.organizations().insert();
      addUserAsMemberOfOrganization(organization);
      ComponentDto project = db.components().insertMainBranch(organization);
      userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
      ComponentDto branch = db.components().insertProjectBranch(project);

      expectedException.expect(NotFoundException.class);
      expectedException.expectMessage(format("Project id '%s' not found", branch.uuid()));

      newRequest()
        .setParam(PARAM_ORGANIZATION, organization.getKey())
        .setParam(PARAM_PROJECT_ID, branch.uuid())
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();
    }

    private void addUserAsMemberOfOrganization(OrganizationDto organization) {
      db.organizations().addMember(organization, user);
    }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
```

package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Test;
import org.sonar.api.resources.Qualifiers;
import org.sonar.api.web.UserRole;
import org.sonar.db.component.ComponentDbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.component.ResourceTypesRule;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.i18n.I18nRule;
import org.sonarqube.ws.Permissions;

import static java.lang.String.format;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.server.ws.WebService.Param.PAGE;
import static org.sonar.api.server.ws.WebService.Param.PAGE_SIZE;
import static org.sonar.api.server.ws.WebService.Param.TEXT_QUERY;
import static org.sonar.db.component.ComponentTesting.newPrivateProjectDto;
import static org.sonar.db.component.ComponentTesting.newProjectCopy;
import static org.sonar.db.component.ComponentTesting.newView;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.test.JsonAssert.assertJson;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_QUALIFIER;

public class SearchProjectPermissionsActionTest extends
BasePermissionWsTest<SearchProjectPermissionsAction> {

  private ComponentDbTester componentDb = new ComponentDbTester(db);
  private I18nRule i18n = new I18nRule();

```java
@Before
public void setUp() {
  i18n.setProjectPermissions();
  userSession.logIn().setSystemAdministrator();
}

@Override
protected SearchProjectPermissionsAction buildWsAction() {
  i18n.setProjectPermissions();
  ResourceTypesRule rootResourceTypes = newRootResourceTypes();
  PermissionWsSupport wsSupport = newPermissionWsSupport();
  return new SearchProjectPermissionsAction(db.getDbClient(), userSession, i18n, rootResourceTypes, wsSupport);
}

@Test
public void
search_project_permissions_counts_0_users_and_0_groups_on_public_project_without_any_specified_permission_
in_DB() {
  ComponentDto project = db.components().insertPublicProject();

  String result = newRequest().execute().getInput();

  assertJson(result)
    .ignoreFields("permissions")
    .isSimilarTo("{" +
    " \"paging\": {" +
    "   \"pageIndex\": 1," +
    "   \"pageSize\": 25," +
    "   \"total\": 1" +
    " }," +
    " \"projects\": [" +
    "   {" +
    "   \"id\": \"" + project.uuid() + "\"," +
    "   \"key\": \"" + project.getDbKey() + "\"," +
    "   \"name\": \"" + project.name() + "\"," +
    "   \"qualifier\": \"TRK\"," +
    "   \"permissions\": []" +
    "   }" +
    " ]" +
    "}");
}

@Test
public void
search_project_permissions_counts_0_users_and_0_groups_on_private_project_without_any_specified_permission
_in_DB() {
  ComponentDto project = db.components().insertPrivateProject();
```

```
    String result = newRequest().execute().getInput();

    assertJson(result)
      .ignoreFields("permissions")
      .isSimilarTo("{" +
      "  \"paging\": {" +
      "    \"pageIndex\": 1," +
      "    \"pageSize\": 25," +
      "    \"total\": 1" +
      "  }," +
      "  \"projects\": [" +
      "    {" +
      "      \"id\": \"" + project.uuid() + "\"," +
      "      \"key\": \"" + project.getDbKey() + "\"," +
      "      \"name\": \"" + project.name() + "\"," +
      "      \"qualifier\": \"TRK\"," +
      "      \"permissions\": []" +
      "    }" +
      "  ]" +
      "}");
  }

  @Test
  public void search_project_permissions() {
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    UserDto user3 = db.users().insertUser();

    ComponentDto jdk7 = insertJdk7();
    ComponentDto project2 = insertClang();
    ComponentDto view = insertView();
    insertProjectInView(jdk7, view);

    db.users().insertProjectPermissionOnUser(user1, UserRole.ISSUE_ADMIN, jdk7);
    db.users().insertProjectPermissionOnUser(user1, UserRole.ADMIN, jdk7);
    db.users().insertProjectPermissionOnUser(user2, UserRole.ADMIN, jdk7);
    db.users().insertProjectPermissionOnUser(user3, UserRole.ADMIN, jdk7);
    db.users().insertProjectPermissionOnUser(user1, UserRole.ISSUE_ADMIN, project2);
    db.users().insertProjectPermissionOnUser(user1, UserRole.ISSUE_ADMIN, view);
    // global permission
    db.users().insertPermissionOnUser(user1, ADMINISTER);

    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    GroupDto group3 = db.users().insertGroup();

    db.users().insertProjectPermissionOnAnyone(UserRole.ADMIN, jdk7);
```

```java
    db.users().insertProjectPermissionOnGroup(group1, UserRole.ADMIN, jdk7);
    db.users().insertProjectPermissionOnGroup(group2, UserRole.ADMIN, jdk7);
    db.users().insertProjectPermissionOnGroup(group3, UserRole.ADMIN, jdk7);
    db.users().insertProjectPermissionOnGroup(group2, UserRole.ADMIN, view);

    db.commit();

    String result = newRequest().execute().getInput();

    assertJson(result)
      .ignoreFields("permissions")
      .isSimilarTo(getClass().getResource("search_project_permissions-example.json"));
  }

  @Test
  public void empty_result() {
    String result = newRequest().execute().getInput();

    assertJson(result)
      .ignoreFields("permissions")
      .isSimilarTo(getClass().getResource("SearchProjectPermissionsActionTest/empty.json"));
  }

  @Test
  public void search_project_permissions_with_project_permission() {
    ComponentDto project = db.components().insertComponent(newPrivateProjectDto(db.getDefaultOrganization(),
"project-uuid"));
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    String result = newRequest()
      .setParam(PARAM_PROJECT_ID, "project-uuid")
      .execute().getInput();

    assertThat(result).contains("project-uuid");
  }

  @Test
  public void has_projects_ordered_by_name() {
    OrganizationDto organizationDto = db.organizations().insert();
    for (int i = 9; i >= 1; i--) {
      db.components().insertComponent(ComponentTesting.newPrivateProjectDto(organizationDto)
        .setName("project-name-" + i));
    }

    String result = newRequest()
      .setParam(PAGE, "1")
      .setParam(PAGE_SIZE, "3")
      .execute().getInput();
```

```java
    assertThat(result)
      .contains("project-name-1", "project-name-2", "project-name-3")
      .doesNotContain("project-name-4");
  }

  @Test
  public void search_by_query_on_name() {
componentDb.insertProjectAndSnapshot(ComponentTesting.newPrivateProjectDto(db.getDefaultOrganization()).se
tName("project-name"));
componentDb.insertProjectAndSnapshot(ComponentTesting.newPrivateProjectDto(db.getDefaultOrganization()).se
tName("another-name"));

  String result = newRequest()
    .setParam(TEXT_QUERY, "project")
    .execute().getInput();

  assertThat(result).contains("project-name")
    .doesNotContain("another-name");
  }

  @Test
  public void search_by_query_on_key_must_match_exactly() {
    OrganizationDto organizationDto = db.organizations().insert();
componentDb.insertProjectAndSnapshot(ComponentTesting.newPrivateProjectDto(organizationDto).setDbKey("pr
oject-key"));
componentDb.insertProjectAndSnapshot(ComponentTesting.newPrivateProjectDto(organizationDto).setDbKey("an
other-key"));

  String result = newRequest()
    .setParam(TEXT_QUERY, "project-key")
    .execute()
    .getInput();

  assertThat(result).contains("project-key")
    .doesNotContain("another-key");
  }

  @Test
  public void handle_more_than_1000_projects() {
   for (int i = 1; i <= 1001; i++) {
     componentDb.insertProjectAndSnapshot(newPrivateProjectDto(db.getDefaultOrganization(), "project-uuid-" +
i));
   }

  String result = newRequest()
    .setParam(TEXT_QUERY, "project")
    .setParam(PAGE_SIZE, "1001")
```

```java
    .execute()
    .getInput();

  assertThat(result).contains("project-uuid-1", "project-uuid-999", "project-uuid-1001");
}

@Test
public void filter_by_qualifier() {
  OrganizationDto organizationDto = db.organizations().insert();
  db.components().insertComponent(newView(organizationDto, "view-uuid"));
  db.components().insertComponent(newPrivateProjectDto(organizationDto, "project-uuid"));

  Permissions.SearchProjectPermissionsWsResponse result = newRequest()
    .setParam(PARAM_QUALIFIER, Qualifiers.PROJECT)
    .executeProtobuf(Permissions.SearchProjectPermissionsWsResponse.class);

  assertThat(result.getProjectsList())
    .extracting("id")
    .contains("project-uuid")
    .doesNotContain("view-uuid");
}

@Test
public void fail_if_not_logged_in() {
  userSession.anonymous();

  expectedException.expect(UnauthorizedException.class);

  newRequest().execute();
}

@Test
public void fail_if_not_admin() {
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequest().execute();
}

@Test
public void display_all_project_permissions() {
  String result = newRequest().execute().getInput();

  assertJson(result)
    .ignoreFields("permissions")
.isSimilarTo(getClass().getResource("SearchProjectPermissionsActionTest/display_all_project_permissions.json"));
}
```

```
@Test
public void fail_when_using_branch_db_key() throws Exception {
  ComponentDto project = db.components().insertMainBranch();
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
  ComponentDto branch = db.components().insertProjectBranch(project);

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage(format("Project key '%s' not found", branch.getDbKey()));

  newRequest()
    .setParam(PARAM_PROJECT_KEY, branch.getDbKey())
    .execute();
}

private ComponentDto insertView() {
  return db.components().insertComponent(newView(db.getDefaultOrganization())
    .setUuid("752d8bfd-420c-4a83-a4e5-8ab19b13c8fc")
    .setName("Java")
    .setDbKey("Java"));
}

private ComponentDto insertProjectInView(ComponentDto project, ComponentDto view) {
  return db.components().insertComponent(newProjectCopy("project-in-view-uuid", project, view));
}

private ComponentDto insertClang() {
  return db.components().insertComponent(newPrivateProjectDto(db.getDefaultOrganization(), "project-uuid-2")
    .setName("Clang")
    .setDbKey("clang")
    .setUuid("ce4c03d6-430f-40a9-b777-ad877c00aa4d"));
}

private ComponentDto insertJdk7() {
  return db.components().insertComponent(ComponentTesting.newPublicProjectDto(db.getDefaultOrganization())
    .setName("JDK 7")
    .setDbKey("net.java.openjdk:jdk7")
    .setUuid("0bd7b1e7-91d6-439e-a607-4a3a9aad3c6a"));
}
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
```

```
package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Test;
import org.sonar.api.web.UserRole;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.ServerException;

import static java.lang.String.format;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.PROVISIONING;
import static org.sonar.core.permission.GlobalPermissions.QUALITY_GATE_ADMIN;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.component.ComponentTesting.newDirectory;
import static org.sonar.db.component.ComponentTesting.newFileDto;
import static org.sonar.db.component.ComponentTesting.newModuleDto;
import static org.sonar.db.component.ComponentTesting.newPrivateProjectDto;
import static org.sonar.db.component.ComponentTesting.newSubView;
import static org.sonar.db.component.ComponentTesting.newView;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;
```

```java
public class RemoveUserActionTest extends BasePermissionWsTest<RemoveUserAction> {

  private static final String A_PROJECT_UUID = "project-uuid";
  private static final String A_PROJECT_KEY = "project-key";
  private static final String A_LOGIN = "ray.bradbury";

  private UserDto user;

  @Before
  public void setUp() {
    user = db.users().insertUser(A_LOGIN);
  }

  @Override
  protected RemoveUserAction buildWsAction() {
    return new RemoveUserAction(db.getDbClient(), userSession, newPermissionUpdater(),
newPermissionWsSupport());
  }

  @Test
  public void remove_permission_from_user() {
    db.users().insertPermissionOnUser(user, PROVISION_PROJECTS);
    db.users().insertPermissionOnUser(user, ADMINISTER_QUALITY_GATES);
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PERMISSION, QUALITY_GATE_ADMIN)
      .execute();

    assertThat(db.users().selectPermissionsOfUser(user,
db.getDefaultOrganization())).containsOnly(PROVISION_PROJECTS);
  }

  @Test
  public void fail_to_remove_admin_permission_if_last_admin() {
    db.users().insertPermissionOnUser(user, ADMINISTER);
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Last user with permission 'admin'. Permission cannot be removed.");

    newRequest()
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PERMISSION, ADMIN)
      .execute();
  }
```

```java
@Test
public void remove_permission_from_project() {
  ComponentDto project = db.components().insertComponent(newPrivateProjectDto(db.organizations().insert(),
A_PROJECT_UUID).setDbKey(A_PROJECT_KEY));
  db.users().insertProjectPermissionOnUser(user, CODEVIEWER, project);
  db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
   .setParam(PARAM_USER_LOGIN, user.getLogin())
   .setParam(PARAM_PROJECT_ID, project.uuid())
   .setParam(PARAM_PERMISSION, CODEVIEWER)
   .execute();

  assertThat(db.users().selectProjectPermissionsOfUser(user, project)).containsOnly(ISSUE_ADMIN);
}

@Test
public void remove_with_project_key() {
  ComponentDto project = db.components().insertComponent(newPrivateProjectDto(db.organizations().insert(),
A_PROJECT_UUID).setDbKey(A_PROJECT_KEY));
  db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);
  db.users().insertProjectPermissionOnUser(user, CODEVIEWER, project);
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
   .setParam(PARAM_USER_LOGIN, user.getLogin())
   .setParam(PARAM_PROJECT_KEY, project.getDbKey())
   .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
   .execute();

  assertThat(db.users().selectProjectPermissionsOfUser(user, project)).containsOnly(CODEVIEWER);
}

@Test
public void remove_with_view_uuid() {
  ComponentDto view = db.components().insertComponent(newView(db.organizations().insert(), "view-
uuid").setDbKey("view-key"));
  db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, view);
  db.users().insertProjectPermissionOnUser(user, ADMIN, view);
  loginAsAdmin(db.getDefaultOrganization());

  newRequest()
   .setParam(PARAM_USER_LOGIN, user.getLogin())
   .setParam(PARAM_PROJECT_KEY, view.getDbKey())
   .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
   .execute();
```

```java
      assertThat(db.users().selectProjectPermissionsOfUser(user, view)).containsOnly(ADMIN);
    }

    @Test
    public void fail_when_project_does_not_exist() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(NotFoundException.class);

      newRequest()
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PROJECT_ID, "unknown-project-uuid")
        .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
        .execute();
    }

    @Test
    public void fail_when_project_permission_without_permission() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(BadRequestException.class);

      newRequest()
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
        .execute();
    }

    @Test
    public void fail_when_component_is_a_module() {
      ComponentDto module =
    db.components().insertComponent(newModuleDto(ComponentTesting.newPrivateProjectDto(db.organizations().ins
    ert())));

      failIfComponentIsNotAProjectOrView(module);
    }

    @Test
    public void fail_when_component_is_a_directory() {
      ComponentDto file =
    db.components().insertComponent(newDirectory(ComponentTesting.newPrivateProjectDto(db.organizations().inser
    t()), "A/B"));

      failIfComponentIsNotAProjectOrView(file);
    }

    @Test
```

```java
 public void fail_when_component_is_a_file() {
   ComponentDto file =
db.components().insertComponent(newFileDto(ComponentTesting.newPrivateProjectDto(db.organizations().insert()
), null, "file-uuid"));

   failIfComponentIsNotAProjectOrView(file);
 }

 @Test
 public void fail_when_component_is_a_subview() {
   ComponentDto file =
db.components().insertComponent(newSubView(ComponentTesting.newView(db.organizations().insert())));

   failIfComponentIsNotAProjectOrView(file);
 }

 private void failIfComponentIsNotAProjectOrView(ComponentDto file) {
   loginAsAdmin(db.getDefaultOrganization());

   expectedException.expect(BadRequestException.class);
   expectedException.expectMessage("Component '" + file.getDbKey() + "' (id: " + file.uuid() + ") must be a project
or a view.");

   newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_ID, file.uuid())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
 }

 @Test
 public void fail_when_get_request() {
   loginAsAdmin(db.getDefaultOrganization());

   expectedException.expect(ServerException.class);

   newRequest()
    .setMethod("GET")
    .setParam(PARAM_USER_LOGIN, "george.orwell")
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
 }

 @Test
 public void fail_when_user_login_is_missing() {
   loginAsAdmin(db.getDefaultOrganization());

   expectedException.expect(IllegalArgumentException.class);
```

```java
  newRequest()
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
}

@Test
public void fail_when_permission_is_missing() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(IllegalArgumentException.class);

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .execute();
}

@Test
public void fail_when_project_uuid_and_project_key_are_provided() {
  ComponentDto project = db.components().insertComponent(newPrivateProjectDto(db.organizations().insert(),
A_PROJECT_UUID).setDbKey(A_PROJECT_KEY));
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Project id or project key can be provided, not both.");

  newRequest()
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_ID, project.uuid())
    .setParam(PARAM_PROJECT_KEY, project.getDbKey())
    .execute();
}

@Test
public void removing_global_permission_fails_if_not_administrator_of_organization() {
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, PROVISIONING)
    .execute();
}

@Test
public void removing_project_permission_fails_if_not_administrator_of_project() {
```

```java
    ComponentDto project = db.components().insertPrivateProject();
    userSession.logIn();

    expectedException.expect(ForbiddenException.class);

    newRequest()
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .setParam(PARAM_PROJECT_KEY, project.getDbKey())
      .execute();
  }

  /**
   * User is project administrator but not system administrator
   */
  @Test
  public void removing_project_permission_is_allowed_to_project_administrators() {
    ComponentDto project = db.components().insertPrivateProject();
    db.users().insertProjectPermissionOnUser(user, CODEVIEWER, project);
    db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    newRequest()
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .execute();

    assertThat(db.users().selectProjectPermissionsOfUser(user, project)).containsOnly(CODEVIEWER);
  }

  @Test
  public void fail_when_removing_USER_permission_on_a_public_project() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission user can't be removed from a public component");

    newRequest()
      .setParam(PARAM_USER_LOGIN, user.getLogin())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, USER)
      .execute();
  }

  @Test
```

```java
public void fail_when_removing_CODEVIEWER_permission_on_a_public_project() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertPublicProject(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Permission codeviewer can't be removed from a public component");

  newRequest()
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PROJECT_ID, project.uuid())
    .setParam(PARAM_PERMISSION, CODEVIEWER)
    .execute();
}

@Test
public void fail_when_using_branch_db_key() throws Exception {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertMainBranch(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
  ComponentDto branch = db.components().insertProjectBranch(project);

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage(format("Project key '%s' not found", branch.getDbKey()));

  newRequest()
    .setParam(PARAM_ORGANIZATION, organization.getKey())
    .setParam(PARAM_PROJECT_KEY, branch.getDbKey())
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
    .execute();
}

@Test
public void fail_when_using_branch_uuid() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertMainBranch(organization);
  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
  ComponentDto branch = db.components().insertProjectBranch(project);

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage(format("Project id '%s' not found", branch.uuid()));

  newRequest()
    .setParam(PARAM_ORGANIZATION, organization.getKey())
    .setParam(PARAM_PROJECT_ID, branch.uuid())
    .setParam(PARAM_USER_LOGIN, user.getLogin())
    .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
```

```
    .execute();
  }


}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Test;
import org.sonar.api.server.ws.WebService;
import org.sonar.db.DbClient;
import org.sonar.server.issue.ws.AvatarResolverImpl;
import org.sonar.server.permission.ws.template.TemplateGroupsAction;
import org.sonar.server.permission.ws.template.TemplateUsersAction;
import org.sonar.server.user.UserSession;
import org.sonar.server.ws.WsTester;

import static org.assertj.core.api.Assertions.assertThat;
import static org.mockito.Mockito.mock;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;

public class PermissionsWsTest {

  WsTester ws;

  @Before
  public void setUp() {
    DbClient dbClient = mock(DbClient.class);
    UserSession userSession = mock(UserSession.class);
    PermissionWsSupport permissionWsSupport = mock(PermissionWsSupport.class);
```

```java
    ws = new WsTester(new PermissionsWs(
      new TemplateUsersAction(dbClient, userSession, permissionWsSupport, new AvatarResolverImpl()),
      new TemplateGroupsAction(dbClient, userSession, permissionWsSupport)));
  }

  @Test
  public void define_controller() {
    WebService.Controller controller = controller();
    assertThat(controller).isNotNull();
    assertThat(controller.description()).isNotEmpty();
    assertThat(controller.since()).isEqualTo("3.7");
    assertThat(controller.actions()).hasSize(2);
  }

  @Test
  public void define_template_users() {
    WebService.Action action = controller().action("template_users");

    assertThat(action).isNotNull();
    assertThat(action.isPost()).isFalse();
    assertThat(action.isInternal()).isTrue();
    assertThat(action.since()).isEqualTo("5.2");
    assertThat(action.param(PARAM_PERMISSION).isRequired()).isFalse();
  }

  @Test
  public void define_template_groups() {
    WebService.Action action = controller().action("template_groups");

    assertThat(action).isNotNull();
    assertThat(action.isPost()).isFalse();
    assertThat(action.isInternal()).isTrue();
    assertThat(action.since()).isEqualTo("5.2");
  }

  private WebService.Controller controller() {
    return ws.controller("api/permissions");
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
```

```java
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.ws;

import org.junit.Before;
import org.junit.Test;
import org.sonar.api.web.UserRole;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.GroupPermissionDto;
import org.sonar.db.user.GroupDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;

import static java.lang.String.format;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.PROVISIONING;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.component.ComponentTesting.newDirectory;
import static org.sonar.db.component.ComponentTesting.newFileDto;
import static org.sonar.db.component.ComponentTesting.newModuleDto;
import static org.sonar.db.component.ComponentTesting.newSubView;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_GROUP_NAME;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;

public class RemoveGroupActionTest extends BasePermissionWsTest<RemoveGroupAction> {
```

```java
  private GroupDto aGroup;

  @Before
  public void setUp() {
    aGroup = db.users().insertGroup(db.getDefaultOrganization(), "sonar-administrators");
  }

  @Override
  protected RemoveGroupAction buildWsAction() {
    return new RemoveGroupAction(db.getDbClient(), userSession, newPermissionUpdater(),
newPermissionWsSupport());
  }

  @Test
  public void remove_permission_using_group_name() {
    db.users().insertPermissionOnGroup(aGroup, ADMINISTER);
    db.users().insertPermissionOnGroup(aGroup, PROVISION_PROJECTS);
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_NAME, aGroup.getName())
      .setParam(PARAM_PERMISSION, PROVISIONING)
      .execute();

    assertThat(db.users().selectGroupPermissions(aGroup, null)).containsOnly(ADMINISTER.getKey());
  }

  @Test
  public void remove_permission_using_group_id() {
    db.users().insertPermissionOnGroup(aGroup, ADMINISTER);
    db.users().insertPermissionOnGroup(aGroup, PROVISION_PROJECTS);
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_ID, aGroup.getId().toString())
      .setParam(PARAM_PERMISSION, PROVISION_PROJECTS.getKey())
      .execute();

    assertThat(db.users().selectGroupPermissions(aGroup, null)).containsOnly(ADMINISTER.getKey());
  }

  @Test
  public void remove_project_permission() {
    ComponentDto project = db.components().insertPrivateProject();
    db.users().insertPermissionOnGroup(aGroup, ADMINISTER);
    db.users().insertProjectPermissionOnGroup(aGroup, ADMIN, project);
    db.users().insertProjectPermissionOnGroup(aGroup, ISSUE_ADMIN, project);
```

```java
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_NAME, aGroup.getName())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, ADMIN)
      .execute();

    assertThat(db.users().selectGroupPermissions(aGroup, null)).containsOnly(ADMINISTER.getKey());
    assertThat(db.users().selectGroupPermissions(aGroup, project)).containsOnly(ISSUE_ADMIN);
  }

  @Test
  public void remove_with_view_uuid() {
    ComponentDto view = db.components().insertView();
    db.users().insertPermissionOnGroup(aGroup, ADMINISTER);
    db.users().insertProjectPermissionOnGroup(aGroup, ADMIN, view);
    db.users().insertProjectPermissionOnGroup(aGroup, ISSUE_ADMIN, view);
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_NAME, aGroup.getName())
      .setParam(PARAM_PROJECT_ID, view.uuid())
      .setParam(PARAM_PERMISSION, ADMIN)
      .execute();

    assertThat(db.users().selectGroupPermissions(aGroup, null)).containsOnly(ADMINISTER.getKey());
    assertThat(db.users().selectGroupPermissions(aGroup, view)).containsOnly(ISSUE_ADMIN);
  }

  @Test
  public void remove_with_project_key() {
    ComponentDto project = db.components().insertPrivateProject();
    db.users().insertPermissionOnGroup(aGroup, ADMINISTER);
    db.users().insertProjectPermissionOnGroup(aGroup, ADMIN, project);
    db.users().insertProjectPermissionOnGroup(aGroup, ISSUE_ADMIN, project);
    loginAsAdmin(db.getDefaultOrganization());

    newRequest()
      .setParam(PARAM_GROUP_NAME, aGroup.getName())
      .setParam(PARAM_PROJECT_KEY, project.getDbKey())
      .setParam(PARAM_PERMISSION, ADMIN)
      .execute();

    assertThat(db.users().selectGroupPermissions(aGroup, null)).containsOnly(ADMINISTER.getKey());
    assertThat(db.users().selectGroupPermissions(aGroup, project)).containsOnly(ISSUE_ADMIN);
  }
```

```java
@Test
public void fail_to_remove_last_admin_permission() throws Exception {
  db.users().insertPermissionOnGroup(aGroup, ADMINISTER);
  db.users().insertPermissionOnGroup(aGroup, PROVISION_PROJECTS);
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Last group with permission 'admin'. Permission cannot be removed.");

  executeRequest(aGroup, SYSTEM_ADMIN);
}

@Test
public void fail_when_project_does_not_exist() {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(NotFoundException.class);
  expectedException.expectMessage("Project id 'unknown-project-uuid' not found");

  newRequest()
    .setParam(PARAM_GROUP_NAME, aGroup.getName())
    .setParam(PARAM_PROJECT_ID, "unknown-project-uuid")
    .setParam(PARAM_PERMISSION, ADMINISTER.getKey())
    .execute();
}

@Test
public void fail_when_project_project_permission_without_project() throws Exception {
  loginAsAdmin(db.getDefaultOrganization());

  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Invalid global permission 'issueadmin'. Valid values are [admin, profileadmin,
gateadmin, scan, provisioning]");

  executeRequest(aGroup, ISSUE_ADMIN);
}

@Test
public void fail_when_component_is_a_module() {
  ComponentDto module =
db.components().insertComponent(newModuleDto(ComponentTesting.newPrivateProjectDto(db.organizations().ins
ert())));

  failIfComponentIsNotAProjectOrView(module);
}

@Test
public void fail_when_component_is_a_directory() {
```

```java
    ComponentDto file =
db.components().insertComponent(newDirectory(ComponentTesting.newPrivateProjectDto(db.organizations().inser
t()), "A/B"));

    failIfComponentIsNotAProjectOrView(file);
  }

  @Test
  public void fail_when_component_is_a_file() {
    ComponentDto file =
db.components().insertComponent(newFileDto(ComponentTesting.newPrivateProjectDto(db.organizations().insert()
), null, "file-uuid"));

    failIfComponentIsNotAProjectOrView(file);
  }

  @Test
  public void fail_when_component_is_a_subview() {
    ComponentDto file =
db.components().insertComponent(newSubView(ComponentTesting.newView(db.organizations().insert())));

    failIfComponentIsNotAProjectOrView(file);
  }

  private void failIfComponentIsNotAProjectOrView(ComponentDto file) {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Component '" + file.getDbKey() + "' (id: " + file.uuid() + ") must be a project
or a view.");

    newRequest()
      .setParam(PARAM_GROUP_NAME, aGroup.getName())
      .setParam(PARAM_PROJECT_ID, file.uuid())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();
  }

  @Test
  public void fail_when_group_name_is_missing() {
    loginAsAdmin(db.getDefaultOrganization());

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Group name or group id must be provided");

    newRequest()
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();
```

```
    }

    @Test
    public void fail_when_permission_name_and_id_are_missing() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(IllegalArgumentException.class);
      expectedException.expectMessage("The 'permission' parameter is missing");

      newRequest()
        .setParam(PARAM_GROUP_NAME, aGroup.getName())
        .execute();
    }

    @Test
    public void fail_when_group_id_does_not_exist() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(NotFoundException.class);
      expectedException.expectMessage("No group with id '999999'");

      newRequest()
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .setParam(PARAM_GROUP_ID, "999999")
        .execute();
    }

    @Test
    public void fail_when_project_uuid_and_project_key_are_provided() {
      ComponentDto project = db.components().insertPrivateProject();
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(BadRequestException.class);
      expectedException.expectMessage("Project id or project key can be provided, not both.");

      newRequest()
        .setParam(PARAM_GROUP_NAME, aGroup.getName())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .setParam(PARAM_PROJECT_ID, project.uuid())
        .setParam(PARAM_PROJECT_KEY, project.getDbKey())
        .execute();
    }

    private void executeRequest(GroupDto groupDto, String permission) {
      newRequest()
        .setParam(PARAM_GROUP_NAME, groupDto.getName())
        .setParam(PARAM_PERMISSION, permission)
        .execute();
```

```java
}

private void executeRequest(GroupDto groupDto, OrganizationDto organizationDto, String permission) {
  newRequest()
    .setParam(PARAM_GROUP_NAME, groupDto.getName())
    .setParam(PARAM_PERMISSION, permission)
    .setParam(PARAM_ORGANIZATION, organizationDto.getKey())
    .execute();
}

@Test
public void removing_global_permission_fails_if_not_administrator_of_organization() {
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequest()
    .setParam(PARAM_GROUP_NAME, aGroup.getName())
    .setParam(PARAM_PERMISSION, PROVISIONING)
    .execute();
}

@Test
public void removing_project_permission_fails_if_not_administrator_of_project() {
  ComponentDto project = db.components().insertPrivateProject();
  userSession.logIn();

  expectedException.expect(ForbiddenException.class);

  newRequest()
    .setParam(PARAM_GROUP_NAME, aGroup.getName())
    .setParam(PARAM_PERMISSION, PROVISIONING)
    .setParam(PARAM_PROJECT_KEY, project.getDbKey())
    .execute();
}

/**
 * User is project administrator but not system administrator
 */
@Test
public void removing_project_permission_is_allowed_to_project_administrators() {
  ComponentDto project = db.components().insertPrivateProject();
  db.users().insertProjectPermissionOnGroup(aGroup, CODEVIEWER, project);
  db.users().insertProjectPermissionOnGroup(aGroup, ISSUE_ADMIN, project);

  userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
  newRequest()
    .setParam(PARAM_GROUP_NAME, aGroup.getName())
```

```java
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .execute();

    assertThat(db.users().selectGroupPermissions(aGroup, project)).containsOnly(CODEVIEWER);
  }

  @Test
  public void no_effect_when_removing_any_permission_from_group_AnyOne_on_a_private_project() {
    ComponentDto project = db.components().insertPrivateProject();
    ProjectPermissions.ALL
      .forEach(perm -> unsafeInsertProjectPermissionOnAnyone(perm, project));
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    ProjectPermissions.ALL
      .forEach(permission -> {
       newRequest()
         .setParam(PARAM_GROUP_NAME, "anyone")
         .setParam(PARAM_PROJECT_ID, project.uuid())
         .setParam(PARAM_PERMISSION, permission)
         .execute();

       assertThat(db.users().selectAnyonePermissions(db.getDefaultOrganization(), project)).contains(permission);
      });
  }

  @Test
  public void fail_when_removing_USER_permission_from_group_AnyOne_on_a_public_project() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission user can't be removed from a public component");

    newRequest()
      .setParam(PARAM_GROUP_NAME, "anyone")
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, USER)
      .execute();
  }

  @Test
  public void fail_when_removing_CODEVIEWER_permission_from_group_AnyOne_on_a_public_project() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
```

```java
    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission codeviewer can't be removed from a public component");

    newRequest()
      .setParam(PARAM_GROUP_NAME, "anyone")
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, CODEVIEWER)
      .execute();
  }

  @Test
  public void fail_when_removing_USER_permission_from_group_on_a_public_project() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = db.components().insertPublicProject(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission user can't be removed from a public component");

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, USER)
      .execute();
  }

  @Test
  public void fail_when_removing_CODEVIEWER_permission_from_group_on_a_public_project() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = db.components().insertPublicProject(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission codeviewer can't be removed from a public component");

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .setParam(PARAM_PERMISSION, CODEVIEWER)
      .execute();
  }

  @Test
  public void fail_when_using_branch_db_key() throws Exception {
```

```java
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = db.components().insertMainBranch(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
    ComponentDto branch = db.components().insertProjectBranch(project);

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage(format("Project key '%s' not found", branch.getDbKey()));

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_PROJECT_KEY, branch.getDbKey())
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();
  }

  @Test
  public void fail_when_using_branch_uuid() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = db.components().insertMainBranch(organization);
    userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
    ComponentDto branch = db.components().insertProjectBranch(project);

    expectedException.expect(NotFoundException.class);
    expectedException.expectMessage(format("Project id '%s' not found", branch.uuid()));

    newRequest()
      .setParam(PARAM_ORGANIZATION, organization.getKey())
      .setParam(PARAM_PROJECT_ID, branch.uuid())
      .setParam(PARAM_GROUP_NAME, group.getName())
      .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
      .execute();
  }

  private void unsafeInsertProjectPermissionOnAnyone(String perm, ComponentDto project) {
    GroupPermissionDto dto = new GroupPermissionDto()
      .setOrganizationUuid(project.getOrganizationUuid())
      .setGroupId(null)
      .setRole(perm)
      .setResourceId(project.getId());
    db.getDbClient().groupPermissionDao().insert(db.getSession(), dto);
    db.commit();
  }
}
/*
 * SonarQube
```

package org.sonar.server.permission.ws;

import org.junit.Test;
import org.sonar.api.server.ws.WebService.Param;
import org.sonar.api.server.ws.WebService.SelectionMode;
import org.sonar.api.web.UserRole;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.exceptions.ForbiddenException;
import org.sonar.server.exceptions.NotFoundException;
import org.sonar.server.exceptions.UnauthorizedException;
import org.sonar.server.issue.ws.AvatarResolverImpl;

import static java.lang.String.format;
import static org.apache.commons.lang.StringUtils.countMatches;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.server.ws.WebService.Param.PAGE;
import static org.sonar.api.server.ws.WebService.Param.PAGE_SIZE;
import static org.sonar.api.server.ws.WebService.Param.TEXT_QUERY;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.component.ComponentTesting.newPrivateProjectDto;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_PROFILES;
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;
import static org.sonar.db.permission.OrganizationPermission.SCAN;
import static org.sonar.db.user.UserTesting.newUserDto;

```java
import static org.sonar.test.JsonAssert.assertJson;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_ORGANIZATION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PERMISSION;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_ID;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_PROJECT_KEY;
import static org.sonarqube.ws.client.permission.PermissionsWsParameters.PARAM_USER_LOGIN;

public class UsersActionTest extends BasePermissionWsTest<UsersAction> {

 @Override
 protected UsersAction buildWsAction() {
   return new UsersAction(db.getDbClient(), userSession, newPermissionWsSupport(), new AvatarResolverImpl());
 }

 @Test
 public void search_for_users_with_response_example() {
   UserDto user1 =
db.users().insertUser(newUserDto().setLogin("admin").setName("Administrator").setEmail("admin@admin.com"));
   db.organizations().addMember(db.getDefaultOrganization(), user1);
   UserDto user2 = db.users().insertUser(newUserDto().setLogin("george.orwell").setName("George
Orwell").setEmail("george.orwell@1984.net"));
   db.organizations().addMember(db.getDefaultOrganization(), user2);
   db.users().insertPermissionOnUser(user1, ADMINISTER_QUALITY_PROFILES);
   db.users().insertPermissionOnUser(user1, ADMINISTER);
   db.users().insertPermissionOnUser(user1, ADMINISTER_QUALITY_GATES);
   db.users().insertPermissionOnUser(user2, SCAN);

   loginAsAdmin(db.getDefaultOrganization());
   String result = newRequest().execute().getInput();

   assertJson(result).withStrictArrayOrder().isSimilarTo(getClass().getResource("users-example.json"));
 }

 @Test
 public void search_for_users_with_one_permission() {
   insertUsersHavingGlobalPermissions();

   loginAsAdmin(db.getDefaultOrganization());
   String result = newRequest().setParam("permission", "scan").execute().getInput();

   assertJson(result).withStrictArrayOrder().isSimilarTo(getClass().getResource("UsersActionTest/users.json"));
 }

 @Test
 public void search_for_users_with_permission_on_project() {
   // User has permission on project
   ComponentDto project =
db.components().insertComponent(ComponentTesting.newPrivateProjectDto(db.getDefaultOrganization()));
```

```java
    UserDto user = db.users().insertUser(newUserDto());
    db.organizations().addMember(db.getDefaultOrganization(), user);
    db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);

    // User has permission on another project
    ComponentDto anotherProject =
db.components().insertComponent(ComponentTesting.newPrivateProjectDto(db.getDefaultOrganization()));
    UserDto userHavePermissionOnAnotherProject = db.users().insertUser(newUserDto());
    db.organizations().addMember(db.getDefaultOrganization(), userHavePermissionOnAnotherProject);
    db.users().insertProjectPermissionOnUser(userHavePermissionOnAnotherProject, ISSUE_ADMIN,
anotherProject);

    // User has no permission
    UserDto withoutPermission = db.users().insertUser(newUserDto());
    db.organizations().addMember(db.getDefaultOrganization(), withoutPermission);

    userSession.logIn().addProjectPermission(SYSTEM_ADMIN, project);
    String result = newRequest()
      .setParam(PARAM_PERMISSION, ISSUE_ADMIN)
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .execute()
      .getInput();

    assertThat(result).contains(user.getLogin())
      .doesNotContain(userHavePermissionOnAnotherProject.getLogin())
      .doesNotContain(withoutPermission.getLogin());
  }

  @Test
  public void search_only_for_users_with_permission_when_no_search_query() {
    // User have permission on project
    ComponentDto project = db.components().insertPrivateProject();
    UserDto user = db.users().insertUser();
    db.organizations().addMember(db.getDefaultOrganization(), user);
    db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);

    // User has no permission
    UserDto withoutPermission = db.users().insertUser();
    db.organizations().addMember(db.getDefaultOrganization(), withoutPermission);

    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
      .setParam(PARAM_PROJECT_ID, project.uuid())
      .execute()
      .getInput();

    assertThat(result)
      .contains(user.getLogin())
```

```java
      .doesNotContain(withoutPermission.getLogin());
 }

 @Test
 public void search_also_for_users_without_permission_when_filtering_name() {
  // User with permission on project
  ComponentDto project =
db.components().insertComponent(ComponentTesting.newPrivateProjectDto(db.organizations().insert()));
  UserDto user = db.users().insertUser(newUserDto("with-permission-login", "with-permission-name", "with-
permission-email"));
  db.organizations().addMember(db.getDefaultOrganization(), user);
  db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);

  // User without permission
  UserDto withoutPermission = db.users().insertUser(newUserDto("without-permission-login", "without-
permission-name", "without-permission-email"));
  db.organizations().addMember(db.getDefaultOrganization(), withoutPermission);
  UserDto anotherUser = db.users().insertUser(newUserDto("another-user", "another-user", "another-user"));
  db.organizations().addMember(db.getDefaultOrganization(), anotherUser);

  loginAsAdmin(db.getDefaultOrganization());
  String result = newRequest()
   .setParam(PARAM_PROJECT_ID, project.uuid())
   .setParam(TEXT_QUERY, "with")
   .execute()
   .getInput();

  assertThat(result).contains(user.getLogin(),
withoutPermission.getLogin()).doesNotContain(anotherUser.getLogin());
 }

 @Test
 public void search_also_for_users_without_permission_when_filtering_email() {
  // User with permission on project
  ComponentDto project =
db.components().insertComponent(ComponentTesting.newPrivateProjectDto(db.organizations().insert()));
  UserDto user = db.users().insertUser(newUserDto("with-permission-login", "with-permission-name", "with-
permission-email"));
  db.organizations().addMember(db.getDefaultOrganization(), user);
  db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);

  // User without permission
  UserDto withoutPermission = db.users().insertUser(newUserDto("without-permission-login", "without-
permission-name", "without-permission-email"));
  db.organizations().addMember(db.getDefaultOrganization(), withoutPermission);
  UserDto anotherUser = db.users().insertUser(newUserDto("another-user", "another-user", "another-user"));
  db.organizations().addMember(db.getDefaultOrganization(), anotherUser);
```

```java
    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest().setParam(PARAM_PROJECT_ID, project.uuid()).setParam(TEXT_QUERY,
"email").execute().getInput();

    assertThat(result).contains(user.getLogin(),
withoutPermission.getLogin()).doesNotContain(anotherUser.getLogin());
  }

  @Test
  public void search_also_for_users_without_permission_when_filtering_login() {
    // User with permission on project
    ComponentDto project =
db.components().insertComponent(ComponentTesting.newPrivateProjectDto(db.organizations().insert()));
    UserDto user = db.users().insertUser(newUserDto("with-permission-login", "with-permission-name", "with-
permission-email"));
    db.organizations().addMember(db.getDefaultOrganization(), user);
    db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);

    // User without permission
    UserDto withoutPermission = db.users().insertUser(newUserDto("without-permission-login", "without-
permission-name", "without-permission-email"));
    db.organizations().addMember(db.getDefaultOrganization(), withoutPermission);
    UserDto anotherUser = db.users().insertUser(newUserDto("another-user", "another-user", "another-user"));
    db.organizations().addMember(db.getDefaultOrganization(), anotherUser);

    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest().setParam(PARAM_PROJECT_ID, project.uuid()).setParam(TEXT_QUERY,
"login").execute().getInput();

    assertThat(result).contains(user.getLogin(),
withoutPermission.getLogin()).doesNotContain(anotherUser.getLogin());
  }

  @Test
  public void search_for_users_with_query_as_a_parameter() {
    insertUsersHavingGlobalPermissions();

    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
      .setParam("permission", "scan")
      .setParam(TEXT_QUERY, "ame-1")
      .execute()
      .getInput();

    assertThat(result).contains("login-1")
      .doesNotContain("login-2")
      .doesNotContain("login-3");
  }
```

```java
  @Test
  public void search_for_users_with_select_as_a_parameter() {
    insertUsersHavingGlobalPermissions();

    loginAsAdmin(db.getDefaultOrganization());
    String result = newRequest()
      .execute()
      .getInput();

    assertThat(result).contains("login-1", "login-2", "login-3");
  }

  @Test
  public void search_for_users_is_paginated() {
   for (int i = 9; i >= 0; i--) {
     UserDto user = db.users().insertUser(newUserDto().setName("user-" + i));
     db.organizations().addMember(db.getDefaultOrganization(), user);
     db.users().insertPermissionOnUser(user, ADMINISTER);
     db.users().insertPermissionOnUser(user, ADMINISTER_QUALITY_GATES);
    }
    loginAsAdmin(db.getDefaultOrganization());

    assertJson(newRequest().setParam(PAGE, "1").setParam(PAGE_SIZE,
"2").execute().getInput()).withStrictArrayOrder().isSimilarTo("{\n" +
      " \"paging\": {\n" +
      "   \"pageIndex\": 1,\n" +
      "   \"pageSize\": 2,\n" +
      "   \"total\": 10\n" +
      " },\n" +
      " \"users\": [\n" +
      "   {\n" +
      "     \"name\": \"user-0\"\n" +
      "   },\n" +
      "   {\n" +
      "     \"name\": \"user-1\"\n" +
      "   }\n" +
      " ]\n" +
      "}");
    assertJson(newRequest().setParam(PAGE, "3").setParam(PAGE_SIZE,
"4").execute().getInput()).withStrictArrayOrder().isSimilarTo("{\n" +
      " \"paging\": {\n" +
      "   \"pageIndex\": 3,\n" +
      "   \"pageSize\": 4,\n" +
      "   \"total\": 10\n" +
      " },\n" +
      " \"users\": [\n" +
      "   {\n" +
```

```
            "   \"name\": \"user-8\"\n" +
            "   },\n" +
            "   {\n" +
            "   \"name\": \"user-9\"\n" +
            "   }\n" +
            " ]\n" +
            "}");
    }

    @Test
    public void return_more_than_20_permissions() {
      loginAsAdmin(db.getDefaultOrganization());
      for (int i = 0; i < 30; i++) {
        UserDto user = db.users().insertUser(newUserDto().setLogin("user-" + i));
        db.organizations().addMember(db.getDefaultOrganization(), user);
        db.users().insertPermissionOnUser(user, SCAN);
        db.users().insertPermissionOnUser(user, PROVISION_PROJECTS);
      }

      String result = newRequest()
        .setParam(PAGE_SIZE, "100")
        .execute()
        .getInput();

      assertThat(countMatches(result, "scan")).isEqualTo(30);
    }

    @Test
    public void fail_if_project_permission_without_project() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(BadRequestException.class);

      newRequest()
        .setParam(PARAM_PERMISSION, UserRole.ISSUE_ADMIN)
        .setParam(Param.SELECTED, SelectionMode.ALL.value())
        .execute();
    }

    @Test
    public void fail_if_insufficient_privileges() {
      userSession.logIn("login");

      expectedException.expect(ForbiddenException.class);

      newRequest()
        .setParam("permission", SYSTEM_ADMIN)
        .execute();
```

```
    }

    @Test
    public void fail_if_not_logged_in() {
      userSession.anonymous();

      expectedException.expect(UnauthorizedException.class);

      newRequest()
        .setParam("permission", SYSTEM_ADMIN)
        .execute();
    }

    @Test
    public void fail_if_project_uuid_and_project_key_are_provided() {
      db.components().insertComponent(newPrivateProjectDto(db.organizations().insert(), "project-
uuid").setDbKey("project-key"));
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(BadRequestException.class);
      expectedException.expectMessage("Project id or project key can be provided, not both.");

      newRequest()
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .setParam(PARAM_PROJECT_ID, "project-uuid")
        .setParam(PARAM_PROJECT_KEY, "project-key")
        .execute();
    }

    @Test
    public void fail_if_search_query_is_too_short() {
      loginAsAdmin(db.getDefaultOrganization());

      expectedException.expect(IllegalArgumentException.class);
      expectedException.expectMessage("'q' length (2) is shorter than the minimum authorized (3)");

      newRequest().setParam(TEXT_QUERY, "ab").execute();
    }

    @Test
    public void fail_when_using_branch_db_key() throws Exception {
      OrganizationDto organization = db.organizations().insert();
      UserDto user = db.users().insertUser(newUserDto());
      ComponentDto project = db.components().insertMainBranch(organization);
      ComponentDto branch = db.components().insertProjectBranch(project);
      db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);
      userSession.logIn().addProjectPermission(UserRole.ADMIN, project);
```

```
      expectedException.expect(NotFoundException.class);
      expectedException.expectMessage(format("Project key '%s' not found", branch.getDbKey()));

      newRequest()
        .setParam(PARAM_ORGANIZATION, organization.getKey())
        .setParam(PARAM_PROJECT_KEY, branch.getDbKey())
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();
    }

    @Test
    public void fail_when_using_branch_uuid() {
      OrganizationDto organization = db.organizations().insert();
      UserDto user = db.users().insertUser(newUserDto());
      ComponentDto project = db.components().insertMainBranch(organization);
      ComponentDto branch = db.components().insertProjectBranch(project);
      db.users().insertProjectPermissionOnUser(user, ISSUE_ADMIN, project);
      userSession.logIn().addProjectPermission(UserRole.ADMIN, project);

      expectedException.expect(NotFoundException.class);
      expectedException.expectMessage(format("Project id '%s' not found", branch.uuid()));

      newRequest()
        .setParam(PARAM_ORGANIZATION, organization.getKey())
        .setParam(PARAM_PROJECT_ID, branch.uuid())
        .setParam(PARAM_USER_LOGIN, user.getLogin())
        .setParam(PARAM_PERMISSION, SYSTEM_ADMIN)
        .execute();
    }

    private void insertUsersHavingGlobalPermissions() {
      UserDto user1 = db.users().insertUser(newUserDto("login-1", "name-1", "email-1"));
      db.organizations().addMember(db.getDefaultOrganization(), user1);
      UserDto user2 = db.users().insertUser(newUserDto("login-2", "name-2", "email-2"));
      db.organizations().addMember(db.getDefaultOrganization(), user2);
      UserDto user3 = db.users().insertUser(newUserDto("login-3", "name-3", "email-3"));
      db.organizations().addMember(db.getDefaultOrganization(), user3);
      db.users().insertPermissionOnUser(user1, SCAN);
      db.users().insertPermissionOnUser(user2, SCAN);
      db.users().insertPermissionOnUser(user3, ADMINISTER);
    }

}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
```

```java
package org.sonar.server.permission;

import java.util.Collections;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
import org.sonar.server.exceptions.BadRequestException;

import static com.google.common.collect.Lists.newArrayList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.server.permission.ApplyPermissionTemplateQuery.create;

public class ApplyPermissionTemplateQueryTest {

 @Rule
 public ExpectedException expectedException = ExpectedException.none();

 @Test
 public void should_populate_with_params() {
  ApplyPermissionTemplateQuery query = create("my_template_key", newArrayList("1", "2", "3"));

  assertThat(query.getTemplateUuid()).isEqualTo("my_template_key");
  assertThat(query.getComponentKeys()).containsOnly("1", "2", "3");
 }

 @Test
 public void should_invalidate_query_with_empty_name() {
  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Permission template is mandatory");

  ApplyPermissionTemplateQuery.create("", newArrayList("1", "2", "3"));
 }
```

```java
  @Test
  public void should_invalidate_query_with_no_components() {
    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("No project provided. Please provide at least one project.");

    ApplyPermissionTemplateQuery.create("my_template_key", Collections.emptyList());
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;

import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import java.util.Map;
import java.util.function.Function;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.core.util.stream.MoreCollectors;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.component.ComponentDbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.GroupPermissionDto;
import org.sonar.db.user.GroupDto;
```

```java
import org.sonar.db.user.UserDbTester;
import org.sonar.db.user.UserDto;

import static java.util.Arrays.asList;
import static java.util.Collections.singletonList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.resources.Qualifiers.APP;
import static org.sonar.api.resources.Qualifiers.PROJECT;
import static org.sonar.api.resources.Qualifiers.VIEW;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.USER;

public class PermissionIndexerDaoTest {

  @Rule
  public DbTester dbTester = DbTester.create(System2.INSTANCE);

  private DbClient dbClient = dbTester.getDbClient();
  private DbSession dbSession = dbTester.getSession();

  private ComponentDbTester componentDbTester = new ComponentDbTester(dbTester);
  private UserDbTester userDbTester = new UserDbTester(dbTester);

  private OrganizationDto organization;
  private ComponentDto publicProject;
  private ComponentDto privateProject1;
  private ComponentDto privateProject2;
  private ComponentDto view1;
  private ComponentDto view2;
  private ComponentDto application;
  private UserDto user1;
  private UserDto user2;
  private GroupDto group;

  private PermissionIndexerDao underTest = new PermissionIndexerDao();

  @Before
  public void setUp() {
    organization = dbTester.organizations().insert();
    publicProject = componentDbTester.insertPublicProject(organization);
    privateProject1 = componentDbTester.insertPrivateProject(organization);
    privateProject2 = componentDbTester.insertPrivateProject(organization);
    view1 = componentDbTester.insertView(organization);
    view2 = componentDbTester.insertView(organization);
    application = componentDbTester.insertApplication(organization);
    user1 = userDbTester.insertUser();
    user2 = userDbTester.insertUser();
    group = userDbTester.insertGroup(organization);
```

```java
  }

  @Test
  public void select_all() {
    insertTestDataForProjectsAndViews();

    Collection<PermissionIndexerDao.Dto> dtos = underTest.selectAll(dbClient, dbSession);
    assertThat(dtos).hasSize(6);

    PermissionIndexerDao.Dto publicProjectAuthorization = getByProjectUuid(publicProject.uuid(), dtos);
    isPublic(publicProjectAuthorization, PROJECT);

    PermissionIndexerDao.Dto view1Authorization = getByProjectUuid(view1.uuid(), dtos);
    isPublic(view1Authorization, VIEW);

    PermissionIndexerDao.Dto applicationAuthorization = getByProjectUuid(application.uuid(), dtos);
    isPublic(applicationAuthorization, APP);

    PermissionIndexerDao.Dto privateProject1Authorization = getByProjectUuid(privateProject1.uuid(), dtos);
    assertThat(privateProject1Authorization.getGroupIds()).containsOnly(group.getId());
    assertThat(privateProject1Authorization.isAllowAnyone()).isFalse();
    assertThat(privateProject1Authorization.getUserIds()).containsOnly(user1.getId(), user2.getId());
    assertThat(privateProject1Authorization.getQualifier()).isEqualTo(PROJECT);

    PermissionIndexerDao.Dto privateProject2Authorization = getByProjectUuid(privateProject2.uuid(), dtos);
    assertThat(privateProject2Authorization.getGroupIds()).isEmpty();
    assertThat(privateProject2Authorization.isAllowAnyone()).isFalse();
    assertThat(privateProject2Authorization.getUserIds()).containsOnly(user1.getId());
    assertThat(privateProject2Authorization.getQualifier()).isEqualTo(PROJECT);

    PermissionIndexerDao.Dto view2Authorization = getByProjectUuid(view2.uuid(), dtos);
    isPublic(view2Authorization, VIEW);
  }

  @Test
  public void selectByUuids() {
    insertTestDataForProjectsAndViews();

    Map<String, PermissionIndexerDao.Dto> dtos = underTest
      .selectByUuids(dbClient, dbSession, asList(publicProject.uuid(), privateProject1.uuid(), privateProject2.uuid(),
view1.uuid(), view2.uuid(), application.uuid()))
      .stream()
      .collect(MoreCollectors.uniqueIndex(PermissionIndexerDao.Dto::getProjectUuid, Function.identity()));
    assertThat(dtos).hasSize(6);

    PermissionIndexerDao.Dto publicProjectAuthorization = dtos.get(publicProject.uuid());
    isPublic(publicProjectAuthorization, PROJECT);
```

```java
      PermissionIndexerDao.Dto view1Authorization = dtos.get(view1.uuid());
      isPublic(view1Authorization, VIEW);

      PermissionIndexerDao.Dto applicationAuthorization = dtos.get(application.uuid());
      isPublic(applicationAuthorization, APP);

      PermissionIndexerDao.Dto privateProject1Authorization = dtos.get(privateProject1.uuid());
      assertThat(privateProject1Authorization.getGroupIds()).containsOnly(group.getId());
      assertThat(privateProject1Authorization.isAllowAnyone()).isFalse();
      assertThat(privateProject1Authorization.getUserIds()).containsOnly(user1.getId(), user2.getId());
      assertThat(privateProject1Authorization.getQualifier()).isEqualTo(PROJECT);

      PermissionIndexerDao.Dto privateProject2Authorization = dtos.get(privateProject2.uuid());
      assertThat(privateProject2Authorization.getGroupIds()).isEmpty();
      assertThat(privateProject2Authorization.isAllowAnyone()).isFalse();
      assertThat(privateProject2Authorization.getUserIds()).containsOnly(user1.getId());
      assertThat(privateProject2Authorization.getQualifier()).isEqualTo(PROJECT);

      PermissionIndexerDao.Dto view2Authorization = dtos.get(view2.uuid());
      isPublic(view2Authorization, VIEW);
    }

    @Test
    public void selectByUuids_returns_empty_list_when_project_does_not_exist() {
      insertTestDataForProjectsAndViews();

      List<PermissionIndexerDao.Dto> dtos = underTest.selectByUuids(dbClient, dbSession, asList("missing"));
      assertThat(dtos).isEmpty();
    }

    @Test
    public void select_by_projects_with_high_number_of_projects() {
      List<String> projectUuids = new ArrayList<>();
      for (int i = 0; i < 350; i++) {
        ComponentDto project = ComponentTesting.newPrivateProjectDto(organization, Integer.toString(i));
        dbClient.componentDao().insert(dbSession, project);
        projectUuids.add(project.uuid());
        GroupPermissionDto dto = new GroupPermissionDto()
          .setOrganizationUuid(group.getOrganizationUuid())
          .setGroupId(group.getId())
          .setRole(USER)
          .setResourceId(project.getId());
        dbClient.groupPermissionDao().insert(dbSession, dto);
      }
      dbSession.commit();

      assertThat(underTest.selectByUuids(dbClient, dbSession, projectUuids))
        .hasSize(350)
```

```java
    .extracting(PermissionIndexerDao.Dto::getProjectUuid)
    .containsAll(projectUuids);
}

@Test
public void return_private_project_without_any_permission_when_no_permission_in_DB() {
  List<PermissionIndexerDao.Dto> dtos = underTest.selectByUuids(dbClient, dbSession,
singletonList(privateProject1.uuid()));

  // no permissions
  assertThat(dtos).hasSize(1);
  PermissionIndexerDao.Dto dto = dtos.get(0);
  assertThat(dto.getGroupIds()).isEmpty();
  assertThat(dto.getUserIds()).isEmpty();
  assertThat(dto.isAllowAnyone()).isFalse();
  assertThat(dto.getProjectUuid()).isEqualTo(privateProject1.uuid());
  assertThat(dto.getQualifier()).isEqualTo(privateProject1.qualifier());
}

@Test
public void return_public_project_with_only_AllowAnyone_true_when_no_permission_in_DB() {
  List<PermissionIndexerDao.Dto> dtos = underTest.selectByUuids(dbClient, dbSession,
singletonList(publicProject.uuid()));

  assertThat(dtos).hasSize(1);
  PermissionIndexerDao.Dto dto = dtos.get(0);
  assertThat(dto.getGroupIds()).isEmpty();
  assertThat(dto.getUserIds()).isEmpty();
  assertThat(dto.isAllowAnyone()).isTrue();
  assertThat(dto.getProjectUuid()).isEqualTo(publicProject.uuid());
  assertThat(dto.getQualifier()).isEqualTo(publicProject.qualifier());
}

@Test
public void
return_private_project_with_AllowAnyone_false_and_user_id_when_user_is_granted_USER_permission_directly(
) {
  dbTester.users().insertProjectPermissionOnUser(user1, USER, privateProject1);
  List<PermissionIndexerDao.Dto> dtos = underTest.selectByUuids(dbClient, dbSession,
singletonList(privateProject1.uuid()));

  assertThat(dtos).hasSize(1);
  PermissionIndexerDao.Dto dto = dtos.get(0);
  assertThat(dto.getGroupIds()).isEmpty();
  assertThat(dto.getUserIds()).containsOnly(user1.getId());
  assertThat(dto.isAllowAnyone()).isFalse();
  assertThat(dto.getProjectUuid()).isEqualTo(privateProject1.uuid());
  assertThat(dto.getQualifier()).isEqualTo(privateProject1.qualifier());
```

```java
  }

  @Test
  public void
return_private_project_with_AllowAnyone_false_and_group_id_but_not_user_id_when_user_is_granted_USER_p
ermission_through_group() {
    dbTester.users().insertMember(group, user1);
    dbTester.users().insertProjectPermissionOnGroup(group, USER, privateProject1);
    List<PermissionIndexerDao.Dto> dtos = underTest.selectByUuids(dbClient, dbSession,
singletonList(privateProject1.uuid()));

    assertThat(dtos).hasSize(1);
    PermissionIndexerDao.Dto dto = dtos.get(0);
    assertThat(dto.getGroupIds()).containsOnly(group.getId());
    assertThat(dto.getUserIds()).isEmpty();
    assertThat(dto.isAllowAnyone()).isFalse();
    assertThat(dto.getProjectUuid()).isEqualTo(privateProject1.uuid());
    assertThat(dto.getQualifier()).isEqualTo(privateProject1.qualifier());
  }

  private void isPublic(PermissionIndexerDao.Dto view1Authorization, String qualifier) {
    assertThat(view1Authorization.getGroupIds()).isEmpty();
    assertThat(view1Authorization.isAllowAnyone()).isTrue();
    assertThat(view1Authorization.getUserIds()).isEmpty();
    assertThat(view1Authorization.getQualifier()).isEqualTo(qualifier);
  }

  private static PermissionIndexerDao.Dto getByProjectUuid(String projectUuid,
Collection<PermissionIndexerDao.Dto> dtos) {
    return dtos.stream().filter(dto ->
dto.getProjectUuid().equals(projectUuid)).findFirst().orElseThrow(IllegalArgumentException::new);
  }

  private void insertTestDataForProjectsAndViews() {
    // user1 has USER access on both private projects
    userDbTester.insertProjectPermissionOnUser(user1, ADMIN, publicProject);
    userDbTester.insertProjectPermissionOnUser(user1, USER, privateProject1);
    userDbTester.insertProjectPermissionOnUser(user1, USER, privateProject2);
    userDbTester.insertProjectPermissionOnUser(user1, ADMIN, view1);
    userDbTester.insertProjectPermissionOnUser(user1, ADMIN, application);

    // user2 has USER access on privateProject1 only
    userDbTester.insertProjectPermissionOnUser(user2, USER, privateProject1);
    userDbTester.insertProjectPermissionOnUser(user2, ADMIN, privateProject2);

    // group1 has USER access on privateProject1 only
    userDbTester.insertProjectPermissionOnGroup(group, USER, privateProject1);
    userDbTester.insertProjectPermissionOnGroup(group, ADMIN, privateProject1);
```

```java
    userDbTester.insertProjectPermissionOnGroup(group, ADMIN, view1);
    userDbTester.insertProjectPermissionOnGroup(group, ADMIN, application);
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;

import org.elasticsearch.index.query.MatchAllQueryBuilder;
import org.elasticsearch.index.query.QueryBuilder;
import org.elasticsearch.join.query.HasParentQueryBuilder;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.GroupTesting;
import org.sonar.server.tester.UserSessionRule;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.test.JsonAssert.assertJson;

public class AuthorizationTypeSupportTest {

  @Rule
  public UserSessionRule userSession = UserSessionRule.standalone();

  private AuthorizationTypeSupport underTest = new AuthorizationTypeSupport(userSession);

  @Test
  public void createQueryFilter_does_not_include_permission_filters_if_user_is_flagged_as_root() {
    userSession.logIn().setRoot();
```

```java
    QueryBuilder filter = underTest.createQueryFilter();

    assertThat(filter).isInstanceOf(MatchAllQueryBuilder.class);
  }

  @Test
  public void createQueryFilter_sets_filter_on_anyone_group_if_user_is_anonymous() {
    userSession.anonymous();

    HasParentQueryBuilder filter = (HasParentQueryBuilder) underTest.createQueryFilter();

    assertJson(filter.toString()).isSimilarTo("{" +
      " \"has_parent\" : {" +
      "   \"query\" : {" +
      "     \"bool\" : {" +
      "       \"filter\" : [{" +
      "         \"bool\" : {" +
      "           \"should\" : [{" +
      "             \"term\" : {" +
      "               \"allowAnyone\" : {\"value\": true}" +
      "             }" +
      "           }]" +
      "         }" +
      "       }]" +
      "     }" +
      "   }," +
      "   \"parent_type\" : \"authorization\"" +
      " }" +
      "}");
  }

  @Test
  public void createQueryFilter_sets_filter_on_anyone_and_user_id_if_user_is_logged_in_but_has_no_groups() {
    userSession.logIn().setUserId(1234);

    HasParentQueryBuilder filter = (HasParentQueryBuilder) underTest.createQueryFilter();

    assertJson(filter.toString()).isSimilarTo("{" +
      " \"has_parent\": {" +
      "   \"query\": {" +
      "     \"bool\": {" +
      "       \"filter\": [{" +
      "         \"bool\": {" +
      "           \"should\": [" +
      "             {" +
      "               \"term\": {" +
      "                 \"allowAnyone\": {\"value\": true}" +
      "               }" +
```

```
"          }," +
"          {" +
"            \"term\": {" +
"              \"userIds\": {\"value\": 1234}" +
"            }" +
"          }" +
"        ]" +
"      }" +
"    }]" +
"    }" +
"  }," +
"  \"parent_type\": \"authorization\"" +
"  }" +
"}");
}

@Test
public void
createQueryFilter_sets_filter_on_anyone_and_user_id_and_group_ids_if_user_is_logged_in_and_has_groups() {
  GroupDto group1 = GroupTesting.newGroupDto().setId(10);
  GroupDto group2 = GroupTesting.newGroupDto().setId(11);
  userSession.logIn().setUserId(1234).setGroups(group1, group2);

  HasParentQueryBuilder filter = (HasParentQueryBuilder) underTest.createQueryFilter();

  assertJson(filter.toString()).isSimilarTo("{" +
  "  \"has_parent\": {" +
  "    \"query\": {" +
  "      \"bool\": {" +
  "        \"filter\": [{" +
  "          \"bool\": {" +
  "            \"should\": [" +
  "              {" +
  "                \"term\": {" +
  "                  \"allowAnyone\": {\"value\": true}" +
  "                }" +
  "              }," +
  "              {" +
  "                \"term\": {" +
  "                  \"userIds\": {\"value\": 1234}" +
  "                }" +
  "              }," +
  "              {" +
  "                \"term\": {" +
  "                  \"groupIds\": {\"value\": 10}" +
  "                }" +
  "              }," +
  "              {" +
```

```
"            \"term\": {" +
"              \"groupIds\": {\"value\": 11}" +
"            }" +
"          }" +
"        ]" +
"      }" +
"    }]" +
"  }" +
"  }," +
"  \"parent_type\": \"authorization\"" +
" }" +
"}");
  }
}
```

```
package org.sonar.server.permission.index;

import java.util.Collection;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
import org.sonar.api.utils.System2;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.es.EsQueueDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.es.EsTester;
```

```java
import org.sonar.server.es.IndexType;
import org.sonar.server.es.IndexingResult;
import org.sonar.server.es.ProjectIndexer;
import org.sonar.server.tester.UserSessionRule;

import static java.util.Arrays.asList;
import static java.util.Collections.singletonList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.server.es.ProjectIndexer.Cause.PERMISSION_CHANGE;

public class PermissionIndexerTest {

 private static final IndexType INDEX_TYPE_FOO_AUTH =
 AuthorizationTypeSupport.getAuthorizationIndexType(FooIndexDefinition.INDEX_TYPE_FOO);

 @Rule
 public ExpectedException expectedException = ExpectedException.none();
 @Rule
 public DbTester db = DbTester.create(System2.INSTANCE);
 @Rule
 public EsTester es = EsTester.createCustom(new FooIndexDefinition());
 @Rule
 public UserSessionRule userSession = UserSessionRule.standalone();

 private FooIndex fooIndex = new FooIndex(es.client(), new AuthorizationTypeSupport(userSession));
 private FooIndexer fooIndexer = new FooIndexer(es.client());
 private PermissionIndexer underTest = new PermissionIndexer(db.getDbClient(), es.client(), fooIndexer);

 @Test
 public void indexOnStartup_grants_access_to_any_user_and_to_group_Anyone_on_public_projects() {
  ComponentDto project = createAndIndexPublicProject();
  UserDto user1 = db.users().insertUser();
  UserDto user2 = db.users().insertUser();

  indexOnStartup();

  verifyAnyoneAuthorized(project);
  verifyAuthorized(project, user1);
  verifyAuthorized(project, user2);
 }

 @Test
 public void deletion_resilience_will_deindex_projects() {
  ComponentDto project1 = createUnindexedPublicProject();
  ComponentDto project2 = createUnindexedPublicProject();
  //UserDto user1 = db.users().insertUser();
```

```java
    indexOnStartup();
    assertThat(es.countDocuments(INDEX_TYPE_FOO_AUTH)).isEqualTo(2);

    // Simulate a indexation issue
    db.getDbClient().componentDao().delete(db.getSession(), project1.getId());
    underTest.prepareForRecovery(db.getSession(), asList(project1.uuid()),
ProjectIndexer.Cause.PROJECT_DELETION);
    assertThat(db.countRowsOfTable(db.getSession(), "es_queue")).isEqualTo(1);
    Collection<EsQueueDto> esQueueDtos = db.getDbClient().esQueueDao().selectForRecovery(db.getSession(),
Long.MAX_VALUE, 2);

    underTest.index(db.getSession(), esQueueDtos);

    assertThat(db.countRowsOfTable(db.getSession(), "es_queue")).isEqualTo(0);
    assertThat(es.countDocuments(INDEX_TYPE_FOO_AUTH)).isEqualTo(1);
  }

  @Test
  public void indexOnStartup_grants_access_to_user() {
    ComponentDto project = createAndIndexPrivateProject();
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user1, USER, project);
    db.users().insertProjectPermissionOnUser(user2, ADMIN, project);

    indexOnStartup();

    // anonymous
    verifyAnyoneNotAuthorized(project);

    // user1 has access
    verifyAuthorized(project, user1);

    // user2 has not access (only USER permission is accepted)
    verifyNotAuthorized(project, user2);
  }

  @Test
  public void indexOnStartup_grants_access_to_group_on_private_project() {
    ComponentDto project = createAndIndexPrivateProject();
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    UserDto user3 = db.users().insertUser();
    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    db.users().insertProjectPermissionOnGroup(group1, USER, project);
    db.users().insertProjectPermissionOnGroup(group2, ADMIN, project);
```

```
    indexOnStartup();

    // anonymous
    verifyAnyoneNotAuthorized(project);

    // group1 has access
    verifyAuthorized(project, user1, group1);

    // group2 has not access (only USER permission is accepted)
    verifyNotAuthorized(project, user2, group2);

    // user3 is not in any group
    verifyNotAuthorized(project, user3);
  }

  @Test
  public void indexOnStartup_grants_access_to_user_and_group() {
    ComponentDto project = createAndIndexPrivateProject();
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    GroupDto group = db.users().insertGroup();
    db.users().insertMember(group, user2);
    db.users().insertProjectPermissionOnUser(user1, USER, project);
    db.users().insertProjectPermissionOnGroup(group, USER, project);

    indexOnStartup();

    // anonymous
    verifyAnyoneNotAuthorized(project);

    // has direct access
    verifyAuthorized(project, user1);

    // has access through group
    verifyAuthorized(project, user1, group);

    // no access
    verifyNotAuthorized(project, user2);
  }

  @Test
  public void indexOnStartup_does_not_grant_access_to_anybody_on_private_project() {
    ComponentDto project = createAndIndexPrivateProject();
    UserDto user = db.users().insertUser();
    GroupDto group = db.users().insertGroup();

    indexOnStartup();
```

```java
    verifyAnyoneNotAuthorized(project);
    verifyNotAuthorized(project, user);
    verifyNotAuthorized(project, user, group);
  }

  @Test
  public void indexOnStartup_grants_access_to_anybody_on_public_project() {
    ComponentDto project = createAndIndexPublicProject();
    UserDto user = db.users().insertUser();
    GroupDto group = db.users().insertGroup();

    indexOnStartup();

    verifyAnyoneAuthorized(project);
    verifyAuthorized(project, user);
    verifyAuthorized(project, user, group);
  }

  @Test
  public void indexOnStartup_grants_access_to_anybody_on_view() {
    ComponentDto view = createAndIndexView();
    UserDto user = db.users().insertUser();
    GroupDto group = db.users().insertGroup();

    indexOnStartup();

    verifyAnyoneAuthorized(view);
    verifyAuthorized(view, user);
    verifyAuthorized(view, user, group);
  }

  @Test
  public void indexOnStartup_grants_access_on_many_projects() {
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    ComponentDto project = null;
    for (int i = 0; i < 10; i++) {
      project = createAndIndexPrivateProject();
      db.users().insertProjectPermissionOnUser(user1, USER, project);
    }

    indexOnStartup();

    verifyAnyoneNotAuthorized(project);
    verifyAuthorized(project, user1);
    verifyNotAuthorized(project, user2);
  }
```

```java
@Test
public void public_projects_are_visible_to_anybody_whatever_the_organization() {
  ComponentDto projectOnOrg1 = createAndIndexPublicProject(db.organizations().insert());
  ComponentDto projectOnOrg2 = createAndIndexPublicProject(db.organizations().insert());
  UserDto user = db.users().insertUser();

  indexOnStartup();

  verifyAnyoneAuthorized(projectOnOrg1);
  verifyAnyoneAuthorized(projectOnOrg2);
  verifyAuthorized(projectOnOrg1, user);
  verifyAuthorized(projectOnOrg2, user);
}

@Test
public void indexOnAnalysis_does_nothing_because_CE_does_not_touch_permissions() {
  ComponentDto project = createAndIndexPublicProject();

  underTest.indexOnAnalysis(project.uuid());

  assertThatAuthIndexHasSize(0);
  verifyAnyoneNotAuthorized(project);
}

@Test
public void permissions_are_not_updated_on_project_tags_update() {
  ComponentDto project = createAndIndexPublicProject();

  indexPermissions(project, ProjectIndexer.Cause.PROJECT_TAGS_UPDATE);

  assertThatAuthIndexHasSize(0);
  verifyAnyoneNotAuthorized(project);
}

@Test
public void permissions_are_not_updated_on_project_key_update() {
  ComponentDto project = createAndIndexPublicProject();

  indexPermissions(project, ProjectIndexer.Cause.PROJECT_TAGS_UPDATE);

  assertThatAuthIndexHasSize(0);
  verifyAnyoneNotAuthorized(project);
}

@Test
public void index_permissions_on_project_creation() {
  ComponentDto project = createAndIndexPrivateProject();
  UserDto user = db.users().insertUser();
```

```java
    db.users().insertProjectPermissionOnUser(user, USER, project);

    indexPermissions(project, ProjectIndexer.Cause.PROJECT_CREATION);

    assertThatAuthIndexHasSize(1);
    verifyAuthorized(project, user);
  }

  @Test
  public void index_permissions_on_permission_change() {
    ComponentDto project = createAndIndexPrivateProject();
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user1, USER, project);
    indexPermissions(project, ProjectIndexer.Cause.PROJECT_CREATION);
    verifyAuthorized(project, user1);
    verifyNotAuthorized(project, user2);

    db.users().insertProjectPermissionOnUser(user2, USER, project);
    indexPermissions(project, PERMISSION_CHANGE);

    verifyAuthorized(project, user1);
    verifyAuthorized(project, user1);
  }

  @Test
  public void delete_permissions_on_project_deletion() {
    ComponentDto project = createAndIndexPrivateProject();
    UserDto user = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user, USER, project);
    indexPermissions(project, ProjectIndexer.Cause.PROJECT_CREATION);
    verifyAuthorized(project, user);

    db.getDbClient().componentDao().delete(db.getSession(), project.getId());
    indexPermissions(project, ProjectIndexer.Cause.PROJECT_DELETION);

    verifyNotAuthorized(project, user);
    assertThatAuthIndexHasSize(0);
  }

  @Test
  public void errors_during_indexing_are_recovered() {
    ComponentDto project = createAndIndexPublicProject();
    es.lockWrites(INDEX_TYPE_FOO_AUTH);

    IndexingResult result = indexPermissions(project, PERMISSION_CHANGE);
    assertThat(result.getTotal()).isEqualTo(1L);
    assertThat(result.getFailures()).isEqualTo(1L);
```

```java
    // index is still read-only, fail to recover
    result = recover();
    assertThat(result.getTotal()).isEqualTo(1L);
    assertThat(result.getFailures()).isEqualTo(1L);
    assertThatAuthIndexHasSize(0);
    assertThatEsQueueTableHasSize(1);

    es.unlockWrites(INDEX_TYPE_FOO_AUTH);

    result = recover();
    assertThat(result.getTotal()).isEqualTo(1L);
    assertThat(result.getFailures()).isEqualTo(0L);
    verifyAnyoneAuthorized(project);
    assertThatEsQueueTableHasSize(0);
  }

  private void assertThatAuthIndexHasSize(int expectedSize) {
    IndexType authIndexType = underTest.getIndexTypes().iterator().next();
    assertThat(es.countDocuments(authIndexType)).isEqualTo(expectedSize);
  }

  private void indexOnStartup() {
    underTest.indexOnStartup(underTest.getIndexTypes());
  }

  private void verifyAuthorized(ComponentDto project, UserDto user) {
    logIn(user);
    verifyAuthorized(project, true);
  }

  private void verifyAuthorized(ComponentDto project, UserDto user, GroupDto group) {
    logIn(user).setGroups(group);
    verifyAuthorized(project, true);
  }

  private void verifyNotAuthorized(ComponentDto project, UserDto user) {
    logIn(user);
    verifyAuthorized(project, false);
  }

  private void verifyNotAuthorized(ComponentDto project, UserDto user, GroupDto group) {
    logIn(user).setGroups(group);
    verifyAuthorized(project, false);
  }

  private void verifyAnyoneAuthorized(ComponentDto project) {
    userSession.anonymous();
```

```java
    verifyAuthorized(project, true);
  }

  private void verifyAnyoneNotAuthorized(ComponentDto project) {
    userSession.anonymous();
    verifyAuthorized(project, false);
  }

  private void verifyAuthorized(ComponentDto project, boolean expectedAccess) {
    assertThat(fooIndex.hasAccessToProject(project.uuid())).isEqualTo(expectedAccess);
  }

  private UserSessionRule logIn(UserDto u) {
    userSession.logIn(u.getLogin()).setUserId(u.getId());
    return userSession;
  }

  private IndexingResult indexPermissions(ComponentDto project, ProjectIndexer.Cause cause) {
    DbSession dbSession = db.getSession();
    Collection<EsQueueDto> items = underTest.prepareForRecovery(dbSession, singletonList(project.uuid()), cause);
    dbSession.commit();
    return underTest.index(dbSession, items);
  }

  private ComponentDto createUnindexedPublicProject() {
    ComponentDto project = db.components().insertPublicProject();
    return project;
  }

  private ComponentDto createAndIndexPrivateProject() {
    ComponentDto project = db.components().insertPrivateProject();
    fooIndexer.indexOnAnalysis(project.uuid());
    return project;
  }

  private ComponentDto createAndIndexPublicProject() {
    ComponentDto project = db.components().insertPublicProject();
    fooIndexer.indexOnAnalysis(project.uuid());
    return project;
  }

  private ComponentDto createAndIndexView() {
    ComponentDto view = db.components().insertView();
    fooIndexer.indexOnAnalysis(view.uuid());
    return view;
  }

  private ComponentDto createAndIndexPublicProject(OrganizationDto org) {
```

```java
    ComponentDto project = db.components().insertPublicProject(org);
    fooIndexer.indexOnAnalysis(project.uuid());
    return project;
  }

  private IndexingResult recover() {
    Collection<EsQueueDto> items = db.getDbClient().esQueueDao().selectForRecovery(db.getSession(),
System.currentTimeMillis() + 1_000L, 10);
    return underTest.index(db.getSession(), items);
  }

  private void assertThatEsQueueTableHasSize(int expectedSize) {
    assertThat(db.countRowsOfTable("es_queue")).isEqualTo(expectedSize);
  }

}
```

```java
package org.sonar.server.permission.index;

import org.sonar.api.config.internal.MapSettings;
import org.sonar.server.es.IndexDefinition;
import org.sonar.server.es.IndexType;
import org.sonar.server.es.NewIndex;

import static org.sonar.server.es.NewIndex.SettingsConfiguration.MANUAL_REFRESH_INTERVAL;
import static org.sonar.server.es.NewIndex.SettingsConfiguration.newBuilder;

public class FooIndexDefinition implements IndexDefinition {

  public static final String FOO_INDEX = "foos";
```

```java
  public static final String FOO_TYPE = "foo";
  public static final IndexType INDEX_TYPE_FOO = new IndexType(FOO_INDEX, FOO_TYPE);
  public static final String FIELD_NAME = "name";
  public static final String FIELD_PROJECT_UUID = "projectUuid";

  @Override
  public void define(IndexDefinitionContext context) {
    NewIndex index = context.create(FOO_INDEX, newBuilder(new
MapSettings().asConfig()).setRefreshInterval(MANUAL_REFRESH_INTERVAL).build());

    NewIndex.NewIndexType type = index.createType(FOO_TYPE)
      .requireProjectAuthorization();

    type.keywordFieldBuilder(FIELD_NAME).build();
    type.keywordFieldBuilder(FIELD_PROJECT_UUID).build();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;

import java.util.Arrays;
import java.util.stream.Stream;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.es.EsTester;

import static java.util.Arrays.asList;

public class PermissionIndexerTester {
```

```java
  private final PermissionIndexer permissionIndexer;

  public PermissionIndexerTester(EsTester esTester, NeedAuthorizationIndexer indexer,
NeedAuthorizationIndexer... others) {
    NeedAuthorizationIndexer[] indexers = Stream.concat(Stream.of(indexer),
Arrays.stream(others)).toArray(NeedAuthorizationIndexer[]::new);
    this.permissionIndexer = new PermissionIndexer(null, esTester.client(), indexers);
  }

  public PermissionIndexerTester allowOnlyAnyone(ComponentDto project) {
    PermissionIndexerDao.Dto dto = new PermissionIndexerDao.Dto(project.uuid(), project.qualifier());
    dto.allowAnyone();
    permissionIndexer.index(asList(dto));
    return this;
  }

  public PermissionIndexerTester allowOnlyUser(ComponentDto project, UserDto user) {
    PermissionIndexerDao.Dto dto = new PermissionIndexerDao.Dto(project.uuid(), project.qualifier())
      .addUserId(user.getId());
    permissionIndexer.index(asList(dto));
    return this;
  }

  public PermissionIndexerTester allowOnlyGroup(ComponentDto project, GroupDto group) {
    PermissionIndexerDao.Dto dto = new PermissionIndexerDao.Dto(project.uuid(), project.qualifier())
      .addGroupId(group.getId());
    permissionIndexer.index(asList(dto));
    return this;
  }

  public PermissionIndexerTester allow(PermissionIndexerDao.Dto access) {
    permissionIndexer.index(asList(access));
    return this;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
```

```
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;

import com.google.common.collect.ImmutableMap;
import com.google.common.collect.ImmutableSet;
import java.util.Collection;
import java.util.Set;
import org.sonar.db.DbSession;
import org.sonar.db.es.EsQueueDto;
import org.sonar.server.es.EsClient;
import org.sonar.server.es.IndexType;
import org.sonar.server.es.IndexingResult;
import org.sonar.server.es.ProjectIndexer;

import static org.sonar.server.permission.index.FooIndexDefinition.INDEX_TYPE_FOO;

public class FooIndexer implements ProjectIndexer, NeedAuthorizationIndexer {

 private static final AuthorizationScope AUTHORIZATION_SCOPE = new
AuthorizationScope(INDEX_TYPE_FOO, p -> true);

 private final EsClient esClient;

 public FooIndexer(EsClient esClient) {
  this.esClient = esClient;
 }

 @Override
 public AuthorizationScope getAuthorizationScope() {
  return AUTHORIZATION_SCOPE;
 }

 @Override
 public void indexOnAnalysis(String branchUuid) {
  addToIndex(branchUuid, "bar");
  addToIndex(branchUuid, "baz");
 }

 @Override
 public Collection<EsQueueDto> prepareForRecovery(DbSession dbSession, Collection<String> projectUuids,
Cause cause) {
  throw new UnsupportedOperationException();
```

```java
  }

  private void addToIndex(String projectUuid, String name) {
    esClient.prepareIndex(INDEX_TYPE_FOO)
      .setRouting(projectUuid)
      .setParent(projectUuid)
      .setSource(ImmutableMap.of(
        FooIndexDefinition.FIELD_NAME, name,
        FooIndexDefinition.FIELD_PROJECT_UUID, projectUuid))
      .get();
  }

  @Override
  public void indexOnStartup(Set<IndexType> uninitializedIndexTypes) {
    throw new UnsupportedOperationException();
  }

  @Override
  public Set<IndexType> getIndexTypes() {
    return ImmutableSet.of(INDEX_TYPE_FOO);
  }

  @Override
  public IndexingResult index(DbSession dbSession, Collection<EsQueueDto> items) {
    throw new UnsupportedOperationException();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission.index;
```

```java
import java.util.Arrays;
import java.util.List;
import org.elasticsearch.index.query.QueryBuilders;
import org.elasticsearch.search.SearchHits;
import org.sonar.core.util.stream.MoreCollectors;
import org.sonar.server.es.EsClient;

import static org.sonar.server.permission.index.FooIndexDefinition.FOO_INDEX;
import static org.sonar.server.permission.index.FooIndexDefinition.FOO_TYPE;

public class FooIndex {

  private final EsClient esClient;
  private final AuthorizationTypeSupport authorizationTypeSupport;

  public FooIndex(EsClient esClient, AuthorizationTypeSupport authorizationTypeSupport) {
    this.esClient = esClient;
    this.authorizationTypeSupport = authorizationTypeSupport;
  }

  public boolean hasAccessToProject(String projectUuid) {
    SearchHits hits = esClient.prepareSearch(FOO_INDEX)
      .setTypes(FOO_TYPE)
      .setQuery(QueryBuilders.boolQuery()
        .must(QueryBuilders.termQuery(FooIndexDefinition.FIELD_PROJECT_UUID, projectUuid))
        .filter(authorizationTypeSupport.createQueryFilter()))
      .get()
      .getHits();
    List<String> names = Arrays.stream(hits.hits())
      .map(h -> h.getSource().get(FooIndexDefinition.FIELD_NAME).toString())
      .collect(MoreCollectors.toList());
    return names.size() == 2 && names.contains("bar") && names.contains("baz");
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
```

```
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
import org.sonar.api.utils.System2;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.OrganizationPermission;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;

import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.QUALITY_GATE_ADMIN;
import static org.sonar.core.permission.GlobalPermissions.SCAN_EXECUTION;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonar.db.permission.OrganizationPermission.SCAN;
import static org.sonar.server.permission.PermissionChange.Operation.ADD;
import static org.sonar.server.permission.PermissionChange.Operation.REMOVE;

public class UserPermissionChangerTest {
 @Rule
 public DbTester db = DbTester.create(System2.INSTANCE);

 @Rule
 public ExpectedException expectedException = ExpectedException.none();

 private UserPermissionChanger underTest = new UserPermissionChanger(db.getDbClient());
 private OrganizationDto org1;
 private OrganizationDto org2;
 private UserDto user1;
 private UserDto user2;
 private ComponentDto privateProject;
```

```
  private ComponentDto publicProject;

 @Before
 public void setUp() throws Exception {
   org1 = db.organizations().insert();
   org2 = db.organizations().insert();
   user1 = db.users().insertUser();
   user2 = db.users().insertUser();
   privateProject = db.components().insertPrivateProject(org1);
   publicProject = db.components().insertPublicProject(org1);
 }

 @Test
 public void apply_adds_any_organization_permission_to_user() {
   OrganizationPermission.all()
     .forEach(perm -> {
       UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), perm.getKey(), null,
UserId.from(user1));

       apply(change);

       assertThat(db.users().selectPermissionsOfUser(user1, org1)).contains(perm);
     });
 }

 @Test
 public void apply_removes_any_organization_permission_to_user() {
   // give ADMIN perm to user2 so that user1 is not the only one with this permission and it can be removed from
user1
   db.users().insertPermissionOnUser(org1, user2, OrganizationPermission.ADMINISTER);
   OrganizationPermission.all()
     .forEach(perm -> db.users().insertPermissionOnUser(org1, user1, perm));
   assertThat(db.users().selectPermissionsOfUser(user1, org1)).containsOnly(OrganizationPermission.values());

   OrganizationPermission.all()
     .forEach(perm -> {
       UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), perm.getKey(), null,
UserId.from(user1));

       apply(change);

       assertThat(db.users().selectPermissionsOfUser(user1, org1)).doesNotContain(perm);
     });
 }

 @Test
 public void apply_has_no_effect_when_adding_permission_USER_on_a_public_project() {
   UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), USER, new
```

```
    ProjectId(publicProject), UserId.from(user1));

    apply(change);

    assertThat(db.users().selectProjectPermissionsOfUser(user1, publicProject)).doesNotContain(USER);
  }

  @Test
  public void apply_has_no_effect_when_adding_permission_CODEVIEWER_on_a_public_project() {
    UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), CODEVIEWER, new
    ProjectId(publicProject), UserId.from(user1));

    apply(change);

    assertThat(db.users().selectProjectPermissionsOfUser(user1, publicProject)).doesNotContain(CODEVIEWER);
  }

  @Test
  public void apply_adds_permission_ADMIN_on_a_public_project() {
    applyAddsPermissionOnAPublicProject(ADMIN);
  }

  @Test
  public void apply_adds_permission_ISSUE_ADMIN_on_a_public_project() {
    applyAddsPermissionOnAPublicProject(ISSUE_ADMIN);
  }

  @Test
  public void apply_adds_permission_SCAN_EXECUTION_on_a_public_project() {
    applyAddsPermissionOnAPublicProject(SCAN_EXECUTION);
  }

  private void applyAddsPermissionOnAPublicProject(String permission) {
    UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), permission, new
    ProjectId(publicProject), UserId.from(user1));

    apply(change);

    assertThat(db.users().selectProjectPermissionsOfUser(user1, publicProject)).containsOnly(permission);
  }

  @Test
  public void apply_fails_with_BadRequestException_when_removing_permission_USER_from_a_public_project()
  {
    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), USER, new
    ProjectId(publicProject), UserId.from(user1));

    expectedException.expect(BadRequestException.class);
```

```java
    expectedException.expectMessage("Permission user can't be removed from a public component");

    apply(change);
  }

  @Test
  public void
apply_fails_with_BadRequestException_when_removing_permission_CODEVIEWER_from_a_public_project() {
    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), CODEVIEWER, new
ProjectId(publicProject), UserId.from(user1));

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission codeviewer can't be removed from a public component");

    apply(change);
  }

  @Test
  public void apply_removes_permission_ADMIN_from_a_public_project() {
    applyRemovesPermissionFromPublicProject(ADMIN);
  }

  @Test
  public void apply_removes_permission_ISSUE_ADMIN_from_a_public_project() {
    applyRemovesPermissionFromPublicProject(ISSUE_ADMIN);
  }

  @Test
  public void apply_removes_permission_SCAN_EXECUTION_from_a_public_project() {
    applyRemovesPermissionFromPublicProject(SCAN_EXECUTION);
  }

  private void applyRemovesPermissionFromPublicProject(String permission) {
    db.users().insertProjectPermissionOnUser(user1, permission, publicProject);
    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), permission, new
ProjectId(publicProject), UserId.from(user1));

    apply(change);

    assertThat(db.users().selectProjectPermissionsOfUser(user1, publicProject)).isEmpty();
  }

  @Test
  public void apply_adds_any_permission_to_a_private_project() {
    ProjectPermissions.ALL
      .forEach(permission -> {
        UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), permission, new
ProjectId(privateProject), UserId.from(user1));
```

```java
      apply(change);

      assertThat(db.users().selectProjectPermissionsOfUser(user1, privateProject)).contains(permission);
    });
  }

  @Test
  public void apply_removes_any_permission_from_a_private_project() {
    ProjectPermissions.ALL
      .forEach(permission -> db.users().insertProjectPermissionOnUser(user1, permission, privateProject));

    ProjectPermissions.ALL
      .forEach(permission -> {
        UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), permission, new
ProjectId(privateProject), UserId.from(user1));

        apply(change);

        assertThat(db.users().selectProjectPermissionsOfUser(user1, privateProject)).doesNotContain(permission);
      });
  }

  @Test
  public void add_global_permission_to_user() {
    UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), SCAN_EXECUTION, null,
UserId.from(user1));

    apply(change);

    assertThat(db.users().selectPermissionsOfUser(user1, org1)).containsOnly(SCAN);
    assertThat(db.users().selectPermissionsOfUser(user1, org2)).isEmpty();
    assertThat(db.users().selectProjectPermissionsOfUser(user1, privateProject)).isEmpty();
    assertThat(db.users().selectPermissionsOfUser(user2, org1)).isEmpty();
    assertThat(db.users().selectProjectPermissionsOfUser(user2, privateProject)).isEmpty();
  }

  @Test
  public void add_project_permission_to_user() {
    UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), ISSUE_ADMIN, new
ProjectId(privateProject), UserId.from(user1));
    apply(change);

    assertThat(db.users().selectPermissionsOfUser(user1, org1)).isEmpty();
    assertThat(db.users().selectProjectPermissionsOfUser(user1, privateProject)).contains(ISSUE_ADMIN);
    assertThat(db.users().selectPermissionsOfUser(user2, org1)).isEmpty();
    assertThat(db.users().selectProjectPermissionsOfUser(user2, privateProject)).isEmpty();
  }
```

```java
@Test
public void do_nothing_when_adding_global_permission_that_already_exists() {
  db.users().insertPermissionOnUser(org1, user1, ADMINISTER_QUALITY_GATES);

  UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(),
QUALITY_GATE_ADMIN, null, UserId.from(user1));
  apply(change);

  assertThat(db.users().selectPermissionsOfUser(user1, org1)).containsOnly(ADMINISTER_QUALITY_GATES);
}

@Test
public void fail_to_add_global_permission_on_project() {
  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Invalid project permission 'gateadmin'. Valid values are [admin, codeviewer,
issueadmin, scan, user]");

  UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(),
QUALITY_GATE_ADMIN, new ProjectId(privateProject), UserId.from(user1));
  apply(change);
}

@Test
public void fail_to_add_project_permission_on_organization() {
  expectedException.expect(BadRequestException.class);
  expectedException.expectMessage("Invalid global permission 'issueadmin'. Valid values are [admin, profileadmin,
gateadmin, scan, provisioning]");

  UserPermissionChange change = new UserPermissionChange(ADD, org1.getUuid(), ISSUE_ADMIN, null,
UserId.from(user1));
  apply(change);
}

@Test
public void remove_global_permission_from_user() {
  db.users().insertPermissionOnUser(org1, user1, QUALITY_GATE_ADMIN);
  db.users().insertPermissionOnUser(org1, user1, SCAN_EXECUTION);
  db.users().insertPermissionOnUser(org2, user1, QUALITY_GATE_ADMIN);
  db.users().insertPermissionOnUser(org1, user2, QUALITY_GATE_ADMIN);
  db.users().insertProjectPermissionOnUser(user1, ISSUE_ADMIN, privateProject);

  UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(),
QUALITY_GATE_ADMIN, null, UserId.from(user1));
  apply(change);

  assertThat(db.users().selectPermissionsOfUser(user1, org1)).containsOnly(SCAN);
  assertThat(db.users().selectPermissionsOfUser(user1, org2)).containsOnly(ADMINISTER_QUALITY_GATES);
```

```java
    assertThat(db.users().selectPermissionsOfUser(user2, org1)).containsOnly(ADMINISTER_QUALITY_GATES);
    assertThat(db.users().selectProjectPermissionsOfUser(user1, privateProject)).containsOnly(ISSUE_ADMIN);
  }

  @Test
  public void remove_project_permission_from_user() {
    ComponentDto project2 = db.components().insertPrivateProject(org1);
    db.users().insertPermissionOnUser(user1, ADMINISTER_QUALITY_GATES);
    db.users().insertProjectPermissionOnUser(user1, ISSUE_ADMIN, privateProject);
    db.users().insertProjectPermissionOnUser(user1, USER, privateProject);
    db.users().insertProjectPermissionOnUser(user2, ISSUE_ADMIN, privateProject);
    db.users().insertProjectPermissionOnUser(user1, ISSUE_ADMIN, project2);

    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), ISSUE_ADMIN, new
ProjectId(privateProject), UserId.from(user1));
    apply(change);

    assertThat(db.users().selectProjectPermissionsOfUser(user1, privateProject)).containsOnly(USER);
    assertThat(db.users().selectProjectPermissionsOfUser(user2, privateProject)).containsOnly(ISSUE_ADMIN);
    assertThat(db.users().selectProjectPermissionsOfUser(user1, project2)).containsOnly(ISSUE_ADMIN);
  }

  @Test
  public void do_not_fail_if_removing_a_global_permission_that_does_not_exist() {
    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(),
QUALITY_GATE_ADMIN, null, UserId.from(user1));
    apply(change);

    assertThat(db.users().selectPermissionsOfUser(user1, org1)).isEmpty();
  }

  @Test
  public void do_not_fail_if_removing_a_project_permission_that_does_not_exist() {
    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), ISSUE_ADMIN, new
ProjectId(privateProject), UserId.from(user1));
    apply(change);

    assertThat(db.users().selectProjectPermissionsOfUser(user1, privateProject)).isEmpty();
  }

  @Test
  public void fail_to_remove_admin_global_permission_if_no_more_admins() {
    db.users().insertPermissionOnUser(org1, user1, SYSTEM_ADMIN);

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Last user with permission 'admin'. Permission cannot be removed.");

    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(), SYSTEM_ADMIN,
```

```
null, UserId.from(user1));
    underTest.apply(db.getSession(), change);
  }

  @Test
  public void remove_admin_user_if_still_other_admins() {
    db.users().insertPermissionOnUser(org1, user1, ADMINISTER);
    GroupDto admins = db.users().insertGroup(org1, "admins");
    db.users().insertMember(admins, user2);
    db.users().insertPermissionOnGroup(admins, ADMINISTER);

    UserPermissionChange change = new UserPermissionChange(REMOVE, org1.getUuid(),
ADMINISTER.getKey(), null, UserId.from(user1));
    underTest.apply(db.getSession(), change);

    assertThat(db.users().selectPermissionsOfUser(user1, org1)).isEmpty();
  }

  private void apply(UserPermissionChange change) {
    underTest.apply(db.getSession(), change);
    db.commit();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.permission;

import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
```

```java
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
import org.sonar.core.permission.GlobalPermissions;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.db.DbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.GroupPermissionDto;
import org.sonar.db.permission.OrganizationPermission;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;
import org.sonar.server.exceptions.BadRequestException;
import org.sonar.server.usergroups.ws.GroupIdOrAnyone;

import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.fail;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;

public class GroupPermissionChangerTest {

  @Rule
  public DbTester db = DbTester.create(System2.INSTANCE);
  @Rule
  public ExpectedException expectedException = ExpectedException.none();

  private GroupPermissionChanger underTest = new GroupPermissionChanger(db.getDbClient());
  private OrganizationDto org;
  private GroupDto group;
  private ComponentDto privateProject;
  private ComponentDto publicProject;

  @Before
  public void setUp() throws Exception {
    org = db.organizations().insert();
    group = db.users().insertGroup(org, "a-group");
    privateProject = db.components().insertPrivateProject(org);
    publicProject = db.components().insertPublicProject(org);
  }

  @Test
  public void apply_adds_organization_permission_to_group() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);

    apply(new GroupPermissionChange(PermissionChange.Operation.ADD,
GlobalPermissions.QUALITY_GATE_ADMIN, null, groupId));
```

```java
    assertThat(db.users().selectGroupPermissions(group,
null)).containsOnly(GlobalPermissions.QUALITY_GATE_ADMIN);
  }

  @Test
  public void apply_adds_organization_permission_to_group_AnyOne() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

    apply(new GroupPermissionChange(PermissionChange.Operation.ADD,
GlobalPermissions.QUALITY_GATE_ADMIN, null, groupId));

    assertThat(db.users().selectAnyonePermissions(org,
null)).containsOnly(GlobalPermissions.QUALITY_GATE_ADMIN);
  }

  @Test
  public void
apply_fails_with_BadRequestException_when_adding_any_permission_to_group_AnyOne_on_private_project() {
    GroupIdOrAnyone anyOneGroupId = GroupIdOrAnyone.forAnyone(org.getUuid());
    ProjectPermissions.ALL
      .forEach(perm -> {
      try {
        apply(new GroupPermissionChange(PermissionChange.Operation.ADD, perm, new ProjectId(privateProject),
anyOneGroupId));
        fail("a BadRequestException should have been thrown");
      } catch (BadRequestException e) {
        assertThat(e).hasMessage("No permission can be granted to Anyone on a private component");
      }
    });
  }

  @Test
  public void apply_has_no_effect_when_removing_any_permission_to_group_AnyOne_on_private_project() {
    ProjectPermissions.ALL
      .forEach(this::unsafeInsertProjectPermissionOnAnyone);

    GroupIdOrAnyone anyOneGroupId = GroupIdOrAnyone.forAnyone(org.getUuid());
    ProjectPermissions.ALL
      .forEach(perm -> {
      apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, perm, new
ProjectId(privateProject), anyOneGroupId));

      assertThat(db.users().selectAnyonePermissions(org, privateProject)).contains(perm);
    });
  }

  @Test
  public void apply_adds_permission_USER_to_group_on_private_project() {
```

```java
  applyAddsPermissionToGroupOnPrivateProject(UserRole.USER);
}

@Test
public void apply_adds_permission_CODEVIEWER_to_group_on_private_project() {
  applyAddsPermissionToGroupOnPrivateProject(UserRole.CODEVIEWER);
}

@Test
public void apply_adds_permission_ADMIN_to_group_on_private_project() {
  applyAddsPermissionToGroupOnPrivateProject(UserRole.ADMIN);
}

@Test
public void apply_adds_permission_ISSUE_ADMIN_to_group_on_private_project() {
  applyAddsPermissionToGroupOnPrivateProject(UserRole.ISSUE_ADMIN);
}

@Test
public void apply_adds_permission_SCAN_EXECUTION_to_group_on_private_project() {
  applyAddsPermissionToGroupOnPrivateProject(GlobalPermissions.SCAN_EXECUTION);
}

private void applyAddsPermissionToGroupOnPrivateProject(String permission) {
  GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);

  apply(new GroupPermissionChange(PermissionChange.Operation.ADD, permission, new
ProjectId(privateProject), groupId));

  assertThat(db.users().selectGroupPermissions(group, null)).isEmpty();
  assertThat(db.users().selectGroupPermissions(group, privateProject)).containsOnly(permission);
}

@Test
public void apply_removes_permission_USER_from_group_on_private_project() {
  applyRemovesPermissionFromGroupOnPrivateProject(UserRole.USER);
}

@Test
public void apply_removes_permission_CODEVIEWER_from_group_on_private_project() {
  applyRemovesPermissionFromGroupOnPrivateProject(UserRole.CODEVIEWER);
}

@Test
public void apply_removes_permission_ADMIN_from_on_private_project() {
  applyRemovesPermissionFromGroupOnPrivateProject(UserRole.ADMIN);
}
```

```java
@Test
public void apply_removes_permission_ISSUE_ADMIN_from_on_private_project() {
  applyRemovesPermissionFromGroupOnPrivateProject(UserRole.ISSUE_ADMIN);
}

@Test
public void apply_removes_permission_SCAN_EXECUTION_from_on_private_project() {
  applyRemovesPermissionFromGroupOnPrivateProject(GlobalPermissions.SCAN_EXECUTION);
}

private void applyRemovesPermissionFromGroupOnPrivateProject(String permission) {
  GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);
  db.users().insertProjectPermissionOnGroup(group, permission, privateProject);

  apply(new GroupPermissionChange(PermissionChange.Operation.ADD, permission, new
ProjectId(privateProject), groupId));

  assertThat(db.users().selectGroupPermissions(group, privateProject)).containsOnly(permission);
}

@Test
public void apply_has_no_effect_when_adding_USER_permission_to_group_AnyOne_on_a_public_project() {
  GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

  apply(new GroupPermissionChange(PermissionChange.Operation.ADD, UserRole.USER, new
ProjectId(publicProject), groupId));

  assertThat(db.users().selectAnyonePermissions(org, publicProject)).isEmpty();
}

@Test
public void
apply_has_no_effect_when_adding_CODEVIEWER_permission_to_group_AnyOne_on_a_public_project() {
  GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

  apply(new GroupPermissionChange(PermissionChange.Operation.ADD, UserRole.CODEVIEWER, new
ProjectId(publicProject), groupId));

  assertThat(db.users().selectAnyonePermissions(org, publicProject)).isEmpty();
}

@Test
public void
apply_fails_with_BadRequestException_when_adding_permission_ADMIN_to_group_AnyOne_on_a_public_proje
ct() {
  GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

  expectedException.expect(BadRequestException.class);
```

```java
    expectedException.expectMessage("It is not possible to add the 'admin' permission to group 'Anyone'");

    apply(new GroupPermissionChange(PermissionChange.Operation.ADD, UserRole.ADMIN, new
ProjectId(publicProject), groupId));
  }

 @Test
 public void apply_adds_permission_ISSUE_ADMIN_to_group_AnyOne_on_a_public_project() {
   GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

   apply(new GroupPermissionChange(PermissionChange.Operation.ADD, UserRole.ISSUE_ADMIN, new
ProjectId(publicProject), groupId));

   assertThat(db.users().selectAnyonePermissions(org, publicProject)).containsOnly(UserRole.ISSUE_ADMIN);
  }

 @Test
 public void apply_adds_permission_SCAN_EXECUTION_to_group_AnyOne_on_a_public_project() {
   GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

   apply(new GroupPermissionChange(PermissionChange.Operation.ADD,
GlobalPermissions.SCAN_EXECUTION, new ProjectId(publicProject), groupId));

   assertThat(db.users().selectAnyonePermissions(org,
publicProject)).containsOnly(GlobalPermissions.SCAN_EXECUTION);
  }

 @Test
 public void
apply_fails_with_BadRequestException_when_removing_USER_permission_from_group_AnyOne_on_a_public_p
roject() {
   GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

   expectedException.expect(BadRequestException.class);
   expectedException.expectMessage("Permission user can't be removed from a public component");

   apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, UserRole.USER, new
ProjectId(publicProject), groupId));
  }

 @Test
 public void
apply_fails_with_BadRequestException_when_removing_CODEVIEWER_permission_from_group_AnyOne_on_a
_public_project() {
   GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());

   expectedException.expect(BadRequestException.class);
   expectedException.expectMessage("Permission codeviewer can't be removed from a public component");
```

```java
    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, UserRole.CODEVIEWER, new
ProjectId(publicProject), groupId));
  }

  @Test
  public void apply_removes_ADMIN_permission_from_group_AnyOne_on_a_public_project() {
    applyRemovesPermissionFromGroupAnyOneOnAPublicProject(UserRole.ADMIN);
  }

  @Test
  public void apply_removes_ISSUE_ADMIN_permission_from_group_AnyOne_on_a_public_project() {
    applyRemovesPermissionFromGroupAnyOneOnAPublicProject(UserRole.ISSUE_ADMIN);
  }

  @Test
  public void apply_removes_SCAN_EXECUTION_permission_from_group_AnyOne_on_a_public_project() {
    applyRemovesPermissionFromGroupAnyOneOnAPublicProject(GlobalPermissions.SCAN_EXECUTION);
  }

  private void applyRemovesPermissionFromGroupAnyOneOnAPublicProject(String permission) {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(org.getUuid());
    db.users().insertProjectPermissionOnAnyone(permission, publicProject);

    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, permission, new
ProjectId(publicProject), groupId));

    assertThat(db.users().selectAnyonePermissions(org, publicProject)).isEmpty();
  }

  @Test
  public void
apply_fails_with_BadRequestException_when_removing_USER_permission_from_a_group_on_a_public_project()
{
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission user can't be removed from a public component");

    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, UserRole.USER, new
ProjectId(publicProject), groupId));
  }

  @Test
  public void
apply_fails_with_BadRequestException_when_removing_CODEVIEWER_permission_from_a_group_on_a_public
_project() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);
```

```java
    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Permission codeviewer can't be removed from a public component");

    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, UserRole.CODEVIEWER, new
ProjectId(publicProject), groupId));
  }

  @Test
  public void add_permission_to_anyone() {
    OrganizationDto defaultOrganization = db.getDefaultOrganization();
    GroupIdOrAnyone groupId = GroupIdOrAnyone.forAnyone(defaultOrganization.getUuid());

    apply(new GroupPermissionChange(PermissionChange.Operation.ADD,
GlobalPermissions.QUALITY_GATE_ADMIN, null, groupId));

    assertThat(db.users().selectGroupPermissions(group, null)).isEmpty();
    assertThat(db.users().selectAnyonePermissions(defaultOrganization,
null)).containsOnly(GlobalPermissions.QUALITY_GATE_ADMIN);
  }

  @Test
  public void do_nothing_when_adding_permission_that_already_exists() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);
    db.users().insertPermissionOnGroup(group, ADMINISTER_QUALITY_GATES);

    apply(new GroupPermissionChange(PermissionChange.Operation.ADD,
ADMINISTER_QUALITY_GATES.getKey(), null, groupId));

    assertThat(db.users().selectGroupPermissions(group,
null)).containsOnly(ADMINISTER_QUALITY_GATES.getKey());
  }

  @Test
  public void fail_to_add_global_permission_but_SCAN_and_ADMIN_on_private_project() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);

    OrganizationPermission.all()
      .map(OrganizationPermission::getKey)
      .filter(perm -> !UserRole.ADMIN.equals(perm) && !GlobalPermissions.SCAN_EXECUTION.equals(perm))
      .forEach(perm -> {
       try {
         apply(new GroupPermissionChange(PermissionChange.Operation.ADD, perm, new ProjectId(privateProject),
groupId));
         fail("a BadRequestException should have been thrown for permission " + perm);
       } catch (BadRequestException e) {
         assertThat(e).hasMessage("Invalid project permission '" + perm + "'. Valid values are [admin, codeviewer,
issueadmin, scan, user]");
```

```java
      }
    });
  }

  @Test
  public void fail_to_add_global_permission_but_SCAN_and_ADMIN_on_public_project() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);

    OrganizationPermission.all()
      .map(OrganizationPermission::getKey)
      .filter(perm -> !UserRole.ADMIN.equals(perm) && !GlobalPermissions.SCAN_EXECUTION.equals(perm))
      .forEach(perm -> {
        try {
          apply(new GroupPermissionChange(PermissionChange.Operation.ADD, perm, new ProjectId(publicProject),
groupId));
          fail("a BadRequestException should have been thrown for permission " + perm);
        } catch (BadRequestException e) {
          assertThat(e).hasMessage("Invalid project permission '" + perm + "'. Valid values are [admin, codeviewer,
issueadmin, scan, user]");
        }
      });
  }

  @Test
  public void fail_to_add_project_permission_but_SCAN_and_ADMIN_on_global_group() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);

    ProjectPermissions.ALL
      .stream()
      .filter(perm -> !GlobalPermissions.SCAN_EXECUTION.equals(perm) &&
!OrganizationPermission.ADMINISTER.getKey().equals(perm))
      .forEach(permission -> {
        try {
          apply(new GroupPermissionChange(PermissionChange.Operation.ADD, permission, null, groupId));
          fail("a BadRequestException should have been thrown for permission " + permission);
        } catch (BadRequestException e) {
          assertThat(e).hasMessage("Invalid global permission '" + permission + "'. Valid values are [admin,
profileadmin, gateadmin, scan, provisioning]");
        }
      });
  }

  @Test
  public void remove_permission_from_group() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);
    db.users().insertPermissionOnGroup(group, ADMINISTER_QUALITY_GATES);
    db.users().insertPermissionOnGroup(group, PROVISION_PROJECTS);
```

```java
    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE,
ADMINISTER_QUALITY_GATES.getKey(), null, groupId));

    assertThat(db.users().selectGroupPermissions(group, null)).containsOnly(PROVISION_PROJECTS.getKey());
  }

  @Test
  public void remove_project_permission_from_group() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);
    db.users().insertPermissionOnGroup(group, ADMINISTER_QUALITY_GATES);
    db.users().insertProjectPermissionOnGroup(group, UserRole.ISSUE_ADMIN, privateProject);
    db.users().insertProjectPermissionOnGroup(group, UserRole.CODEVIEWER, privateProject);

    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, UserRole.ISSUE_ADMIN, new
ProjectId(privateProject), groupId));

    assertThat(db.users().selectGroupPermissions(group,
null)).containsOnly(ADMINISTER_QUALITY_GATES.getKey());
    assertThat(db.users().selectGroupPermissions(group, privateProject)).containsOnly(UserRole.CODEVIEWER);
  }

  @Test
  public void do_not_fail_if_removing_a_permission_that_does_not_exist() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);

    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, UserRole.ISSUE_ADMIN, new
ProjectId(privateProject), groupId));

    assertThat(db.users().selectGroupPermissions(group, null)).isEmpty();
    assertThat(db.users().selectGroupPermissions(group, privateProject)).isEmpty();
  }

  @Test
  public void fail_to_remove_admin_permission_if_no_more_admins() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);
    db.users().insertPermissionOnGroup(group, ADMINISTER);

    expectedException.expect(BadRequestException.class);
    expectedException.expectMessage("Last group with permission 'admin'. Permission cannot be removed.");

    underTest.apply(db.getSession(), new GroupPermissionChange(PermissionChange.Operation.REMOVE,
ADMINISTER.getKey(), null, groupId));
  }

  @Test
  public void remove_admin_group_if_still_other_admins() {
    GroupIdOrAnyone groupId = GroupIdOrAnyone.from(group);
    db.users().insertPermissionOnGroup(group, ADMINISTER);
```

```
    UserDto admin = db.users().insertUser();
    db.users().insertPermissionOnUser(org, admin, ADMINISTER);

    apply(new GroupPermissionChange(PermissionChange.Operation.REMOVE, ADMINISTER.getKey(), null,
groupId));

    assertThat(db.users().selectGroupPermissions(group, null)).isEmpty();
  }

  private void apply(GroupPermissionChange change) {
    underTest.apply(db.getSession(), change);
    db.commit();
  }

  private void unsafeInsertProjectPermissionOnAnyone(String perm) {
    GroupPermissionDto dto = new GroupPermissionDto()
      .setOrganizationUuid(privateProject.getOrganizationUuid())
      .setGroupId(null)
      .setRole(perm)
      .setResourceId(privateProject.getId());
    db.getDbClient().groupPermissionDao().insert(db.getSession(), dto);
    db.commit();
  }
}
```

GNU LESSER GENERAL PUBLIC LICENSE
              Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

 This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

 0. Additional Definitions.

 As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

 "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

 An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode

of using an interface provided by the Library.

 A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

 The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

 The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

 1. Exception to Section 3 of the GNU GPL.

 You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

 2. Conveying Modified Versions.

 If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

  a) under this License, provided that you make a good faith effort to
  ensure that, in the event an Application does not supply the
  function or data, the facility still operates, and performs
  whatever part of its purpose remains meaningful, or

  b) under the GNU GPL, with none of the additional permissions of
  this License applicable to that copy.

 3. Object Code Incorporating Material from Library Header Files.

 The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

  a) Give prominent notice with each copy of the object code that the

Library is used in it and that the Library and its use are
covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license
document.

4. Combined Works.

 You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

 a) Give prominent notice with each copy of the Combined Work that
 the Library is used in it and that the Library and its use are
 covered by this License.

 b) Accompany the Combined Work with a copy of the GNU GPL and this license
 document.

 c) For a Combined Work that displays copyright notices during
 execution, include the copyright notice for the Library among
 these notices, as well as a reference directing the user to the
 copies of the GNU GPL and this license document.

 d) Do one of the following:

    0) Convey the Minimal Corresponding Source under the terms of this
    License, and the Corresponding Application Code in a form
    suitable for, and under terms that permit, the user to
    recombine or relink the Application with a modified version of
    the Linked Version to produce a modified Combined Work, in the
    manner specified by section 6 of the GNU GPL for conveying
    Corresponding Source.

    1) Use a suitable shared library mechanism for linking with the
    Library. A suitable mechanism is one that (a) uses at run time
    a copy of the Library already present on the user's computer
    system, and (b) will operate properly with a modified version
    of the Library that is interface-compatible with the Linked
    Version.

 e) Provide Installation Information, but only if you would otherwise
 be required to provide such information under section 6 of the
 GNU GPL, and only to the extent that such information is
 necessary to install and execute a modified version of the
 Combined Work produced by recombining or relinking the

Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based
on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "mybatis-3-mapper.dtd">
```

```xml
<mapper namespace="org.sonar.db.permission.UserPermissionMapper">

 <select id="selectUserPermissionsByQueryAndUserIds" parameterType="map"
resultType="org.sonar.db.permission.UserPermissionDto">
   select
     u.id as userId,
     ur.organization_uuid as organizationUuid,
     ur.resource_id as componentId,
     ur.role as permission
   <include refid="sqlQueryJoins" />
   <where>
     u.id in <foreach collection="userIds" open="(" close=")" item="userId"
separator=",">#{userId,jdbcType=INTEGER}</foreach>
     <include refid="sqlQueryFilters" />
   </where>
 </select>

 <select id="selectUserIdsByQuery" parameterType="map" resultType="int">
   select
     distinct u.id, lower(u.name) as lowerName
     <include refid="sqlQueryJoins" />
     <where>
       <include refid="sqlQueryFilters" />
     </where>
   order by lowerName asc
 </select>

 <select id="countUsersByQuery" parameterType="map" resultType="int">
   select count(distinct(u.id))
   <include refid="sqlQueryJoins" />
   <where>
     <include refid="sqlQueryFilters" />
   </where>
 </select>

 <sql id="sqlQueryJoins">
   from users u
   left join user_roles ur on ur.user_id = u.id
   left join projects p on ur.resource_id = p.id
   inner join organization_members om on u.id=om.user_id and
om.organization_uuid=#{query.organizationUuid,jdbcType=VARCHAR}
 </sql>

 <sql id="sqlQueryFilters">
   and u.active = ${_true}
   <if test="query.searchQueryToSql != null">
     and (
     lower(u.name) like #{query.searchQueryToSqlLowercase,jdbcType=VARCHAR} ESCAPE '/'
```

```
    or u.email like #{query.searchQueryToSql,jdbcType=VARCHAR} ESCAPE '/'
    or u.login like #{query.searchQueryToSql,jdbcType=VARCHAR} ESCAPE '/')
  </if>
  <!-- filter rows with user permissions -->
  <if test="query.withAtLeastOnePermission()">
    and ur.organization_uuid = om.organization_uuid
    and ur.role is not null
    <if test="query.componentUuid==null">
      and ur.resource_id is null
    </if>
    <if test="query.componentUuid!=null">
      and p.uuid = #{query.componentUuid,jdbcType=VARCHAR}
    </if>
    <if test="query.permission!=null">
      and ur.role = #{query.permission,jdbcType=VARCHAR}
    </if>
  </if>
</sql>

<select id="selectGlobalPermissionsOfUser" parameterType="map" resultType="string">
  select ur.role
  from user_roles ur
  where
  ur.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
  ur.user_id = #{userId,jdbcType=INTEGER} and
  ur.resource_id is null
</select>

<select id="selectProjectPermissionsOfUser" parameterType="map" resultType="string">
  select ur.role
  from user_roles ur
  where
  ur.user_id = #{userId,jdbcType=INTEGER} and
  ur.resource_id = #{projectId,jdbcType=BIGINT}
</select>

<select id="countUsersByProjectPermission" resultType="org.sonar.db.permission.CountPerProjectPermission">
  select ur.resource_id as componentId, ur.role as permission, count(u.login) as count
  from users u
  inner join user_roles ur on ur.user_id = u.id
  inner join projects p on p.id = ur.resource_id
  where u.active = ${_true}
  and p.id in <foreach collection="projectIds" open="(" close=")" item="projectId"
separator=",">#{projectId}</foreach>
  group by ur.resource_id, ur.role
</select>

<select id="selectUserIdsWithPermissionOnProjectBut" resultType="Integer">
```

```
    select
      distinct ur1.user_id
    from
      user_roles ur1
    where
      ur1.resource_id = #{projectId,jdbcType=BIGINT}
      and role <> #{permission,jdbcType=VARCHAR}
      and not exists (
        select
          1
        from
          user_roles ur2
        where
          ur2.resource_id = ur1.resource_id
          and ur2.user_id = ur1.user_id
          and role = #{permission,jdbcType=VARCHAR}
      )
  </select>

  <insert id="insert" parameterType="org.sonar.db.permission.UserPermissionDto" useGeneratedKeys="false">
    insert into user_roles (
    organization_uuid,
    user_id,
    resource_id,
    role
    ) values (
    #{organizationUuid,jdbcType=VARCHAR},
    #{userId,jdbcType=INTEGER},
    #{componentId,jdbcType=BIGINT},
    #{permission,jdbcType=VARCHAR}
    )
  </insert>

  <delete id="deleteGlobalPermission" parameterType="map">
    delete from user_roles
    where
    role = #{permission,jdbcType=VARCHAR} and
    user_id = #{userId,jdbcType=INTEGER} and
    organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    resource_id is null
  </delete>

  <delete id="deleteProjectPermission" parameterType="map">
    delete from user_roles
    where
    role = #{permission,jdbcType=VARCHAR} and
    user_id = #{userId,jdbcType=INTEGER} and
    resource_id = #{projectId,jdbcType=BIGINT}
```

```xml
  </delete>

  <delete id="deleteProjectPermissions" parameterType="map">
   delete from user_roles
   where
   resource_id = #{projectId,jdbcType=BIGINT}
  </delete>

  <delete id="deleteProjectPermissionOfAnyUser" parameterType="map">
   delete from
    user_roles
   where
    resource_id = #{projectId,jdbcType=BIGINT}
    and role = #{permission,jdbcType=VARCHAR}
  </delete>

  <delete id="deleteByOrganization" parameterType="String">
   delete from
    user_roles
   where
    organization_uuid = #{organizationUuid,jdbcType=VARCHAR}
  </delete>

  <delete id="deleteOrganizationMemberPermissions" parameterType="map">
   delete from
    user_roles
   where
    organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    user_id = #{userId,jdbcType=INTEGER}
  </delete>

  <delete id="deleteByUserId" parameterType="int">
   DELETE FROM user_roles WHERE user_id=#{userId,jdbcType=INTEGER}
  </delete>
</mapper>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "mybatis-3-mapper.dtd">
<mapper namespace="org.sonar.db.permission.template.PermissionTemplateCharacteristicMapper">
 <sql id="columns">
  ptc.id,
  ptc.template_id as templateId,
  ptc.permission_key as permission,
  ptc.with_project_creator as withProjectCreator,
  ptc.created_at as createdAt,
  ptc.updated_at as updatedAt
 </sql>

 <select id="selectById" parameterType="long" resultType="PermissionTemplateCharacteristic">
```

```xml
    select
    <include refid="columns" />
    from perm_tpl_characteristics ptc
    where ptc.id=#{id,jdbcType=BIGINT}
  </select>

  <select id="selectByTemplateIds" parameterType="long" resultType="PermissionTemplateCharacteristic">
    select
    <include refid="columns" />
    from perm_tpl_characteristics ptc
    where
    ptc.template_id in
    <foreach collection="templateIds" open="(" close=")" item="templateId" separator=",">
      #{templateId}
    </foreach>
    order by id
  </select>

  <select id="selectByPermissionAndTemplateId" parameterType="map"
resultType="PermissionTemplateCharacteristic">
    select
    <include refid="columns" />
    from perm_tpl_characteristics ptc
    where ptc.template_id=#{templateId}
    and ptc.permission_key=#{permission}
    order by id
  </select>

  <insert id="insert" parameterType="PermissionTemplateCharacteristic" keyColumn="id"
useGeneratedKeys="true" keyProperty="id">
    insert into perm_tpl_characteristics(template_id, permission_key, with_project_creator, created_at, updated_at)
    values(#{templateId, jdbcType=BIGINT}, #{permission, jdbcType=VARCHAR}, #{withProjectCreator,
jdbcType=BOOLEAN}, #{createdAt, jdbcType=BIGINT}, #{updatedAt, jdbcType=BIGINT})
  </insert>

  <update id="update" parameterType="PermissionTemplateCharacteristic" useGeneratedKeys="false">
    update perm_tpl_characteristics set
    with_project_creator=#{withProjectCreator, jdbcType=BOOLEAN},
    updated_at=#{updatedAt, jdbcType=BIGINT}
    where id=#{id}
  </update>

  <delete id="deleteByTemplateId" parameterType="long">
    DELETE FROM perm_tpl_characteristics
    WHERE template_id = #{permissionTemplateId}
  </delete>

  <delete id="deleteByTemplateIds" parameterType="long">
```

```
      delete from
        perm_tpl_characteristics
      where
        template_id in
        <foreach collection="templateIds" open="(" close=")" item="templateId" separator=",">
          #{templateId}
        </foreach>
  </delete>
</mapper>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "mybatis-3-mapper.dtd">


<mapper namespace="org.sonar.db.permission.template.PermissionTemplateMapper">

  <insert id="insert" parameterType="PermissionTemplate" keyColumn="id" useGeneratedKeys="true"
keyProperty="id">
    INSERT INTO permission_templates (organization_uuid, name, kee, description, key_pattern, created_at,
updated_at)
    VALUES (
    #{organizationUuid,jdbcType=VARCHAR},
    #{name,jdbcType=VARCHAR},
    #{kee,jdbcType=VARCHAR},
    #{description,jdbcType=VARCHAR},
    #{keyPattern,jdbcType=VARCHAR},
    #{createdAt},
    #{updatedAt})
  </insert>

  <update id="update" parameterType="PermissionTemplate">
    UPDATE permission_templates
    SET name = #{name}, description = #{description}, key_pattern = #{keyPattern}, updated_at = #{updatedAt}
    WHERE id = #{id}
  </update>

  <delete id="deleteById" parameterType="long">
    DELETE FROM permission_templates
    WHERE id = #{templateId}
  </delete>

  <delete id="deleteByIds" parameterType="long">
    delete from
      permission_templates
    where
      id in <foreach collection="templateIds" open="(" close=")" item="templateId" separator=",">
        #{templateId,jdbcType=BIGINT}
      </foreach>
  </delete>
```

```xml
<delete id="deleteUserPermissionsByTemplateId" parameterType="long">
  delete from
    perm_templates_users
  where
    template_id = #{templateId,jdbcType=BIGINT}
</delete>

<delete id="deleteUserPermissionsByTemplateIds">
  delete from
    perm_templates_users
  where
    template_id in <foreach collection="templateIds" open="(" close=")" item="templateId" separator=",">
      #{templateId,jdbcType=BIGINT}
    </foreach>
</delete>

<delete id="deleteUserPermission" parameterType="PermissionTemplateUser">
  DELETE FROM perm_templates_users
  WHERE template_id = #{templateId}
  AND user_id = #{userId}
  AND permission_reference = #{permission}
</delete>

<delete id="deleteUserPermissionsByOrganization" parameterType="map">
  delete from perm_templates_users
  where
  user_id = #{userId,jdbcType=INTEGER}
  and template_id in (select id from permission_templates where
organization_uuid=#{organizationUuid,jdbcType=VARCHAR})
</delete>

<delete id="deleteUserPermissionsByUserId" parameterType="integer">
  delete from perm_templates_users
  where
  user_id = #{userId,jdbcType=INTEGER}
</delete>

<delete id="deleteGroupPermissionsByTemplateId" parameterType="long">
  delete from
    perm_templates_groups
  where
    template_id = #{templateId,jdbcType=BIGINT}
</delete>

<delete id="deleteGroupPermissionsByTemplateIds">
  delete from
    perm_templates_groups
  where
```

```
      template_id in <foreach collection="templateIds" open="(" close=")" item="templateId" separator=",">
        #{templateId,jdbcType=BIGINT}
      </foreach>
  </delete>

  <delete id="deleteGroupPermission" parameterType="PermissionTemplateGroup">
    DELETE FROM perm_templates_groups
    WHERE template_id = #{templateId}
    AND permission_reference = #{permission}
    AND
    <choose>
      <when test="groupId != null">
        group_id = #{groupId}
      </when>
      <otherwise>
        group_id IS NULL
      </otherwise>
    </choose>
  </delete>

  <insert id="insertUserPermission" parameterType="PermissionTemplateUser">
    INSERT INTO perm_templates_users (template_id, user_id, permission_reference, created_at, updated_at)
    VALUES (#{templateId}, #{userId}, #{permission}, #{createdAt}, #{updatedAt})
  </insert>

  <insert id="insertGroupPermission" parameterType="PermissionTemplateGroup">
    INSERT INTO perm_templates_groups (template_id, group_id, permission_reference, created_at, updated_at)
    VALUES (
    #{templateId,jdbcType=BIGINT},
    #{groupId,jdbcType=INTEGER},
    #{permission,jdbcType=VARCHAR},
    #{createdAt,jdbcType=TIMESTAMP},
    #{updatedAt,jdbcType=TIMESTAMP}
    )
  </insert>

  <delete id="deleteByGroupId" parameterType="int">
    delete from perm_templates_groups
    where group_id = #{groupId,jdbcType=INTEGER}
  </delete>

  <select id="selectUserLoginsByQueryAndTemplate" parameterType="map" resultType="string">
    SELECT u.login FROM
      (SELECT DISTINCT u.login AS login, u.name AS name
      <include refid="userLoginsByQueryAndTemplate"/>
      ) u
    ORDER BY u.name
  </select>
```

```xml
<select id="countUserLoginsByQueryAndTemplate" parameterType="map" resultType="int">
 SELECT count(1)
  FROM (
  SELECT DISTINCT u.login AS login, u.name AS name
  <include refid="userLoginsByQueryAndTemplate"/>) u
</select>

<sql id="userLoginsByQueryAndTemplate">
 FROM users u
 LEFT JOIN perm_templates_users ptu ON ptu.user_id=u.id AND ptu.template_id=#{templateId}
 INNER JOIN organization_members om ON u.id=om.user_id AND
om.organization_uuid=#{query.organizationUuid}
 <where>
  u.active = ${_true}
  <if test="query.getSearchQueryToSql() != null">
   AND lower(u.name) like #{query.searchQueryToSqlLowercase} ESCAPE '/'
  </if>
  <if test="query.withAtLeastOnePermission()">
   and ptu.permission_reference is not null
   <if test="query.getPermission()!=null">
    and ptu.permission_reference=#{query.permission}
   </if>
  </if>
 </where>
</sql>

<select id="selectGroupNamesByQueryAndTemplate" parameterType="map" resultType="string">
 SELECT DISTINCT groups.name, LOWER(groups.name), groups.group_id
 <include refid="groupNamesByQueryAndTemplate" />
 ORDER BY LOWER(groups.name), groups.name, groups.group_id
</select>

<select id="countGroupNamesByQueryAndTemplate" parameterType="map" resultType="int">
 SELECT COUNT(1)
 FROM (
  SELECT DISTINCT group_id
  <include refid="groupNamesByQueryAndTemplate" />) g
</select>

<sql id="groupNamesByQueryAndTemplate">
 FROM
 (SELECT
  g.id AS group_id,
  g.name AS name,
  ptg.permission_reference AS permission,
  ptg.template_id AS templateId
 FROM groups g
```

```xml
    LEFT JOIN perm_templates_groups ptg ON
      ptg.group_id=g.id
    where
      g.organization_uuid=#{query.organizationUuid,jdbcType=VARCHAR}
    UNION ALL
    SELECT
      0 AS group_id,
      'Anyone' AS name,
      ptg.permission_reference AS permission,
      ptg.template_id AS templateId
    FROM perm_templates_groups ptg
    <where>
      <if test="query.withAtLeastOnePermission()">
        AND ptg.group_id IS NULL
      </if>
    </where>
    ) groups
    <where>
      <if test="query.searchQueryToSql != null">
        AND LOWER(groups.name) LIKE #{query.searchQueryToSqlLowercase} ESCAPE '/'
      </if>
      <if test="query.withAtLeastOnePermission()">
        AND groups.permission IS NOT NULL
        AND groups.templateId=#{templateId}
        <if test="query.permission != null">
          AND groups.permission=#{query.permission}
        </if>
      </if>
    </where>
  </sql>

  <sql id="templateColumns">
    id, organization_uuid as organizationUuid, name, kee, description, key_pattern AS keyPattern, created_at AS
createdAt, updated_at AS updatedAt
  </sql>

  <select id="selectByUuid" parameterType="String" resultType="PermissionTemplate">
    SELECT
    <include refid="templateColumns"/>
    FROM permission_templates
    WHERE kee=#{uuid}
  </select>

  <select id="selectAll" parameterType="map" resultType="PermissionTemplate">
    select
    <include refid="templateColumns"/>
    from permission_templates
    where
```

```xml
          organization_uuid = #{organizationUuid,jdbcType=VARCHAR}
          <if test="upperCaseNameLikeSql != null">
            and upper(name) like #{upperCaseNameLikeSql} escape '/'
          </if>
          order by upper(name), name
        </select>

        <select id="selectByName" parameterType="map" resultType="PermissionTemplate">
          select
          <include refid="templateColumns"/>
          from permission_templates
          where
          organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
          upper(name) = #{name,jdbcType=VARCHAR}
        </select>

        <sql id="permissionTemplateUserColumns">
          ptu.id,
          ptu.template_id as templateId,
          ptu.permission_reference AS permission,
          ptu.user_id AS userId,
          u.name AS userName,
          u.login AS userLogin,
          ptu.created_at AS createdAt,
          ptu.updated_at AS updatedAt
        </sql>

       <select id="selectUserPermissionsByTemplateIdAndUserLogins" parameterType="Long"
       resultType="PermissionTemplateUser">
         SELECT
         <include refid="permissionTemplateUserColumns"/>
         FROM perm_templates_users ptu
         INNER JOIN users u ON u.id = ptu.user_id AND u.active = ${_true}
         <where>
          AND ptu.template_id = #{templateId}
          <if test="!logins.isEmpty()">
            AND u.login IN <foreach collection="logins" open="(" close=")" item="login" separator=",">
             #{login}
            </foreach>
          </if>
         </where>
       </select>

      <select id="selectGroupPermissionsByTemplateIdAndGroupNames" parameterType="Long"
      resultType="PermissionTemplateGroup">
        SELECT
          sub.id,
          sub.templateId,
```

```
      sub.permission,
      sub.groupId,
      sub.groupName,
      sub.createdAt,
      sub.updatedAt
    FROM (
    SELECT
      ptg.id,
      ptg.template_id as templateId,
      ptg.permission_reference AS permission,
      ptg.group_id AS groupId,
      g.name AS groupName,
      ptg.created_at as createdAt,
      ptg.updated_at as updatedAt
    FROM perm_templates_groups ptg
    INNER JOIN groups g ON
      g.id=ptg.group_id
    UNION ALL
      SELECT
        ptg.id,
        ptg.template_id as templateId,
        ptg.permission_reference AS permission,
        0 AS groupId,
        'Anyone' AS groupName,
        ptg.created_at as createdAt,
        ptg.updated_at as updatedAt
      FROM perm_templates_groups ptg
      WHERE ptg.group_id IS NULL
    ) sub
    <where>
      sub.templateId=#{templateId}
      <if test="!groups.isEmpty()">
      AND sub.groupName IN <foreach collection="groups" open="(" close=")" item="group" separator=",">
      #{group}
      </foreach>
      </if>
    </where>
</select>

<select id="selectPotentialPermissionsByUserIdAndTemplateId" parameterType="map" resultType="String">
  <if test="userId!=null">
    -- from template users
    select ptu.permission_reference as permission_key
    from perm_templates_users ptu
    <where>
      and ptu.user_id=#{userId}
      and ptu.template_id=#{templateId}
    </where>
```

```xml
        UNION
        -- from template groups except anyone group
        select ptg.permission_reference as permission_key
        from perm_templates_groups ptg
        inner join groups_users gu on ptg.group_id = gu.group_id
        <where>
          and gu.user_id=#{userId}
          and ptg.template_id=#{templateId}
        </where>
        UNION
        -- from template characteristics
        select ptc.permission_key as permission_key
        from perm_tpl_characteristics ptc
        <where>
          and with_project_creator = ${_true}
          and ptc.template_id = #{templateId}
        </where>
        UNION
    </if>
    -- from anyone group
    select ptg.permission_reference as permission_key
    from perm_templates_groups ptg
    where ptg.template_id=#{templateId}
    and ptg.group_id IS NULL
</select>

<select id="usersCountByTemplateIdAndPermission" parameterType="map"
        resultType="org.sonar.db.permission.template.CountByTemplateAndPermissionDto">
  SELECT ptu.template_id as templateId, ptu.permission_reference as permission, count(u.login) as count
  FROM users u
  INNER JOIN perm_templates_users ptu ON ptu.user_id=u.id
  AND ptu.template_id in
  <foreach collection="templateIds" open="(" close=")" item="id" separator=",">
    #{id}
  </foreach>
  <where>
    AND u.active = ${_true}
  </where>
  GROUP BY ptu.template_id, ptu.permission_reference
</select>

<select id="groupsCountByTemplateIdAndPermission" parameterType="map"
        resultType="org.sonar.db.permission.template.CountByTemplateAndPermissionDto">
  SELECT count(1) as count, permission, templateId
  FROM
  (SELECT g.name as name, ptg.permission_reference as permission, ptg.template_id as templateId
  FROM groups g
  INNER JOIN perm_templates_groups ptg ON ptg.group_id=g.id
```

```xml
  UNION
  -- Add Anyone group permission
  SELECT #{anyoneGroup} as name, ptg.permission_reference as permission, ptg.template_id as templateId
  FROM perm_templates_groups ptg
  <where>
    AND ptg.group_id IS NULL
  </where>
  ) groups
  <where>
    AND groups.templateId in
    <foreach collection="templateIds" open="(" close=")" item="id" separator=",">
      #{id}
    </foreach>
  </where>
  GROUP BY groups.permission, groups.templateId
</select>

<select id="countGroupsWithPermission" resultType="int" parameterType="map">
  select count(1)
  from perm_templates_groups ptg
  where ptg.template_id = #{templateId}
  and ptg.permission_reference = #{permission}
  and
  <if test="groupId == null">
    ptg.group_id is null
  </if>
  <if test="groupId != null">
    ptg.group_id = #{groupId}
  </if>
</select>

<select id="selectTemplateIdsByOrganization" resultType="Long">
  select
    id
  from
    permission_templates
  where
    organization_uuid = #{organizationUuid,jdbcType=VARCHAR}
</select>

<select id="selectAllGroupPermissionTemplatesByGroupId" parameterType="Long"
resultType="PermissionTemplateGroup">
  SELECT
    ptg.id,
    ptg.template_id as templateId,
    ptg.permission_reference AS permission,
    ptg.group_id AS groupId,
    g.name AS groupName,
```

```xml
      ptg.created_at as createdAt,
      ptg.updated_at as updatedAt
    FROM perm_templates_groups ptg
    INNER JOIN groups g ON g.id=ptg.group_id
    <where>
      ptg.group_id=#{groupId,jdbcType=INTEGER}
    </where>
  </select>
</mapper>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "mybatis-3-mapper.dtd">


<mapper namespace="org.sonar.db.permission.GroupPermissionMapper">


  <select id="groupsCountByProjectIdAndPermission" parameterType="map"
        resultType="org.sonar.db.permission.CountPerProjectPermission">
    SELECT
      count(1) as count,
      permission,
      componentId
    FROM (
      SELECT
        g.name as name,
        group_role.role as permission,
        group_role.resource_id as componentId
      FROM
        groups g
      INNER JOIN group_roles group_role ON
        group_role.group_id=g.id
      UNION
      -- Add Anyone group permission
      SELECT
        #{anyoneGroup} as name,
        group_role.role as permission,
        group_role.resource_id as componentId
      FROM
        group_roles group_role
      where
        group_role.group_id IS NULL
    ) groups
    where
      groups.componentId in
      <foreach collection="componentIds" open="(" close=")" item="id" separator=",">
        #{id,jdbcType=BIGINT}
      </foreach>
    GROUP BY
      groups.permission,
      groups.componentId
```

```
</select>

<select id="selectGroupNamesByQuery" parameterType="map" resultType="string">
  select distinct sub.name, lower(sub.name), sub.groupId
  <include refid="groupsByQuery" />
  order by lower(sub.name), sub.name, sub.groupId
</select>

<select id="countGroupsByQuery" parameterType="map" resultType="int">
  select count(1)
  from (
    select distinct sub.groupId
    <include refid="groupsByQuery" />) g
</select>

<sql id="groupsByQuery">
  from (
    select g.id as groupId, g.name as name, gr.role as permission, gr.resource_id as componentId, gr.id as id
    from groups g
    left join group_roles gr on g.id = gr.group_id
    where
    g.organization_uuid = #{query.organizationUuid,jdbcType=VARCHAR}

  union all

    select 0 as groupId, 'Anyone' as name, gr.role as permission, gr.resource_id as componentId, gr.id as id
    from group_roles gr
    <if test="query.withAtLeastOnePermission()">
      where
      gr.organization_uuid = #{query.organizationUuid,jdbcType=VARCHAR} and
      gr.group_id is null
    </if>

  ) sub
  left join projects p on sub.componentId = p.id
  <where>
    <if test="query.searchQueryToSql != null">
      and lower(sub.name) like #{query.searchQueryToSqlLowercase,jdbcType=VARCHAR} ESCAPE '/'
    </if>
    <!-- filter rows with group permissions -->
    <if test="query.withAtLeastOnePermission()">
      and sub.permission is not null
      <if test="query.componentUuid==null">
        and sub.componentId is null
      </if>
      <if test="query.componentUuid!=null">
        and p.uuid = #{query.componentUuid,jdbcType=VARCHAR}
      </if>
```

```xml
    <if test="query.permission!=null">
      and sub.permission = #{query.permission,jdbcType=VARCHAR}
    </if>
   </if>
  </where>
 </sql>

 <select id="selectByGroupIds" parameterType="map" resultType="GroupPermission">
   select sub.groupId as groupId, sub.componentId as resourceId, sub.permission as role, sub.organizationUuid as
organizationUuid
   from
    (
    select gr.group_id as groupId, gr.resource_id as componentId, gr.role as permission, g.name as name,
gr.organization_uuid as organizationUuid
     from group_roles gr
     inner join groups g ON g.id = gr.group_id
     where gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
     gr.group_id is not null

     union all

     select 0 as groupId, gr.resource_id as componentId, gr.role as permission, 'Anyone' as name,
gr.organization_uuid as organizationUuid
     from group_roles gr
     where
     gr.group_id is null and
     gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR}
     ) sub
   where
    sub.groupId in
    <foreach collection="groupIds" open="(" close=")" item="groupId" separator=",">
      #{groupId,jdbcType=INTEGER}
    </foreach>
    <if test="projectId != null">
      and sub.componentId=#{projectId,jdbcType=BIGINT}
    </if>
    <if test="projectId==null">
      and sub.componentId is null
    </if>
 </select>

 <select id="selectGlobalPermissionsOfGroup" parameterType="map" resultType="String">
   select gr.role
   from group_roles gr
   where
   gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
   gr.resource_id is null and
   <choose>
```

```
      <when test="groupId != null">
        gr.group_id = #{groupId,jdbcType=INTEGER}
      </when>
      <otherwise>
        gr.group_id is null
      </otherwise>
    </choose>
  </select>

  <select id="selectProjectPermissionsOfGroup" parameterType="map" resultType="String">
    select gr.role
    from group_roles gr
    where
    gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    gr.resource_id = #{projectId,jdbcType=BIGINT} and
    <choose>
      <when test="groupId != null">
        gr.group_id = #{groupId,jdbcType=INTEGER}
      </when>
      <otherwise>
        gr.group_id is null
      </otherwise>
    </choose>
  </select>

  <select id="selectAllPermissionsByGroupId" parameterType="map" resultType="GroupPermission">
    select gr.group_id as groupId, gr.resource_id as resourceId, gr.role as role, gr.organization_uuid as
organizationUuid
    from group_roles gr
    where gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR}
    and gr.group_id = #{groupId,jdbcType=INTEGER}
  </select>

  <select id="selectGroupIdsWithPermissionOnProjectBut" resultType="Integer">
    select
      distinct gr1.group_id
    from
      group_roles gr1
    where
      gr1.resource_id = #{projectId,jdbcType=BIGINT}
      and gr1.group_id is not null
      and not exists (
        select
          1
        from
          group_roles gr2
        where
          gr2.resource_id = gr1.resource_id
```

```
    and gr2.group_id = gr1.group_id
    and gr2.role = #{role,jdbcType=VARCHAR}
  )
</select>

<insert id="insert" parameterType="GroupPermission" keyColumn="id" useGeneratedKeys="false"
keyProperty="id">
  insert into group_roles (
  organization_uuid,
  group_id,
  resource_id,
  role
  ) values (
  #{organizationUuid,jdbcType=VARCHAR},
  #{groupId,jdbcType=INTEGER},
  #{resourceId,jdbcType=BIGINT},
  #{role,jdbcType=VARCHAR}
  )
</insert>

<delete id="deleteByRootComponentId" parameterType="long">
  delete from group_roles
  where resource_id=#{rootComponentId,jdbcType=BIGINT}
</delete>

<delete id="deleteByRootComponentIdAndGroupId">
  delete from
    group_roles
  where
    resource_id=#{rootComponentId,jdbcType=BIGINT}
    <choose>
      <when test="groupId != null">
        and group_id = #{groupId,jdbcType=INTEGER}
      </when>
      <otherwise>
        and group_id is null
      </otherwise>
    </choose>
</delete>

<delete id="deleteByRootComponentIdAndPermission">
  delete from
    group_roles
  where
    resource_id=#{rootComponentId,jdbcType=BIGINT}
    and role=#{permission,jdbcType=VARCHAR}
</delete>
```

```xml
<delete id="delete" parameterType="map">
  delete from group_roles
  where
  role=#{permission,jdbcType=VARCHAR} and
  organization_uuid=#{organizationUuid,jdbcType=VARCHAR} and
  <choose>
    <when test="rootComponentId != null">
      resource_id=#{rootComponentId,jdbcType=BIGINT}
    </when>
    <otherwise>
      resource_id is null
    </otherwise>
  </choose>
  and
  <choose>
    <when test="groupId != null">
      group_id=#{groupId,jdbcType=INTEGER}
    </when>
    <otherwise>
      group_id is null
    </otherwise>
  </choose>
</delete>

<delete id="deleteByOrganization" parameterType="String">
  delete from
    group_roles
  where
    organization_uuid=#{organizationUuid,jdbcType=VARCHAR}
</delete>

</mapper>
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "mybatis-3-mapper.dtd">

<mapper namespace="org.sonar.db.permission.AuthorizationMapper">

<select id="selectOrganizationPermissions" parameterType="map" resultType="string">
  select gr.role
  from group_roles gr
  inner join groups_users gu on gr.group_id=gu.group_id
  where
  gr.organization_uuid=#{organizationUuid,jdbcType=VARCHAR} and
  gr.resource_id is null and
  gu.user_id=#{userId,jdbcType=INTEGER}

  union
```

```
  select gr.role
  from group_roles gr
  where
  gr.organization_uuid=#{organizationUuid,jdbcType=VARCHAR} and
  gr.group_id is null and
  gr.resource_id is null

  union

  select ur.role
  from user_roles ur
  where
  ur.organization_uuid=#{organizationUuid,jdbcType=VARCHAR} and
  ur.user_id=#{userId,jdbcType=INTEGER}
  and ur.resource_id is null
</select>

<select id="selectOrganizationPermissionsOfAnonymous" parameterType="map" resultType="string">
  select gr.role
  from group_roles gr
  where
  gr.organization_uuid=#{organizationUuid,jdbcType=VARCHAR} and
  gr.resource_id is null and
  gr.group_id is null
</select>

<select id="countUsersWithGlobalPermissionExcludingGroup" parameterType="map" resultType="int">
  select count(1) from
  (
    select gu.user_id
    from groups_users gu
    inner join group_roles gr on gr.group_id = gu.group_id
    where
    gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    gr.role = #{permission,jdbcType=VARCHAR} and
    gr.resource_id is null and
    gr.group_id is not null and
    gr.group_id != #{excludedGroupId,jdbcType=INTEGER}

    union

    select ur.user_id
    from user_roles ur
    where
    ur.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    ur.resource_id is null and
    ur.role = #{permission,jdbcType=VARCHAR}
  ) remaining
```

```xml
  </select>

  <select id="countUsersWithGlobalPermissionExcludingUser" parameterType="map" resultType="int">
    select count(1) from
    (
    select gu.user_id
    from groups_users gu
    inner join group_roles gr on gr.group_id = gu.group_id
    where
    gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    gr.role = #{permission,jdbcType=VARCHAR} and
    gr.resource_id is null and
    gr.group_id is not null and
    gu.user_id != #{excludedUserId,jdbcType=INTEGER}

    union

    select ur.user_id
    from user_roles ur
    where
    ur.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    ur.resource_id is null and
    ur.role = #{permission,jdbcType=VARCHAR} and
    ur.user_id != #{excludedUserId,jdbcType=INTEGER}
    ) remaining
  </select>

  <select id="countUsersWithGlobalPermissionExcludingGroupMember" parameterType="map" resultType="int">
    select count(1) from
    (
    select gu.user_id
    from groups_users gu
    inner join group_roles gr on gr.group_id = gu.group_id
    where
    gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    gr.role = #{permission,jdbcType=VARCHAR} and
    gr.resource_id is null and
    gr.group_id is not null and
    (gu.group_id != #{groupId,jdbcType=INTEGER} or gu.user_id != #{userId,jdbcType=INTEGER})

    union

    select ur.user_id
    from user_roles ur
    where
    ur.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    ur.resource_id is null and
    ur.role = #{permission,jdbcType=VARCHAR}
```

```xml
    ) remaining
</select>

<select id="countUsersWithGlobalPermissionExcludingUserPermission" parameterType="map" resultType="int">
  select count(1) from
  (
  select gu.user_id
  from groups_users gu
  inner join group_roles gr on gr.group_id = gu.group_id
  where
  gr.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
  gr.role = #{permission,jdbcType=VARCHAR} and
  gr.resource_id is null and
  gr.group_id is not null

  union

  select ur.user_id
  from user_roles ur
  where
  ur.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
  ur.resource_id is null and
  ur.role = #{permission,jdbcType=VARCHAR} and
  ur.user_id != #{userId,jdbcType=INTEGER}
  ) remaining
</select>

<select id="selectOrganizationUuidsOfUserWithGlobalPermission" parameterType="map" resultType="String">
  select gr.organization_uuid
  from group_roles gr
  inner join groups_users gu on gr.group_id = gu.group_id
  where
  gr.role = #{permission,jdbcType=VARCHAR} and
  gr.resource_id is null and
  gr.group_id is not null and
  gu.user_id = #{userId,jdbcType=INTEGER}

  union

  select ur.organization_uuid
  from user_roles ur
  where
  ur.resource_id is null and
  ur.role = #{permission,jdbcType=VARCHAR} and
  ur.user_id = #{userId,jdbcType=INTEGER}
</select>

<select id="keepAuthorizedProjectIdsForUser" parameterType="map" resultType="long">
```

```
select
  gr.resource_id
from
  group_roles gr
where
  gr.role=#{role,jdbcType=VARCHAR}
  and (
    gr.group_id is null
    or exists (
      select
        1
      from
        groups_users gu
      where
        gu.user_id = #{userId, jdbcType=INTEGER}
        and gr.group_id = gu.group_id
    )
  )
  and <foreach collection="componentIds" open="(" close=")" item="element" index="index" separator=" or ">
      gr.resource_id=#{element,jdbcType=BIGINT}
      </foreach>

union

select
  p.id
from
  user_roles ur
inner join projects p on
  p.id = ur.resource_id
where
  ur.role=#{role,jdbcType=VARCHAR}
  and ur.user_id=#{userId,jdbcType=INTEGER}
  and <foreach collection="componentIds" open="(" close=")" item="element" index="index" separator=" or ">
    p.id=#{element,jdbcType=BIGINT}
  </foreach>

union

<include refid="sqlSelectPublicProjectsIfRole"/>
</select>

<select id="keepAuthorizedProjectIdsForAnonymous" parameterType="map" resultType="long">
select
  gr.resource_id
from
  group_roles gr
where
```

```
       gr.role=#{role,jdbcType=VARCHAR}
       and gr.group_id is null
       and <foreach collection="componentIds" open="(" close=")" item="element" index="index" separator=" or ">
           gr.resource_id=#{element,jdbcType=BIGINT}
         </foreach>

    union

    <include refid="sqlSelectPublicProjectsIfRole"/>
  </select>

  <sql id="sqlSelectPublicProjectsIfRole">
   select
   p.id
   from
   projects p
   where
   <foreach collection="componentIds" open="(" close=")" item="element" index="index" separator=" or ">
    p.id=#{element,jdbcType=BIGINT}
   </foreach>
   and p.private = ${_false}
   and #{role,jdbcType=VARCHAR} in ('user','codeviewer')
  </sql>

  <select id="keepAuthorizedProjectUuidsForUser" parameterType="map" resultType="String">
   select p.uuid
   from projects p
   inner join group_roles gr on p.id = gr.resource_id
   where
    gr.role = #{permission,jdbcType=VARCHAR}
    and (gr.group_id is null or exists (
      select 1 from groups_users gu
      where
       gu.user_id = #{userId, jdbcType=INTEGER}
       and gr.group_id = gu.group_id)
    )
    and p.uuid in <foreach collection="projectUuids" open="(" close=")" item="projectUuid" index="index"
separator=",">#{projectUuid,jdbcType=VARCHAR}</foreach>

    union

    select p.uuid
    from projects p
    inner join user_roles ur on p.id = ur.resource_id
    where
     ur.role=#{permission,jdbcType=VARCHAR}
     and ur.user_id=#{userId,jdbcType=INTEGER}
     and p.uuid in <foreach collection="projectUuids" open="(" close=")" item="projectUuid" index="index"
```

```
    separator=",">#{projectUuid,jdbcType=VARCHAR}</foreach>

    <if test="permission == 'user' or permission == 'codeviewer'">
      union

      select p.uuid
      from projects p
      where
        p.uuid in <foreach collection="projectUuids" open="(" close=")" item="projectUuid" index="index"
separator=",">#{projectUuid,jdbcType=VARCHAR}</foreach>
        and p.private = ${_false}
    </if>
  </select>

  <select id="keepAuthorizedProjectUuidsForAnonymous" parameterType="map" resultType="String">
    select p.uuid
    from projects p
    inner join group_roles gr on p.id = gr.resource_id
    where
      gr.role=#{permission,jdbcType=VARCHAR}
      and gr.group_id is null
      and p.uuid in <foreach collection="projectUuids" open="(" close=")" item="projectUuid" index="index"
separator=",">#{projectUuid,jdbcType=VARCHAR}</foreach>

    <if test="permission == 'user' or permission == 'codeviewer'">
      union

      select p.uuid
      from projects p
      where
      p.uuid in <foreach collection="projectUuids" open="(" close=")" item="projectUuid" index="index"
separator=",">#{projectUuid,jdbcType=VARCHAR}</foreach>
      and p.private = ${_false}
    </if>
  </select>

  <select id="keepAuthorizedUsersForRoleAndProject" parameterType="map" resultType="int">
    select
      gu.user_id
    from
      groups_users gu
    inner join group_roles gr on
      gr.group_id=gu.group_id
    where
      gr.resource_id=#{componentId,jdbcType=BIGINT}
      and gr.role=#{role,jdbcType=VARCHAR}
      and gu.user_id in
        <foreach collection="userIds" open="(" close=")" item="id" separator=",">
```

```
                    #{id,jdbcType=BIGINT}
                </foreach>

            union

            select
                ur.user_id
            from
                user_roles ur
            where
                ur.resource_id=#{componentId,jdbcType=BIGINT}
                and ur.role=#{role,jdbcType=VARCHAR}
                and ur.user_id IN
                    <foreach collection="userIds" open="(" close=")" item="id" separator=",">
                        #{id,jdbcType=BIGINT}
                    </foreach>

            union

            select
                u.id
            from
                users u
            where
                u.id in
                    <foreach collection="userIds" open="(" close=")" item="id" separator=",">
                        #{id,jdbcType=BIGINT}
                    </foreach>
                and exists (
                    select
                        1
                    from
                        projects p
                    where
                        p.id =#{componentId,jdbcType=BIGINT}
                        and p.private = ${_false}
                        and #{role,jdbcType=VARCHAR} in ('user','codeviewer')
                )
    </select>

    <select id="selectProjectPermissions" parameterType="map" resultType="String">
        select ur.role
        from user_roles ur
        inner join projects p on p.id = ur.resource_id
        where
            p.uuid = #{projectUuid,jdbcType=VARCHAR} and
            p.organization_uuid = ur.organization_uuid and
            ur.user_id = #{userId,jdbcType=BIGINT}
```

```
    union

    select gr.role
    from group_roles gr
    inner join groups_users gu on gr.group_id = gu.group_id
    inner join projects p on p.id = gr.resource_id
    where
      p.uuid = #{projectUuid,jdbcType=VARCHAR} and
      p.organization_uuid = gr.organization_uuid and
      gu.user_id = #{userId,jdbcType=BIGINT}

    union

    <include refid="sql_selectProjectPermissionsOfAnonymous"/>
  </select>

  <select id="selectProjectPermissionsOfAnonymous" parameterType="map" resultType="String">
    <include refid="sql_selectProjectPermissionsOfAnonymous"/>
  </select>

  <sql id="sql_selectProjectPermissionsOfAnonymous">
    select
      gr.role
    from
      group_roles gr
    inner join projects p on
      p.id = gr.resource_id
    where
      p.uuid = #{projectUuid,jdbcType=VARCHAR}
      and p.organization_uuid = gr.organization_uuid
      and gr.group_id is null
  </sql>

  <select id="selectLoginsWithGlobalPermission" parameterType="map" resultType="String">
    select u.login
    from user_roles ur
    inner join users u on u.id=ur.user_id
    where
    ur.role=#{permission,jdbcType=VARCHAR}
    and ur.resource_id is null

    union

    select u.login
    from group_roles gr
    inner join groups_users gu on gr.group_id = gu.group_id
    inner join users u on u.id=gu.user_id
```

```
            where
            gr.role = #{permission,jdbcType=VARCHAR} and
            gr.resource_id is null and
            gr.group_id is not null
    </select>

    <select id="keepAuthorizedLoginsOnProject" parameterType="map" resultType="String">
        SELECT u.login
        FROM users u
        INNER JOIN user_roles ur ON ur.user_id = u.id
        INNER JOIN projects p ON p.kee = #{projectKey,jdbcType=VARCHAR}
        WHERE
          ur.organization_uuid = p.organization_uuid
          AND ur.resource_id = p.id
          AND ur.role = #{permission,jdbcType=VARCHAR}
          AND u.login IN <foreach collection="logins" open="(" close=")" item="login"
separator=",">#{login}</foreach>

        UNION

        SELECT u.login
        FROM users u
        INNER JOIN projects p ON p.kee = #{projectKey,jdbcType=VARCHAR}
        INNER JOIN group_roles gr ON gr.organization_uuid = p.organization_uuid
        INNER JOIN groups_users gu ON gr.group_id = gu.group_id
        WHERE
          gu.user_id = u.id
          AND gr.role = #{permission,jdbcType=VARCHAR}
          AND u.login IN <foreach collection="logins" open="(" close=")" item="login"
separator=",">#{login}</foreach>

      <if test="permission == 'user' or permission == 'codeviewer'">
        UNION

        SELECT u.login
        FROM users u
        INNER JOIN projects p ON p.kee = #{projectKey,jdbcType=VARCHAR}
        WHERE
          p.private = ${_false}
          AND u.login IN <foreach collection="logins" open="(" close=")" item="login"
separator=",">#{login}</foreach>
      </if>
    </select>
</mapper>
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
```

@ParametersAreNonnullByDefault

package org.sonar.core.permission;

import javax.annotation.ParametersAreNonnullByDefault;

/*

* SonarQube

* Copyright (C) 2009-2018 SonarSource SA

* mailto:info AT sonarsource DOT com

*

package org.sonar.core.permission;

import com.google.common.base.Joiner;

import com.google.common.collect.ImmutableList;

import java.util.List;

/**

* Holds the constants representing the various global permissions that can be assigned to users & groups

*

* @deprecated replaced by enum {@link org.sonar.db.permission.OrganizationPermission}

```java
*/
@Deprecated
public final class GlobalPermissions {

  public static final String SYSTEM_ADMIN = "admin";
  public static final String QUALITY_PROFILE_ADMIN = "profileadmin";
  public static final String QUALITY_GATE_ADMIN = "gateadmin";
  public static final String SCAN_EXECUTION = "scan";
  public static final String PROVISIONING = "provisioning";

  /**
   * All the global permissions values, ordered from {@link #SYSTEM_ADMIN} to {@link #PROVISIONING}.
   */
  public static final List<String> ALL = ImmutableList.of(
    SYSTEM_ADMIN, QUALITY_PROFILE_ADMIN, QUALITY_GATE_ADMIN, SCAN_EXECUTION,
PROVISIONING);
  public static final String ALL_ON_ONE_LINE = Joiner.on(", ").join(GlobalPermissions.ALL);

  private GlobalPermissions() {
    // only static methods
  }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.core.permission;

import com.google.common.base.Joiner;
import com.google.common.collect.ImmutableList;
import com.google.common.collect.ImmutableSet;
import java.util.List;
```

```java
import java.util.Set;
import org.sonar.api.web.UserRole;

/**
* Holds the constants representing the various component permissions that can be assigned to users & groups
*/
public final class ProjectPermissions {
 /**
  * Permissions which are implicitly available for any user, any group and to group "AnyOne" on public
components.
  */
 public static final Set<String> PUBLIC_PERMISSIONS = ImmutableSet.of(UserRole.USER,
UserRole.CODEVIEWER);

 /**
  * All the component permissions values, ordered from {@link UserRole#USER} to {@link
GlobalPermissions#SCAN_EXECUTION}.
  */
 public static final List<String> ALL = ImmutableList.of(UserRole.ADMIN, UserRole.CODEVIEWER,
UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION, UserRole.USER);

 public static final String ALL_ON_ONE_LINE = Joiner.on(", ").join(ProjectPermissions.ALL);

 private ProjectPermissions() {
   // static constants only
 }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
package org.sonar.db.permission;
```

```java
import java.util.Locale;
import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
import javax.annotation.concurrent.Immutable;
import org.sonar.db.WildcardPosition;

import static com.google.common.base.MoreObjects.firstNonNull;
import static com.google.common.base.Preconditions.checkArgument;
import static java.util.Objects.requireNonNull;
import static org.apache.commons.lang.StringUtils.defaultIfBlank;
import static org.sonar.api.utils.Paging.offset;
import static org.sonar.db.DaoUtils.buildLikeValue;

/**
 * Query used to get users and groups permissions
 */
@Immutable
public class PermissionQuery {
  public static final int RESULTS_MAX_SIZE = 100;
  public static final int SEARCH_QUERY_MIN_LENGTH = 3;
  public static final int DEFAULT_PAGE_SIZE = 20;
  public static final int DEFAULT_PAGE_INDEX = 1;

  // filter: return only the users or groups that are members of the organization
  private final String organizationUuid;
  // filter: return only the users or groups who have this permission
  private final String permission;
  // filter on project, else filter org permissions
  private final String componentUuid;
  private final String template;

  // filter on login, email or name of users or groups
  private final String searchQuery;
  private final String searchQueryToSql;
  private final String searchQueryToSqlLowercase;

  // filter users or groups who have at least one permission. It does make
  // sense when the filter "permission" is set.
  private final boolean withAtLeastOnePermission;

  private final int pageSize;
  private final int pageOffset;

  private PermissionQuery(Builder builder) {
    this.organizationUuid = builder.organizationUuid;
    this.permission = builder.permission;
    this.withAtLeastOnePermission = builder.withAtLeastOnePermission;
    this.componentUuid = builder.componentUuid;
```

```java
    this.template = builder.template;
    this.searchQuery = builder.searchQuery;
    this.searchQueryToSql = builder.searchQuery == null ? null : buildLikeValue(builder.searchQuery,
WildcardPosition.BEFORE_AND_AFTER);
    this.searchQueryToSqlLowercase = searchQueryToSql == null ? null :
searchQueryToSql.toLowerCase(Locale.ENGLISH);
    this.pageSize = builder.pageSize;
    this.pageOffset = offset(builder.pageIndex, builder.pageSize);
  }

  public String getOrganizationUuid() {
    return organizationUuid;
  }

  @CheckForNull
  public String getPermission() {
    return permission;
  }

  public boolean withAtLeastOnePermission() {
    return withAtLeastOnePermission;
  }

  // TODO remove it, it should not be in the query, but set as a separate parameter
  @Deprecated
  public String template() {
    return template;
  }

  @CheckForNull
  public String getComponentUuid() {
    return componentUuid;
  }

  @CheckForNull
  public String getSearchQuery() {
    return searchQuery;
  }

  @CheckForNull
  public String getSearchQueryToSql() {
    return searchQueryToSql;
  }

  @CheckForNull
  public String getSearchQueryToSqlLowercase() {
    return searchQueryToSqlLowercase;
  }
```

```java
public int getPageSize() {
  return pageSize;
}

public int getPageOffset() {
  return pageOffset;
}

public static Builder builder() {
  return new Builder();
}

public static class Builder {
  private String permission;
  private String organizationUuid;
  private String componentUuid;
  private String template;
  private String searchQuery;
  private boolean withAtLeastOnePermission;

  private Integer pageIndex;
  private Integer pageSize;

  private Builder() {
    // enforce method constructor
  }

  public Builder setPermission(@Nullable String permission) {
    this.withAtLeastOnePermission = permission != null;
    this.permission = permission;
    return this;
  }

  public Builder setTemplate(@Nullable String template) {
    this.template = template;
    return this;
  }

  public Builder setComponentUuid(@Nullable String componentUuid) {
    this.componentUuid = componentUuid;
    return this;
  }

  public Builder setOrganizationUuid(String organizationUuid) {
    this.organizationUuid = organizationUuid;
    return this;
  }
```

```java
    public Builder setSearchQuery(@Nullable String s) {
      this.searchQuery = defaultIfBlank(s, null);
      return this;
    }

    public Builder setPageIndex(@Nullable Integer i) {
      this.pageIndex = i;
      return this;
    }

    public Builder setPageSize(@Nullable Integer i) {
      this.pageSize = i;
      return this;
    }

    public Builder withAtLeastOnePermission() {
      this.withAtLeastOnePermission = true;
      return this;
    }

    public PermissionQuery build() {
      requireNonNull(organizationUuid, "Organization UUID cannot be null");
      this.pageIndex = firstNonNull(pageIndex, DEFAULT_PAGE_INDEX);
      this.pageSize = firstNonNull(pageSize, DEFAULT_PAGE_SIZE);
      checkArgument(searchQuery == null || searchQuery.length() >= SEARCH_QUERY_MIN_LENGTH, "Search
query should contains at least %s characters", SEARCH_QUERY_MIN_LENGTH);
      return new PermissionQuery(this);
    }
  }
}
```
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,

```java
package org.sonar.db.permission.template;

import java.util.List;
import java.util.Optional;
import org.sonar.db.Dao;
import org.sonar.db.DbSession;

import static com.google.common.base.Preconditions.checkArgument;
import static java.util.Collections.emptyList;
import static java.util.Objects.requireNonNull;

public class PermissionTemplateCharacteristicDao implements Dao {

  public List<PermissionTemplateCharacteristicDto> selectByTemplateIds(DbSession dbSession, List<Long>
templateIds) {
    return templateIds.isEmpty() ? emptyList() : mapper(dbSession).selectByTemplateIds(templateIds);
  }

  public Optional<PermissionTemplateCharacteristicDto> selectByPermissionAndTemplateId(DbSession dbSession,
String permission, long templateId) {
    PermissionTemplateCharacteristicDto dto = mapper(dbSession).selectByPermissionAndTemplateId(permission,
templateId);
    return Optional.ofNullable(dto);
  }

  public PermissionTemplateCharacteristicDto insert(DbSession dbSession, PermissionTemplateCharacteristicDto
dto) {
    checkArgument(dto.getCreatedAt() != 0L && dto.getUpdatedAt() != 0L);
    mapper(dbSession).insert(dto);
    return dto;
  }

  public PermissionTemplateCharacteristicDto update(DbSession dbSession, PermissionTemplateCharacteristicDto
templatePermissionDto) {
    requireNonNull(templatePermissionDto.getId());
    mapper(dbSession).update(templatePermissionDto);
    return templatePermissionDto;
  }

  private static PermissionTemplateCharacteristicMapper mapper(DbSession dbSession) {
    return dbSession.getMapper(PermissionTemplateCharacteristicMapper.class);
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
```

package org.sonar.db.permission.template;

import java.util.List;
import org.apache.ibatis.annotations.Param;

public interface PermissionTemplateCharacteristicMapper {

 PermissionTemplateCharacteristicDto selectById(@Param("id") long id);

 List<PermissionTemplateCharacteristicDto> selectByTemplateIds(@Param("templateIds") List<Long>
templateId);

 PermissionTemplateCharacteristicDto selectByPermissionAndTemplateId(@Param("permission") String
permission, @Param("templateId") long templateId);

 void insert(PermissionTemplateCharacteristicDto templatePermissionDto);

 void update(PermissionTemplateCharacteristicDto templatePermissionDto);

 void deleteByTemplateId(long id);

 void deleteByTemplateIds(@Param("templateIds") List<Long> subList);
}

```
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.Date;
import javax.annotation.Nullable;

public class PermissionTemplateGroupDto {
  private Long id;
  private Long templateId;
  private Integer groupId;
  private String permission;
  private String groupName;
  private Date createdAt;
  private Date updatedAt;

  public Long getId() {
    return id;
  }

  public PermissionTemplateGroupDto setId(Long id) {
    this.id = id;
    return this;
  }

  public Long getTemplateId() {
    return templateId;
  }

  public PermissionTemplateGroupDto setTemplateId(Long templateId) {
    this.templateId = templateId;
    return this;
  }

  public Integer getGroupId() {
    return groupId;
  }

  public PermissionTemplateGroupDto setGroupId(@Nullable Integer groupId) {
    this.groupId = groupId;
    return this;
  }
```

```java
  public String getPermission() {
    return permission;
  }

  public PermissionTemplateGroupDto setPermission(String permission) {
    this.permission = permission;
    return this;
  }

  public String getGroupName() {
    return groupName;
  }

  public PermissionTemplateGroupDto setGroupName(String groupName) {
    this.groupName = groupName;
    return this;
  }

  public Date getCreatedAt() {
    return createdAt;
  }

  public PermissionTemplateGroupDto setCreatedAt(Date createdAt) {
    this.createdAt = createdAt;
    return this;
  }

  public Date getUpdatedAt() {
    return updatedAt;
  }

  public PermissionTemplateGroupDto setUpdatedAt(Date updatedAt) {
    this.updatedAt = updatedAt;
    return this;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
```

package org.sonar.db.permission.template;

import static com.google.common.base.Preconditions.checkArgument;

public class PermissionTemplateCharacteristicDto {

 private static final int MAX_PERMISSION_KEY_LENGTH = 64;

 private Long id;
 private long templateId;
 private String permission;
 private boolean withProjectCreator;
 private long createdAt;
 private long updatedAt;

 public Long getId() {
   return id;
 }

 public PermissionTemplateCharacteristicDto setId(Long id) {
   this.id = id;
   return this;
 }

 public long getTemplateId() {
   return templateId;
 }

 public PermissionTemplateCharacteristicDto setTemplateId(long templateId) {
   this.templateId = templateId;
   return this;
 }

 public String getPermission() {
   return permission;
 }

 public PermissionTemplateCharacteristicDto setPermission(String permission) {
   checkArgument(permission.length() <= MAX_PERMISSION_KEY_LENGTH, "Permission key length (%s) is
longer than the maximum authorized (%s). '%s' was provided.",

```java
      permission.length(), MAX_PERMISSION_KEY_LENGTH, permission);
    this.permission = permission;
    return this;
  }

  public boolean getWithProjectCreator() {
    return withProjectCreator;
  }

  public PermissionTemplateCharacteristicDto setWithProjectCreator(boolean withProjectCreator) {
    this.withProjectCreator = withProjectCreator;
    return this;
  }

  public long getCreatedAt() {
    return createdAt;
  }

  public PermissionTemplateCharacteristicDto setCreatedAt(long createdAt) {
    this.createdAt = createdAt;
    return this;
  }

  public long getUpdatedAt() {
    return updatedAt;
  }

  public PermissionTemplateCharacteristicDto setUpdatedAt(long updatedAt) {
    this.updatedAt = updatedAt;
    return this;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
```

```java
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.Collections;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
import org.apache.ibatis.session.ResultHandler;
import org.apache.ibatis.session.RowBounds;
import org.sonar.api.utils.System2;
import org.sonar.db.Dao;
import org.sonar.db.DbSession;
import org.sonar.db.permission.CountPerProjectPermission;
import org.sonar.db.permission.PermissionQuery;

import static java.lang.String.format;
import static org.sonar.api.security.DefaultGroups.ANYONE;
import static org.sonar.db.DatabaseUtils.executeLargeInputs;
import static org.sonar.db.DatabaseUtils.executeLargeInputsWithoutOutput;

public class PermissionTemplateDao implements Dao {

 private static final String ANYONE_GROUP_PARAMETER = "anyoneGroup";

 private final System2 system;

 public PermissionTemplateDao(System2 system) {
   this.system = system;
 }

 /**
  * @return a paginated list of user logins.
  */
 public List<String> selectUserLoginsByQueryAndTemplate(DbSession session, PermissionQuery query, long
templateId) {
   return mapper(session).selectUserLoginsByQueryAndTemplate(query, templateId, new
RowBounds(query.getPageOffset(), query.getPageSize()));
 }

 public int countUserLoginsByQueryAndTemplate(DbSession session, PermissionQuery query, long templateId) {
   return mapper(session).countUserLoginsByQueryAndTemplate(query, templateId);
 }
```

```java
public List<PermissionTemplateUserDto> selectUserPermissionsByTemplateIdAndUserLogins(DbSession
dbSession, long templateId, List<String> logins) {
  return executeLargeInputs(logins, l ->
mapper(dbSession).selectUserPermissionsByTemplateIdAndUserLogins(templateId, l));
 }

 public List<PermissionTemplateUserDto> selectUserPermissionsByTemplateId(DbSession dbSession, long
templateId) {
  return mapper(dbSession).selectUserPermissionsByTemplateIdAndUserLogins(templateId,
Collections.emptyList());
 }

 public List<String> selectGroupNamesByQueryAndTemplate(DbSession session, PermissionQuery query, long
templateId) {
  return mapper(session).selectGroupNamesByQueryAndTemplate(templateId, query, new
RowBounds(query.getPageOffset(), query.getPageSize()));
 }

 public int countGroupNamesByQueryAndTemplate(DbSession session, PermissionQuery query, String
organizationUuid, long templateId) {
  return mapper(session).countGroupNamesByQueryAndTemplate(organizationUuid, query, templateId);
 }

 public List<PermissionTemplateGroupDto> selectGroupPermissionsByTemplateIdAndGroupNames(DbSession
dbSession, long templateId, List<String> groups) {
  return executeLargeInputs(groups, g ->
mapper(dbSession).selectGroupPermissionsByTemplateIdAndGroupNames(templateId, g));
 }

 public List<PermissionTemplateGroupDto> selectGroupPermissionsByTemplateId(DbSession dbSession, long
templateId) {
  return mapper(dbSession).selectGroupPermissionsByTemplateIdAndGroupNames(templateId,
Collections.emptyList());
 }

 /**
  * @return {@code true} if template contains groups that are granted with {@code permission}, else {@code
false}
  */
 public boolean hasGroupsWithPermission(DbSession dbSession, long templateId, String permission, @Nullable
Integer groupId) {
  return mapper(dbSession).countGroupsWithPermission(templateId, permission, groupId) > 0;
 }

 @CheckForNull
 public PermissionTemplateDto selectByUuid(DbSession session, String templateUuid) {
  return mapper(session).selectByUuid(templateUuid);
```

```java
  }

  public List<PermissionTemplateDto> selectAll(DbSession session, String organizationUuid, @Nullable String
nameMatch) {
    String upperCaseNameLikeSql = nameMatch != null ? toUppercaseSqlQuery(nameMatch) : null;
    return mapper(session).selectAll(organizationUuid, upperCaseNameLikeSql);
  }

  private static String toUppercaseSqlQuery(String nameMatch) {
    String wildcard = "%";
    return format("%s%s%s", wildcard, nameMatch.toUpperCase(Locale.ENGLISH), wildcard);
  }

  public PermissionTemplateDto insert(DbSession session, PermissionTemplateDto dto) {
    mapper(session).insert(dto);
    return dto;
  }

  /**
   * Each row returns a #{@link CountPerProjectPermission}
   */
  public void usersCountByTemplateIdAndPermission(DbSession dbSession, List<Long> templateIds,
ResultHandler<CountByTemplateAndPermissionDto> resultHandler) {
    Map<String, Object> parameters = new HashMap<>(1);

    executeLargeInputsWithoutOutput(
      templateIds,
      partitionedTemplateIds -> {
        parameters.put("templateIds", partitionedTemplateIds);
        mapper(dbSession).usersCountByTemplateIdAndPermission(parameters, resultHandler);
      });
  }

  /**
   * Each row returns a #{@link CountPerProjectPermission}
   */
  public void groupsCountByTemplateIdAndPermission(DbSession dbSession, List<Long> templateIds,
ResultHandler<CountByTemplateAndPermissionDto> resultHandler) {
    Map<String, Object> parameters = new HashMap<>(2);
    parameters.put(ANYONE_GROUP_PARAMETER, ANYONE);

    executeLargeInputsWithoutOutput(
      templateIds,
      partitionedTemplateIds -> {
        parameters.put("templateIds", partitionedTemplateIds);
        mapper(dbSession).groupsCountByTemplateIdAndPermission(parameters, resultHandler);
      });
  }
```

```java
public List<PermissionTemplateGroupDto> selectAllGroupPermissionTemplatesByGroupId(DbSession dbSession,
long groupId) {
  return mapper(dbSession).selectAllGroupPermissionTemplatesByGroupId(groupId);
}

public void deleteById(DbSession session, long templateId) {
  PermissionTemplateMapper mapper = mapper(session);
  mapper.deleteUserPermissionsByTemplateId(templateId);
  mapper.deleteGroupPermissionsByTemplateId(templateId);
  session.getMapper(PermissionTemplateCharacteristicMapper.class).deleteByTemplateId(templateId);
  mapper.deleteById(templateId);
}

public PermissionTemplateDto update(DbSession session, PermissionTemplateDto permissionTemplate) {
  mapper(session).update(permissionTemplate);
  return permissionTemplate;
}

public void insertUserPermission(DbSession session, Long templateId, Integer userId, String permission) {
  PermissionTemplateUserDto permissionTemplateUser = new PermissionTemplateUserDto()
    .setTemplateId(templateId)
    .setUserId(userId)
    .setPermission(permission)
    .setCreatedAt(now())
    .setUpdatedAt(now());

  mapper(session).insertUserPermission(permissionTemplateUser);
  session.commit();
}

public void deleteUserPermission(DbSession session, Long templateId, Integer userId, String permission) {
  PermissionTemplateUserDto permissionTemplateUser = new PermissionTemplateUserDto()
    .setTemplateId(templateId)
    .setPermission(permission)
    .setUserId(userId);
  mapper(session).deleteUserPermission(permissionTemplateUser);
  session.commit();
}

public void deleteUserPermissionsByOrganization(DbSession dbSession, String organizationUuid, int userId) {
  mapper(dbSession).deleteUserPermissionsByOrganization(organizationUuid, userId);
}

public void deleteUserPermissionsByUserId(DbSession dbSession, int userId) {
  mapper(dbSession).deleteUserPermissionsByUserId(userId);
}
```

```java
 public void insertGroupPermission(DbSession session, long templateId, @Nullable Integer groupId, String
permission) {
   PermissionTemplateGroupDto permissionTemplateGroup = new PermissionTemplateGroupDto()
     .setTemplateId(templateId)
     .setPermission(permission)
     .setGroupId(groupId)
     .setCreatedAt(now())
     .setUpdatedAt(now());
   mapper(session).insertGroupPermission(permissionTemplateGroup);
 }

 public void insertGroupPermission(DbSession session, PermissionTemplateGroupDto permissionTemplateGroup)
{
   mapper(session).insertGroupPermission(permissionTemplateGroup);
 }

 public void deleteGroupPermission(DbSession session, Long templateId, @Nullable Integer groupId, String
permission) {
   PermissionTemplateGroupDto permissionTemplateGroup = new PermissionTemplateGroupDto()
     .setTemplateId(templateId)
     .setPermission(permission)
     .setGroupId(groupId);
   mapper(session).deleteGroupPermission(permissionTemplateGroup);
   session.commit();
 }

 public PermissionTemplateDto selectByName(DbSession dbSession, String organizationUuid, String name) {
   return mapper(dbSession).selectByName(organizationUuid, name.toUpperCase(Locale.ENGLISH));
 }

 public List<String> selectPotentialPermissionsByUserIdAndTemplateId(DbSession dbSession, @Nullable Integer
currentUserId, long templateId) {
   return mapper(dbSession).selectPotentialPermissionsByUserIdAndTemplateId(currentUserId, templateId);
 }

 /**
  * Remove a group from all templates (used when removing a group)
  */
 public void deleteByGroup(DbSession session, int groupId) {
   session.getMapper(PermissionTemplateMapper.class).deleteByGroupId(groupId);
 }

 private Date now() {
   return new Date(system.now());
 }

 private static PermissionTemplateMapper mapper(DbSession session) {
   return session.getMapper(PermissionTemplateMapper.class);
```

```java
  }

  public void deleteByOrganization(DbSession dbSession, String organizationUuid) {
    PermissionTemplateMapper templateMapper = mapper(dbSession);
    PermissionTemplateCharacteristicMapper templateCharacteristicMapper =
dbSession.getMapper(PermissionTemplateCharacteristicMapper.class);
    List<Long> templateIds = templateMapper.selectTemplateIdsByOrganization(organizationUuid);
    executeLargeInputsWithoutOutput(templateIds, subList -> {
      templateCharacteristicMapper.deleteByTemplateIds(subList);
      templateMapper.deleteGroupPermissionsByTemplateIds(subList);
      templateMapper.deleteUserPermissionsByTemplateIds(subList);
      templateMapper.deleteByIds(subList);
    });
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.Date;

public class PermissionTemplateUserDto {
  private Long id;
  private Long templateId;
  private Integer userId;
  private String permission;
  private String userName;
  private String userLogin;
  private Date createdAt;
  private Date updatedAt;
```

```java
public Long getId() {
 return id;
}

public PermissionTemplateUserDto setId(Long id) {
 this.id = id;
 return this;
}

public Long getTemplateId() {
 return templateId;
}

public PermissionTemplateUserDto setTemplateId(Long templateId) {
 this.templateId = templateId;
 return this;
}

public Integer getUserId() {
 return userId;
}

public PermissionTemplateUserDto setUserId(Integer userId) {
 this.userId = userId;
 return this;
}

public String getUserName() {
 return userName;
}

public PermissionTemplateUserDto setUserName(String userName) {
 this.userName = userName;
 return this;
}

public String getUserLogin() {
 return userLogin;
}

public PermissionTemplateUserDto setUserLogin(String userLogin) {
 this.userLogin = userLogin;
 return this;
}

public String getPermission() {
 return permission;
}
```

```java
  public PermissionTemplateUserDto setPermission(String permission) {
    this.permission = permission;
    return this;
  }

  public Date getCreatedAt() {
    return createdAt;
  }

  public PermissionTemplateUserDto setCreatedAt(Date createdAt) {
    this.createdAt = createdAt;
    return this;
  }

  public Date getUpdatedAt() {
    return updatedAt;
  }

  public PermissionTemplateUserDto setUpdatedAt(Date updatedAt) {
    this.updatedAt = updatedAt;
    return this;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.Date;
import javax.annotation.CheckForNull;
import javax.annotation.Nullable;
```

```java
public class PermissionTemplateDto {

private Long id;
private String name;
private String organizationUuid;
private String uuid;
private String description;
private String keyPattern;
private Date createdAt;
private Date updatedAt;

public Long getId() {
  return id;
}

public PermissionTemplateDto setId(Long id) {
  this.id = id;
  return this;
}

public String getOrganizationUuid() {
  return organizationUuid;
}

public PermissionTemplateDto setOrganizationUuid(String s) {
  this.organizationUuid = s;
  return this;
}

public String getName() {
  return name;
}

public PermissionTemplateDto setName(String name) {
  this.name = name;
  return this;
}

/**
 * @deprecated since 5.2 use {@link #getUuid()}
 */
@Deprecated
public String getKee() {
  return uuid;
}

/**
```

```java
 * @deprecated since 5.2 use {@link #setUuid(String)}
 */
@Deprecated
public PermissionTemplateDto setKee(String kee) {
  this.uuid = kee;
  return this;
}

/**
 * @since 5.2 the kee column is a proper uuid. Before that it was build on the name + timestamp
 */
public String getUuid() {
  return uuid;
}

/**
 * @since 5.2 the kee column is a proper uuid. Before it was build on the name + timestamp
 */
public PermissionTemplateDto setUuid(String uuid) {
  this.uuid = uuid;
  return this;
}

@CheckForNull
public String getDescription() {
  return description;
}

public PermissionTemplateDto setDescription(@Nullable String description) {
  this.description = description;
  return this;
}

@CheckForNull
public String getKeyPattern() {
  return keyPattern;
}

public PermissionTemplateDto setKeyPattern(@Nullable String regexp) {
  this.keyPattern = regexp;
  return this;
}

public Date getCreatedAt() {
  return createdAt;
}

public PermissionTemplateDto setCreatedAt(Date createdAt) {
```

```java
    this.createdAt = createdAt;
    return this;
  }

  public Date getUpdatedAt() {
    return updatedAt;
  }

  public PermissionTemplateDto setUpdatedAt(Date updatedAt) {
    this.updatedAt = updatedAt;
    return this;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.List;
import java.util.Map;
import javax.annotation.Nullable;
import org.apache.ibatis.annotations.Param;
import org.apache.ibatis.session.ResultHandler;
import org.apache.ibatis.session.RowBounds;
import org.sonar.db.permission.PermissionQuery;

/**
 * @since 3.7
 */
public interface PermissionTemplateMapper {

  void insert(PermissionTemplateDto permissionTemplate);
```

void update(PermissionTemplateDto permissionTemplate);

void deleteById(long templateId);

void deleteByIds(@Param("templateIds") List<Long> templateIds);

void deleteUserPermissionsByTemplateId(long templateId);

void deleteUserPermissionsByTemplateIds(@Param("templateIds") List<Long> templateIds);

void deleteUserPermissionsByOrganization(@Param("organizationUuid") String organizationUuid, @Param("userId") int userId);

void deleteUserPermissionsByUserId(@Param("userId") int userId);

void deleteUserPermission(PermissionTemplateUserDto permissionTemplateUser);

void deleteGroupPermissionsByTemplateId(long templateId);

void deleteGroupPermissionsByTemplateIds(@Param("templateIds") List<Long> templateIds);

void deleteGroupPermission(PermissionTemplateGroupDto permissionTemplateGroup);

PermissionTemplateDto selectByUuid(String templateUuid);

List<PermissionTemplateUserDto> selectUserPermissionsByTemplateIdAndUserLogins(@Param("templateId") long templateId, @Param("logins") List<String> logins);

List<PermissionTemplateGroupDto> selectGroupPermissionsByTemplateIdAndGroupNames(@Param("templateId") long templateId, @Param("groups") List<String> groups);

void insertUserPermission(PermissionTemplateUserDto permissionTemplateUser);

void insertGroupPermission(PermissionTemplateGroupDto permissionTemplateGroup);

void deleteByGroupId(int groupId);

PermissionTemplateDto selectByName(@Param("organizationUuid") String organizationUuid, @Param("name") String name);

List<String> selectUserLoginsByQueryAndTemplate(@Param("query") PermissionQuery query, @Param("templateId") long templateId, RowBounds rowBounds);

int countUserLoginsByQueryAndTemplate(@Param("query") PermissionQuery query, @Param("templateId") long templateId);

```java
 List<String> selectGroupNamesByQueryAndTemplate(@Param("templateId") long templateId, @Param("query")
PermissionQuery query, RowBounds rowBounds);

 int countGroupNamesByQueryAndTemplate(@Param("organizationUuid") String organizationUuid,
@Param("query") PermissionQuery query, @Param("templateId") long templateId);

 List<PermissionTemplateDto> selectAll(@Param("organizationUuid") String organizationUuid, @Nullable
@Param("upperCaseNameLikeSql") String upperCaseNameLikeSql);

 void usersCountByTemplateIdAndPermission(Map<String, Object> parameters,
ResultHandler<CountByTemplateAndPermissionDto> resultHandler);

 void groupsCountByTemplateIdAndPermission(Map<String, Object> parameters,
ResultHandler<CountByTemplateAndPermissionDto> resultHandler);

 List<String> selectPotentialPermissionsByUserIdAndTemplateId(@Param("userId") @Nullable Integer
currentUserId, @Param("templateId") long templateId);

 int countGroupsWithPermission(@Param("templateId") long templateId, @Param("permission") String permission,
@Nullable @Param("groupId") Integer groupId);

 List<Long> selectTemplateIdsByOrganization(@Param("organizationUuid") String organizationUuid);

 List<PermissionTemplateGroupDto> selectAllGroupPermissionTemplatesByGroupId(@Param("groupId") Long
groupId);

}
```

package org.sonar.db.permission.template;

```java
public class CountByTemplateAndPermissionDto {
  private long templateId;
  private String permission;
  private int count;

  public long getTemplateId() {
    return templateId;
  }

  public void setTemplateId(long templateId) {
    this.templateId = templateId;
  }

  public String getPermission() {
    return permission;
  }

  public void setPermission(String permission) {
    this.permission = permission;
  }

  public int getCount() {
    return count;
  }

  public void setCount(int count) {
    this.count = count;
  }
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
*/
```

```java
package org.sonar.db.permission;

import javax.annotation.Nullable;

public class GroupPermissionDto {

 private String organizationUuid;
 private Integer groupId;
 private Long resourceId;
 private String role;

 public Integer getGroupId() {
   return groupId;
 }

 public String getOrganizationUuid() {
   return organizationUuid;
 }

 public GroupPermissionDto setOrganizationUuid(String s) {
   this.organizationUuid = s;
   return this;
 }

 /**
  * Null when Anyone
  */
 public GroupPermissionDto setGroupId(@Nullable Integer groupId) {
   this.groupId = groupId;
   return this;
 }

 @Nullable
 public Long getResourceId() {
   return resourceId;
 }

 public GroupPermissionDto setResourceId(@Nullable Long resourceId) {
   this.resourceId = resourceId;
   return this;
 }

 public String getRole() {
   return role;
 }

 public GroupPermissionDto setRole(String role) {
   this.role = role;
```

```java
    return this;
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import java.util.List;
import java.util.Map;
import java.util.Set;
import javax.annotation.Nullable;
import org.apache.ibatis.annotations.Param;
import org.apache.ibatis.session.ResultHandler;
import org.apache.ibatis.session.RowBounds;

public interface GroupPermissionMapper {

 List<String> selectGroupNamesByQuery(@Param("query") PermissionQuery query, RowBounds rowBounds);

 int countGroupsByQuery(@Param("query") PermissionQuery query);

 List<GroupPermissionDto> selectByGroupIds(@Param("organizationUuid") String organizationUuid,
   @Param("groupIds") List<Integer> groupIds, @Nullable @Param("projectId") Long projectId);

 void groupsCountByProjectIdAndPermission(Map<String, Object> parameters, ResultHandler resultHandler);

 void insert(GroupPermissionDto dto);

 void delete(@Param("permission") String permission, @Param("organizationUuid") String organizationUuid,
   @Nullable @Param("groupId") Integer groupId, @Nullable @Param("rootComponentId") Long
rootComponentId);
```

```java
  List<String> selectGlobalPermissionsOfGroup(@Param("organizationUuid") String organizationUuid,
    @Nullable @Param("groupId") Integer groupId);

  List<String> selectProjectPermissionsOfGroup(@Param("organizationUuid") String organizationUuid,
    @Nullable @Param("groupId") Integer groupId, @Param("projectId") long projectId);

  void selectAllPermissionsByGroupId(@Param("organizationUuid") String organizationUuid,
    @Param("groupId") Integer groupId, ResultHandler resultHandler);

  /**
   * Lists id of groups with at least one permission on the specified root component but which do not have the specified
   * permission, <strong>excluding group "AnyOne"</strong> (which implies the returned {@code Set} can't contain
   * {@code null}).
   */
  Set<Integer> selectGroupIdsWithPermissionOnProjectBut(@Param("projectId") long projectId, @Param("role")
String permission);

  void deleteByOrganization(@Param("organizationUuid") String organizationUuid);

  void deleteByRootComponentId(@Param("rootComponentId") long componentId);

  int deleteByRootComponentIdAndGroupId(@Param("rootComponentId") long rootComponentId, @Nullable
@Param("groupId") Integer groupId);

  int deleteByRootComponentIdAndPermission(@Param("rootComponentId") long rootComponentId,
@Param("permission") String permission);
}
```

```java
package org.sonar.db.permission;

import java.util.Collection;
import java.util.List;
import java.util.Set;
import org.apache.ibatis.annotations.Param;

public interface UserPermissionMapper {

  List<UserPermissionDto> selectUserPermissionsByQueryAndUserIds(@Param("query") PermissionQuery query,
  @Param("userIds") Collection<Integer> userIds);

  List<Integer> selectUserIdsByQuery(@Param("query") PermissionQuery query);

  /**
   * Count the number of distinct users returned by {@link #selectUserIdsByQuery(PermissionQuery)}
   * {@link PermissionQuery#getPageOffset()} and {@link PermissionQuery#getPageSize()} are ignored.
   */
  int countUsersByQuery(@Param("query") PermissionQuery query);

  /**
   * Count the number of users per permission for a given list of projects.
   * @param projectIds a non-null and non-empty list of project ids
   */
  List<CountPerProjectPermission> countUsersByProjectPermission(@Param("projectIds") List<Long> projectIds);

  /**
   * select id of users with at least one permission on the specified project but which do not have the specified
   permission.
   */
  Set<Integer> selectUserIdsWithPermissionOnProjectBut(@Param("projectId") long projectId,
  @Param("permission") String permission);

  void insert(UserPermissionDto dto);

  void deleteGlobalPermission(@Param("userId") int userId, @Param("permission") String permission,
    @Param("organizationUuid") String organizationUuid);

  void deleteProjectPermission(@Param("userId") int userId, @Param("permission") String permission,
    @Param("projectId") long projectId);

  void deleteProjectPermissions(@Param("projectId") long projectId);

  int deleteProjectPermissionOfAnyUser(@Param("projectId") long projectId, @Param("permission") String
  permission);

  List<String> selectGlobalPermissionsOfUser(@Param("userId") int userId, @Param("organizationUuid") String
  organizationUuid);
```

List<String> selectProjectPermissionsOfUser(@Param("userId") int userId, @Param("projectId") long projectId);

void deleteByOrganization(@Param("organizationUuid") String organizationUuid);

void deleteOrganizationMemberPermissions(@Param("organizationUuid") String organizationUuid, @Param("userId") int login);

void deleteByUserId(@Param("userId") int userId);
}
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
*/
package org.sonar.db.permission;

import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Set;
import javax.annotation.Nullable;
import org.apache.ibatis.session.ResultHandler;
import org.apache.ibatis.session.RowBounds;
import org.sonar.api.security.DefaultGroups;
import org.sonar.db.Dao;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentMapper;
import org.sonar.db.user.GroupMapper;

import static com.google.common.base.Preconditions.checkArgument;
import static org.sonar.db.DatabaseUtils.executeLargeInputs;
import static org.sonar.db.DatabaseUtils.executeLargeInputsWithoutOutput;

```java
public class GroupPermissionDao implements Dao {

  private static final String ANYONE_GROUP_PARAMETER = "anyoneGroup";

  /**
   * Returns the names of the groups that match the given query, for the given organization.
   * The virtual group "Anyone" may be returned as the value {@link DefaultGroups#ANYONE}.
   * @return group names, sorted in alphabetical order
   */
  public List<String> selectGroupNamesByQuery(DbSession dbSession, PermissionQuery query) {
    return mapper(dbSession).selectGroupNamesByQuery(query, new RowBounds(query.getPageOffset(),
query.getPageSize()));
  }

  /**
   * Count the number of groups returned by {@link #selectGroupNamesByQuery(DbSession, PermissionQuery)},
   * without applying pagination.
   */
  public int countGroupsByQuery(DbSession dbSession, PermissionQuery query) {
    return mapper(dbSession).countGroupsByQuery(query);
  }

  /**
   * Select global or project permission of given groups and organization. Anyone virtual group is supported
   * through the value "zero" (0L) in {@code groupIds}.
   */
  public List<GroupPermissionDto> selectByGroupIds(DbSession dbSession, String organizationUuid, List<Integer>
groupIds, @Nullable Long projectId) {
    return executeLargeInputs(groupIds, groups -> mapper(dbSession).selectByGroupIds(organizationUuid, groups,
projectId));
  }

  /**
   * Select global and project permissions of a given group (Anyone group is NOT supported)
   * Each row returns a {@link GroupPermissionDto}
   */
  public void selectAllPermissionsByGroupId(DbSession dbSession, String organizationUuid, Integer groupId,
ResultHandler resultHandler) {
    mapper(dbSession).selectAllPermissionsByGroupId(organizationUuid, groupId, resultHandler);
  }

  /**
   * Each row returns a {@link CountPerProjectPermission}
   */
  public void groupsCountByComponentIdAndPermission(DbSession dbSession, List<Long> componentIds,
ResultHandler resultHandler) {
    Map<String, Object> parameters = new HashMap<>(2);
    parameters.put(ANYONE_GROUP_PARAMETER, DefaultGroups.ANYONE);
```

```
    executeLargeInputsWithoutOutput(
      componentIds,
      partitionedComponentIds -> {
        parameters.put("componentIds", partitionedComponentIds);
        mapper(dbSession).groupsCountByProjectIdAndPermission(parameters, resultHandler);
      });
  }

  /**
   * Selects the global permissions granted to group. An empty list is returned if the
   * group does not exist.
   */
  public List<String> selectGlobalPermissionsOfGroup(DbSession session, String organizationUuid, @Nullable
Integer groupId) {
    return mapper(session).selectGlobalPermissionsOfGroup(organizationUuid, groupId);
  }

  /**
   * Selects the permissions granted to group and project. An empty list is returned if the
   * group or project do not exist.
   */
  public List<String> selectProjectPermissionsOfGroup(DbSession session, String organizationUuid, @Nullable
Integer groupId, long projectId) {
    return mapper(session).selectProjectPermissionsOfGroup(organizationUuid, groupId, projectId);
  }

  /**
   * Lists id of groups with at least one permission on the specified root component but which do not have the
specified
   * permission, <strong>excluding group "AnyOne"</strong> (which implies the returned {@code Sett} can't
contain
   * {@code null}).
   */
  public Set<Integer> selectGroupIdsWithPermissionOnProjectBut(DbSession session, long projectId, String
permission) {
    return mapper(session).selectGroupIdsWithPermissionOnProjectBut(projectId, permission);
  }

  public void insert(DbSession dbSession, GroupPermissionDto dto) {
    ensureComponentPermissionConsistency(dbSession, dto);
    ensureGroupPermissionConsistency(dbSession, dto);
    mapper(dbSession).insert(dto);
  }

  private static void ensureComponentPermissionConsistency(DbSession dbSession, GroupPermissionDto dto) {
    if (dto.getResourceId() == null) {
      return;
```

```java
    }
    ComponentMapper componentMapper = dbSession.getMapper(ComponentMapper.class);
    checkArgument(
      componentMapper.countComponentByOrganizationAndId(dto.getOrganizationUuid(), dto.getResourceId()) ==
1,
      "Can't insert permission '%s' for component with id '%s' in organization with uuid '%s' because this component
does not belong to organization with uuid '%s'",
      dto.getRole(), dto.getResourceId(), dto.getOrganizationUuid(), dto.getOrganizationUuid());
  }

  private static void ensureGroupPermissionConsistency(DbSession dbSession, GroupPermissionDto dto) {
    if (dto.getGroupId() == null) {
      return;
    }
    GroupMapper groupMapper = dbSession.getMapper(GroupMapper.class);
    checkArgument(
      groupMapper.countGroupByOrganizationAndId(dto.getOrganizationUuid(), dto.getGroupId()) == 1,
      "Can't insert permission '%s' for group with id '%s' in organization with uuid '%s' because this group does not
belong to organization with uuid '%s'",
      dto.getRole(), dto.getGroupId(), dto.getOrganizationUuid(), dto.getOrganizationUuid());
  }

  /**
   * Delete all the permissions associated to a root component (project)
   */
  public void deleteByRootComponentId(DbSession dbSession, long rootComponentId) {
    mapper(dbSession).deleteByRootComponentId(rootComponentId);
  }

  /**
   * Delete all permissions of the specified group (group "AnyOne" if {@code groupId} is {@code null}) for the
specified
   * component.
   */
  public int deleteByRootComponentIdAndGroupId(DbSession dbSession, long rootComponentId, @Nullable
Integer groupId) {
    return mapper(dbSession).deleteByRootComponentIdAndGroupId(rootComponentId, groupId);
  }

  /**
   * Delete the specified permission for the specified component for any group (including group AnyOne).
   */
  public int deleteByRootComponentIdAndPermission(DbSession dbSession, long rootComponentId, String
permission) {
    return mapper(dbSession).deleteByRootComponentIdAndPermission(rootComponentId, permission);
  }

  /**
```

```
 * Delete a single permission. It can be:
 * <ul>
 *   <li>a global permission granted to a group</li>
 *   <li>a global permission granted to anyone</li>
 *   <li>a permission granted to a group for a project</li>
 *   <li>a permission granted to anyone for a project</li>
 * </ul>
 * @param dbSession
 * @param permission the kind of permission
 * @param organizationUuid UUID of organization, even if parameter {@code groupId} is not null
 * @param groupId if null, then anyone, else id of group
 * @param rootComponentId if null, then global permission, else id of root component (project)
 */
public void delete(DbSession dbSession, String permission, String organizationUuid, @Nullable Integer groupId,
@Nullable Long rootComponentId) {
  mapper(dbSession).delete(permission, organizationUuid, groupId, rootComponentId);
}

public void deleteByOrganization(DbSession dbSession, String organizationUuid) {
  mapper(dbSession).deleteByOrganization(organizationUuid);
}

private static GroupPermissionMapper mapper(DbSession session) {
  return session.getMapper(GroupPermissionMapper.class);
}
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import javax.annotation.CheckForNull;
```

```java
import javax.annotation.Nullable;

public class UserPermissionDto {

 private String organizationUuid;
 private String permission;
 private int userId;
 private Long componentId;

 public UserPermissionDto() {
  // used by MyBatis
 }

 public UserPermissionDto(String organizationUuid, String permission, int userId, @Nullable Long componentId) {
  this.organizationUuid = organizationUuid;
  this.permission = permission;
  this.userId = userId;
  this.componentId = componentId;
 }

 public String getPermission() {
  return permission;
 }

 public int getUserId() {
  return userId;
 }

 public String getOrganizationUuid() {
  return organizationUuid;
 }

 /**
  * @return {@code null} if it's a global permission, else return the project id.
  */
 @CheckForNull
 public Long getComponentId() {
  return componentId;
 }

 @Override
 public String toString() {
  StringBuilder sb = new StringBuilder("UserPermissionDto{");
  sb.append("permission='").append(permission).append('\'');
  sb.append(", userId=").append(userId);
  sb.append(", organizationUuid=").append(organizationUuid);
  sb.append(", componentId=").append(componentId);
  sb.append('}');
```

```
    return sb.toString();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import java.util.Collection;
import java.util.List;
import java.util.Set;
import org.apache.ibatis.annotations.Param;

/**
 * @see AuthorizationDao
 */
public interface AuthorizationMapper {

 Set<String> selectOrganizationPermissions(@Param("organizationUuid") String organizationUuid,
@Param("userId") int userId);

 Set<String> selectOrganizationPermissionsOfAnonymous(@Param("organizationUuid") String organizationUuid);

 int countUsersWithGlobalPermissionExcludingGroup(@Param("organizationUuid") String organizationUuid,
   @Param("permission") String permission, @Param("excludedGroupId") int excludedGroupId);

 int countUsersWithGlobalPermissionExcludingUser(@Param("organizationUuid") String organizationUuid,
@Param("permission") String permission,
   @Param("excludedUserId") int excludedUserId);

 int countUsersWithGlobalPermissionExcludingGroupMember(@Param("organizationUuid") String
organizationUuid,
```

```java
    @Param("permission") String permission, @Param("groupId") int groupId, @Param("userId") int userId);

  int countUsersWithGlobalPermissionExcludingUserPermission(@Param("organizationUuid") String
organizationUuid,
    @Param("permission") String permission, @Param("userId") int userId);

  Set<String> selectOrganizationUuidsOfUserWithGlobalPermission(@Param("userId") int userId,
@Param("permission") String permission);

  Set<Long> keepAuthorizedProjectIdsForAnonymous(@Param("role") String role, @Param("componentIds")
Collection<Long> componentIds);

  Set<Long> keepAuthorizedProjectIdsForUser(@Param("userId") int userId, @Param("role") String role,
@Param("componentIds") Collection<Long> componentIds);

  List<Integer> keepAuthorizedUsersForRoleAndProject(@Param("role") String role, @Param("componentId") long
componentId, @Param("userIds") List<Integer> userIds);

  Set<String> keepAuthorizedProjectUuidsForUser(@Param("userId") int userId, @Param("permission") String
permission, @Param("projectUuids") Collection<String> projectUuids);

  Set<String> keepAuthorizedProjectUuidsForAnonymous(@Param("permission") String permission,
@Param("projectUuids") Collection<String> projectUuids);

  Set<String> selectProjectPermissions(@Param("projectUuid") String projectUuid, @Param("userId") long userId);

  Set<String> selectProjectPermissionsOfAnonymous(@Param("projectUuid") String projectUuid);

  List<String> selectQualityProfileAdministratorLogins(@Param("permission") String permission);

  Set<String> keepAuthorizedLoginsOnProject(@Param("logins") List<String> logins, @Param("projectKey")
String projectKey, @Param("permission") String permission);

  List<String> selectLoginsWithGlobalPermission(@Param("permission") String permission);
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
```

```java
package org.sonar.db.permission;

import java.util.Collection;
import java.util.List;
import java.util.Set;
import org.sonar.core.util.stream.MoreCollectors;
import org.sonar.db.Dao;
import org.sonar.db.DatabaseUtils;
import org.sonar.db.DbSession;
import org.sonar.db.component.ComponentMapper;

import static com.google.common.base.Preconditions.checkArgument;
import static java.util.Collections.emptyList;
import static org.sonar.db.DatabaseUtils.executeLargeInputs;

public class UserPermissionDao implements Dao {

  /**
   * List of user permissions ordered by alphabetical order of user names.
   * Pagination is NOT applied.
   * No sort is done.
   *
   * @param query non-null query including optional filters.
   * @param userIds Filter on user ids, including disabled users. Must not be empty and maximum size is {@link
   * DatabaseUtils#PARTITION_SIZE_FOR_ORACLE}.
   */
  public List<UserPermissionDto> selectUserPermissionsByQuery(DbSession dbSession, PermissionQuery query,
  Collection<Integer> userIds) {
    if (userIds.isEmpty()) {
      return emptyList();
    }
    checkArgument(userIds.size() <= DatabaseUtils.PARTITION_SIZE_FOR_ORACLE, "Maximum 1'000 users are
accepted");
    return mapper(dbSession).selectUserPermissionsByQueryAndUserIds(query, userIds);
  }

  public List<Integer> selectUserIdsByQuery(DbSession dbSession, PermissionQuery query) {
    return mapper(dbSession).selectUserIdsByQuery(query)
      .stream()
      // Pagination is done in Java because it's too complex to use SQL pagination in Oracle and MsSQL with the
distinct
      .skip(query.getPageOffset())
```

```java
      .limit(query.getPageSize())
      .collect(MoreCollectors.toList());
  }

  public int countUsersByQuery(DbSession dbSession, PermissionQuery query) {
    return mapper(dbSession).countUsersByQuery(query);
  }

  /**
   * Count the number of users per permission for a given list of projects
   *
   * @param projectIds a non-null list of project ids to filter on. If empty then an empty list is returned.
   */
  public List<CountPerProjectPermission> countUsersByProjectPermission(DbSession dbSession, Collection<Long>
projectIds) {
    return executeLargeInputs(projectIds, mapper(dbSession)::countUsersByProjectPermission);
  }

  /**
   * Gets all the global permissions granted to user for the specified organization.
   *
   * @return the global permissions. An empty list is returned if user or organization do not exist.
   */
  public List<String> selectGlobalPermissionsOfUser(DbSession dbSession, int userId, String organizationUuid) {
    return mapper(dbSession).selectGlobalPermissionsOfUser(userId, organizationUuid);
  }

  /**
   * Gets all the project permissions granted to user for the specified project.
   *
   * @return the project permissions. An empty list is returned if project or user do not exist.
   */
  public List<String> selectProjectPermissionsOfUser(DbSession dbSession, int userId, long projectId) {
    return mapper(dbSession).selectProjectPermissionsOfUser(userId, projectId);
  }

  public Set<Integer> selectUserIdsWithPermissionOnProjectBut(DbSession session, long projectId, String
permission) {
    return mapper(session).selectUserIdsWithPermissionOnProjectBut(projectId, permission);
  }

  public void insert(DbSession dbSession, UserPermissionDto dto) {
    ensureComponentPermissionConsistency(dbSession, dto);
    mapper(dbSession).insert(dto);
  }

  private static void ensureComponentPermissionConsistency(DbSession dbSession, UserPermissionDto dto) {
    if (dto.getComponentId() == null) {
```

```java
    return;
  }
  ComponentMapper componentMapper = dbSession.getMapper(ComponentMapper.class);
  checkArgument(
    componentMapper.countComponentByOrganizationAndId(dto.getOrganizationUuid(), dto.getComponentId())
== 1,
    "Can't insert permission '%s' for component with id '%s' in organization with uuid '%s' because this component
does not belong to organization with uuid '%s'",
    dto.getPermission(), dto.getComponentId(), dto.getOrganizationUuid(), dto.getOrganizationUuid());
}

/**
 * Removes a single global permission from user
 */
public void deleteGlobalPermission(DbSession dbSession, int userId, String permission, String organizationUuid) {
  mapper(dbSession).deleteGlobalPermission(userId, permission, organizationUuid);
}

/**
 * Removes a single project permission from user
 */
public void deleteProjectPermission(DbSession dbSession, int userId, String permission, long projectId) {
  mapper(dbSession).deleteProjectPermission(userId, permission, projectId);
}

/**
 * Deletes all the permissions defined on a project
 */
public void deleteProjectPermissions(DbSession dbSession, long projectId) {
  mapper(dbSession).deleteProjectPermissions(projectId);
}

/**
 * Deletes the specified permission on the specified project for any user.
 */
public int deleteProjectPermissionOfAnyUser(DbSession dbSession, long projectId, String permission) {
  return mapper(dbSession).deleteProjectPermissionOfAnyUser(projectId, permission);
}

public void deleteByOrganization(DbSession dbSession, String organizationUuid) {
  mapper(dbSession).deleteByOrganization(organizationUuid);
}

public void deleteOrganizationMemberPermissions(DbSession dbSession, String organizationUuid, int userId) {
  mapper(dbSession).deleteOrganizationMemberPermissions(organizationUuid, userId);
}

public void deleteByUserId(DbSession dbSession, int userId) {
```

```
    mapper(dbSession).deleteByUserId(userId);
  }

 private static UserPermissionMapper mapper(DbSession dbSession) {
   return dbSession.getMapper(UserPermissionMapper.class);
 }
}
```
/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com
*
* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU Lesser General Public
* License as published by the Free Software Foundation; either
* version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
* Lesser General Public License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
*/
package org.sonar.db.permission;

import java.util.Collection;
import java.util.List;
import java.util.Set;
import javax.annotation.Nullable;
import org.sonar.db.Dao;
import org.sonar.db.DbSession;

import static org.sonar.db.DatabaseUtils.executeLargeInputs;
import static org.sonar.db.DatabaseUtils.executeLargeInputsIntoSet;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_PROFILES;

/**
* The SQL requests used to verify authorization (the permissions
* granted to users)
*
* @see GroupPermissionDao for CRUD of table group_roles
* @see UserPermissionDao for CRUD of table user_roles
*/
public class AuthorizationDao implements Dao {

```java
/**
 * Loads all the permissions granted to logged-in user for the specified organization
 */
public Set<String> selectOrganizationPermissions(DbSession dbSession, String organizationUuid, int userId) {
  return mapper(dbSession).selectOrganizationPermissions(organizationUuid, userId);
}

/**
 * Loads all the permissions granted to anonymous user for the specified organization
 */
public Set<String> selectOrganizationPermissionsOfAnonymous(DbSession dbSession, String organizationUuid) {
  return mapper(dbSession).selectOrganizationPermissionsOfAnonymous(organizationUuid);
}

/**
 * Loads all the permissions granted to logged-in user for the specified project <strong>stored in *_ROLES
 * tables</strong>.
 * An empty Set is returned if user has no permissions on the project.
 *
 * <strong>This method does not support public components</strong>
 */
public Set<String> selectProjectPermissions(DbSession dbSession, String projectUuid, long userId) {
  return mapper(dbSession).selectProjectPermissions(projectUuid, userId);
}

/**
 * Loads all the permissions granted to anonymous for the specified project <strong>stored in *_ROLES
 * tables</strong>.
 * An empty Set is returned if anonymous user has no permissions on the project.
 *
 * <strong>This method does not support public components</strong>
 */
public Set<String> selectProjectPermissionsOfAnonymous(DbSession dbSession, String projectUuid) {
  return mapper(dbSession).selectProjectPermissionsOfAnonymous(projectUuid);
}

/**
 * The number of users who will still have the permission if the group {@code excludedGroupId}
 * is deleted. The anyone virtual group is not taken into account.
 */
public int countUsersWithGlobalPermissionExcludingGroup(DbSession dbSession, String organizationUuid,
  String permission, int excludedGroupId) {
  return mapper(dbSession).countUsersWithGlobalPermissionExcludingGroup(organizationUuid, permission,
excludedGroupId);
}

/**
```

```
 * The number of users who will still have the permission if the user {@code excludedUserId}
 * is deleted. The anyone virtual group is not taken into account.
 */
public int countUsersWithGlobalPermissionExcludingUser(DbSession dbSession, String organizationUuid,
  String permission, int excludedUserId) {
  return mapper(dbSession).countUsersWithGlobalPermissionExcludingUser(organizationUuid, permission,
excludedUserId);
 }

/**
 * The number of users who will still have the permission if the user {@code userId}
 * is removed from group {@code groupId}. The anyone virtual group is not taken into account.
 * Contrary to {@link #countUsersWithGlobalPermissionExcludingUser(DbSession, String, String, int)}, user
 * still exists and may have the permission directly or through other groups.
 */
public int countUsersWithGlobalPermissionExcludingGroupMember(DbSession dbSession, String
organizationUuid,
  String permission, int groupId, int userId) {
  return mapper(dbSession).countUsersWithGlobalPermissionExcludingGroupMember(organizationUuid,
permission, groupId, userId);
 }

/**
 * The number of users who will still have the permission if the permission {@code permission}
 * is removed from user {@code userId}. The anyone virtual group is not taken into account.
 * Contrary to {@link #countUsersWithGlobalPermissionExcludingUser(DbSession, String, String, int)}, user
 * still exists and may have the permission through groups.
 */
public int countUsersWithGlobalPermissionExcludingUserPermission(DbSession dbSession, String
organizationUuid,
  String permission, int userId) {
  return mapper(dbSession).countUsersWithGlobalPermissionExcludingUserPermission(organizationUuid,
permission, userId);
 }

/**
 * The UUIDs of all the organizations in which the specified user has the specified global permission. An empty
 * set is returned if user or permission do not exist. An empty set is also returned if the user is not involved
 * in any organization.
 * <br/>
 * Group membership is taken into account. Anonymous privileges are ignored.
 */
public Set<String> selectOrganizationUuidsOfUserWithGlobalPermission(DbSession dbSession, int userId, String
permission) {
  return mapper(dbSession).selectOrganizationUuidsOfUserWithGlobalPermission(userId, permission);
 }

/**
```

```java
 * @deprecated replaced by {@link #keepAuthorizedProjectUuids(DbSession, Collection, Integer, String)}
 */
@Deprecated
public Set<Long> keepAuthorizedProjectIds(DbSession dbSession, Collection<Long> componentIds, @Nullable
Integer userId, String permission) {
  return executeLargeInputsIntoSet(
    componentIds,
    partition -> {
      if (userId == null) {
        return mapper(dbSession).keepAuthorizedProjectIdsForAnonymous(permission, partition);
      }
      return mapper(dbSession).keepAuthorizedProjectIdsForUser(userId, permission, partition);
    },
    partitionSize -> partitionSize / 2);
}

public Set<String> keepAuthorizedProjectUuids(DbSession dbSession, Collection<String> projectUuids,
@Nullable Integer userId, String permission) {
  return executeLargeInputsIntoSet(
    projectUuids,
    partition -> {
      if (userId == null) {
        return mapper(dbSession).keepAuthorizedProjectUuidsForAnonymous(permission, partition);
      }
      return mapper(dbSession).keepAuthorizedProjectUuidsForUser(userId, permission, partition);
    },
    partitionSize -> partitionSize / 2);
}

/**
 * Keep only authorized user that have the given permission on a given project.
 * Please Note that if the permission is 'Anyone' is NOT taking into account by this method.
 */
public Collection<Integer> keepAuthorizedUsersForRoleAndProject(DbSession dbSession, Collection<Integer>
userIds, String role, long projectId) {
  return executeLargeInputs(
    userIds,
    partitionOfIds -> mapper(dbSession).keepAuthorizedUsersForRoleAndProject(role, projectId, partitionOfIds),
    partitionSize -> partitionSize / 3);
}

public List<String> selectQualityProfileAdministratorLogins(DbSession dbSession) {
  return mapper(dbSession).selectLoginsWithGlobalPermission(ADMINISTER_QUALITY_PROFILES.getKey());
}

/**
 * Used by license notifications
 */
```

```java
 public List<String> selectGlobalAdministratorLogins(DbSession dbSession) {
   return mapper(dbSession).selectLoginsWithGlobalPermission(ADMINISTER.getKey());
 }

 public Set<String> keepAuthorizedLoginsOnProject(DbSession dbSession, Set<String> logins, String projectKey,
String permission) {
   return executeLargeInputsIntoSet(
     logins,
     partitionOfLogins -> mapper(dbSession).keepAuthorizedLoginsOnProject(partitionOfLogins, projectKey,
permission),
     partitionSize -> partitionSize / 3);
 }

 private static AuthorizationMapper mapper(DbSession dbSession) {
   return dbSession.getMapper(AuthorizationMapper.class);
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import com.google.common.annotations.VisibleForTesting;

/**
 * Count the number of users or groups for a given project and permission
 */
public class CountPerProjectPermission {
 private long componentId;
 private String permission;
 private int count;
```

```java
public CountPerProjectPermission() {
  // used by MyBatis
}

@VisibleForTesting
CountPerProjectPermission(long componentId, String permission, int count) {
  this.componentId = componentId;
  this.permission = permission;
  this.count = count;
}

public long getComponentId() {
  return componentId;
}

public String getPermission() {
  return permission;
}

public int getCount() {
  return count;
}
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import java.util.Arrays;
import java.util.stream.Stream;

public enum OrganizationPermission {
```

```java
  ADMINISTER("admin"),
  ADMINISTER_QUALITY_GATES("gateadmin"),
  ADMINISTER_QUALITY_PROFILES("profileadmin"),
  PROVISION_PROJECTS("provisioning"),
  SCAN("scan");

  private final String key;

  OrganizationPermission(String key) {
    this.key = key;
  }

  public String getKey() {
    return key;
  }

  @Override
  public String toString() {
    return key;
  }

  public static OrganizationPermission fromKey(String key) {
    for (OrganizationPermission p : values()) {
      if (p.getKey().equals(key)) {
        return p;
      }
    }
    throw new IllegalArgumentException("Unsupported permission: " + key);
  }

  public static Stream<OrganizationPermission> all() {
    return Arrays.stream(values());
  }
}
<dataset>

<!-- user 100 has the role "user" on the project 300 and in group 200 -->
<user_roles id="1"
      user_id="100"
      resource_id="300"
      role="user"
      organization_uuid="org1"/>
<groups_users user_id="100"
        group_id="200"/>
<group_roles id="1"
      group_id="200"
      resource_id="999"
```

```
                role="user"
                organization_uuid="org1"/>


<projects organization_uuid="org1"
        id="300"
        uuid="ABCD"
        uuid_path="NOT_USED"
        root_uuid="ABCD"
        project_uuid="ABCD"
        module_uuid="[null]"
        kee="pj-w-snapshot"
        scope="PRJ"
        qualifier="TRK"
        enabled="[true]"
        private="[false]"/>
<projects organization_uuid="org1"
        id="301"
        uuid="BCDE"
        uuid_path="NOT_USED"
        root_uuid="BCDE"
        project_uuid="BCDE"
        module_uuid="[null]"
        kee="pj-w-snapshot1"
        scope="PRJ"
        qualifier="TRK"
        enabled="[true]"
        private="[false]"/>
<projects organization_uuid="org1"
        id="302"
        uuid="CDEF"
        uuid_path="NOT_USED"
        root_uuid="CDEF"
        project_uuid="CDEF"
        module_uuid="[null]"
        kee="pj-w-snapshot2"
        scope="PRJ"
        qualifier="TRK"
        enabled="[true]"
        private="[false]"/>


<projects organization_uuid="org1"
        id="303"
        uuid="DEFG"
        uuid_path="NOT_USED"
        root_uuid="DEFG"
        project_uuid="DEFG"
        module_uuid="[null]"
        kee="pj-w-snapshot3"
```

```
            scope="PRJ"
            qualifier="TRK"
            enabled="[true]"
            private="[false]"/>


    </dataset>
    <dataset>


    <user_roles id="1"
            user_id="100"
            resource_id="999"
            role="user"
            organization_uuid="org1"/>
    <groups_users user_id="100"
             group_id="200"/>
    <group_roles id="1"
            group_id="[null]"
            resource_id="300"
            role="user"
            organization_uuid="org1"/>


    <projects organization_uuid="org1"
          id="300"
          uuid="ABCD"
          uuid_path="NOT_USED"
          root_uuid="ABCD"
          project_uuid="ABCD"
          module_uuid="[null]"
          kee="pj-w-snapshot"
          scope="PRJ"
          qualifier="TRK"
          enabled="[true]"
          private="[false]"/>
    <projects organization_uuid="org1"
          id="301"
          uuid="BCDE"
          uuid_path="NOT_USED"
          root_uuid="BCDE"
          project_uuid="BCDE"
          module_uuid="[null]"
          kee="pj-w-snapshot1"
          scope="PRJ"
          qualifier="TRK"
          enabled="[true]"
          private="[false]"/>
    <projects organization_uuid="org1"
          id="302"
          uuid="CDEF"
```

```
       uuid_path="NOT_USED"
       root_uuid="CDEF"
       project_uuid="CDEF"
       module_uuid="[null]"
       kee="pj-w-snapshot2"
       scope="PRJ"
       qualifier="TRK"
       enabled="[true]"
       private="[false]"/>

<projects organization_uuid="org1"
       id="303"
       uuid="DEFG"
       uuid_path="NOT_USED"
       root_uuid="DEFG"
       project_uuid="DEFG"
       module_uuid="[null]"
       kee="pj-w-snapshot3"
       scope="PRJ"
       qualifier="TRK"
       enabled="[true]"
       private="[false]"/>

</dataset>
<dataset>

<!-- user 100 has no direct grant access, but is in the group 200 that has the role "user"
on the project 300  -->
<user_roles id="1"
       user_id="100"
       resource_id="999"
       role="user"
       organization_uuid="org1"/>
<groups_users user_id="100"
        group_id="200"/>
<group_roles id="1"
       group_id="200"
       resource_id="300"
       role="user"
       organization_uuid="org1"/>

<projects organization_uuid="org1"
       id="300"
       uuid="ABCD"
       uuid_path="NOT_USED"
       root_uuid="ABCD"
       project_uuid="ABCD"
       module_uuid="[null]"
```

```
        kee="pj-w-snapshot"
        scope="PRJ"
        qualifier="TRK"
        enabled="[true]"
        private="[false]"/>
  <projects organization_uuid="org1"
        id="301"
        uuid="BCDE"
        uuid_path="NOT_USED"
        root_uuid="BCDE"
        project_uuid="BCDE"
        module_uuid="[null]"
        kee="pj-w-snapshot1"
        scope="PRJ"
        qualifier="TRK"
        enabled="[true]"
        private="[false]"/>
  <projects organization_uuid="org1"
        id="302"
        uuid="CDEF"
        uuid_path="NOT_USED"
        root_uuid="CDEF"
        project_uuid="CDEF"
        module_uuid="[null]"
        kee="pj-w-snapshot2"
        scope="PRJ"
        qualifier="TRK"
        enabled="[true]"
        private="[false]"/>

  <projects organization_uuid="org1"
        id="303"
        uuid="DEFG"
        uuid_path="NOT_USED"
        root_uuid="DEFG"
        project_uuid="DEFG"
        module_uuid="[null]"
        kee="pj-w-snapshot3"
        scope="PRJ"
        qualifier="TRK"
        enabled="[true]"
        private="[false]"/>

</dataset>
```

/*
* SonarQube
* Copyright (C) 2009-2018 SonarSource SA
* mailto:info AT sonarsource DOT com

```
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.core.permission;

import org.junit.Test;
import org.sonar.api.web.UserRole;

import static org.assertj.core.api.Assertions.assertThat;

public class ProjectPermissionsTest {

 @Test
 public void all_permissions() {
   assertThat(ProjectPermissions.ALL).containsExactly(UserRole.ADMIN, UserRole.CODEVIEWER,
UserRole.ISSUE_ADMIN, GlobalPermissions.SCAN_EXECUTION, UserRole.USER);
 }

 @Test
 public void all_permissions_as_string() {
   assertThat(ProjectPermissions.ALL_ON_ONE_LINE).isEqualTo("admin, codeviewer, issueadmin, scan, user");
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
```

package org.sonar.db.permission;

import org.junit.Test;

import static org.assertj.core.api.Assertions.assertThat;

public class OrganizationPermissionTest {

 @Test
 public void fromKey_returns_enum_with_specified_key() {
  for (OrganizationPermission p : OrganizationPermission.values()) {
   assertThat(OrganizationPermission.fromKey(p.getKey())).isEqualTo(p);
  }
 }

 @Test
 public void all_returns_stream_of_values() {
  assertThat(OrganizationPermission.all()).hasSize(OrganizationPermission.values().length);
  for (OrganizationPermission permission : OrganizationPermission.values()) {
   assertThat(OrganizationPermission.all()).contains(permission);
  }
 }
}

```java
package org.sonar.db.permission;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.List;
import java.util.Random;
import java.util.function.Consumer;
import java.util.stream.Collectors;
import org.assertj.core.groups.Tuple;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.UserDto;

import static java.util.Arrays.asList;
import static java.util.Arrays.stream;
import static java.util.Collections.emptyList;
import static java.util.Collections.singletonList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.tuple;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.PROVISIONING;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;
import static org.sonar.db.permission.OrganizationPermission.SCAN;

public class UserPermissionDaoTest {

  @Rule
  public DbTester db = DbTester.create(System2.INSTANCE);

  private DbSession dbSession = db.getSession();
  private UserPermissionDao underTest = new UserPermissionDao();

  @Test
  public void select_global_permissions() {
    OrganizationDto organization = db.organizations().insert();
    OrganizationDto org2 = db.organizations().insert();
```

```
  UserDto user1 = insertUser(u -> u.setLogin("login1").setName("Marius").setEmail("email1@email.com"),
organization, org2);
  UserDto user2 = insertUser(u -> u.setLogin("login2").setName("Marie").setEmail("email2@email.com"),
organization, org2);
  UserDto user3 = insertUser(u -> u.setLogin("zanother").setName("Zoe").setEmail("zanother3@another.com"),
organization);
  ComponentDto project = db.components().insertPrivateProject(organization);
  UserPermissionDto global1 = addGlobalPermission(organization, SYSTEM_ADMIN, user1);
  UserPermissionDto global2 = addGlobalPermission(organization, SYSTEM_ADMIN, user2);
  UserPermissionDto global3 = addGlobalPermission(organization, PROVISIONING, user2);
  UserPermissionDto project1Perm = addProjectPermission(organization, USER, user3, project);
  // permissions on another organization, to be excluded
  UserPermissionDto org2Global1 = addGlobalPermission(org2, SYSTEM_ADMIN, user1);
  UserPermissionDto org2Global2 = addGlobalPermission(org2, PROVISIONING, user2);

  // global permissions of users who has at least one global permission, ordered by user name then permission
  PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().build();
  expectPermissions(query, asList(user2.getId(), user1.getId()), global2, global3, global1);

  // default query returns all users, whatever their permissions nor organizations
  // (that's a non-sense, but still this is required for api/permissions/groups
  // when filtering users by name)
  query = PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).build();
  expectPermissions(query, asList(user2.getId(), user1.getId(), user3.getId()), global2, global3, org2Global2,
global1, org2Global1, project1Perm);

  // global permissions "admin"
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setPermission(SYSTEM_ADMIN).build();
  expectPermissions(query, asList(user2.getId(), user1.getId()), global2, global1);

  // empty if nobody has the specified global permission
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setPermission("missing").build();
  expectPermissions(query, emptyList());

  // search by user name (matches 2 users)
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setSearchQue
ry("mari").build();
  expectPermissions(query, asList(user2.getId(), user1.getId()), global2, global3, global1);

  // search by user login
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setSearchQue
ry("ogin2").build();
  expectPermissions(query, singletonList(user2.getId()), global2, global3);
```

```java
    // search by user email
    query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setSearchQue
ry("mail2").build();
    expectPermissions(query, singletonList(user2.getId()), global2, global3);


    // search by user name (matches 2 users) and global permission
    query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setSearchQuery("Mari").setPermission(PR
OVISIONING).build();
    expectPermissions(query, singletonList(user2.getId()), global3);


    // search by user name (no match)
    query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setSearchQuery("Unknown").build();
    expectPermissions(query, emptyList());
  }

  @Test
  public void select_project_permissions() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = insertUser(u -> u.setLogin("login1").setName("Marius").setEmail("email1@email.com"),
organization);
    UserDto user2 = insertUser(u -> u.setLogin("login2").setName("Marie").setEmail("email2@email.com"),
organization);
    UserDto user3 = insertUser(u -> u.setLogin("zanother").setName("Zoe").setEmail("zanother3@another.com"),
organization);
    addGlobalPermission(organization, SYSTEM_ADMIN, user1);
    ComponentDto project1 = db.components().insertPrivateProject(organization);
    ComponentDto project2 = db.components().insertPrivateProject(organization);
    UserPermissionDto perm1 = addProjectPermission(organization, USER, user1, project1);
    UserPermissionDto perm2 = addProjectPermission(organization, ISSUE_ADMIN, user1, project1);
    UserPermissionDto perm3 = addProjectPermission(organization, ISSUE_ADMIN, user2, project1);
    addProjectPermission(organization, ISSUE_ADMIN, user3, project2);

    // project permissions of users who has at least one permission on this project
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setComponen
tUuid(project1.uuid()).build();
    expectPermissions(query, asList(user2.getId(), user1.getId()), perm3, perm2, perm1);

    // empty if nobody has the specified global permission
    query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setPermission("missing").setComponentUu
id(project1.uuid()).build();
    expectPermissions(query, emptyList());
```

```java
  // search by user name (matches 2 users), users with at least one permission
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setSearchQuery("Mari").withAtLeastOneP
ermission().setComponentUuid(project1.uuid()).build();
  expectPermissions(query, asList(user2.getId(), user1.getId()), perm3, perm2, perm1);

  // search by user name (matches 2 users) and project permission
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setSearchQuery("Mari").setPermission(ISS
UE_ADMIN).setComponentUuid(project1.uuid()).build();
  expectPermissions(query, asList(user2.getId(), user1.getId()), perm3, perm2);

  // search by user name (no match)
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setSearchQuery("Unknown").setCompone
ntUuid(project1.uuid()).build();
  expectPermissions(query, emptyList());

  // permissions of unknown project
  query =
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).setComponentUuid("missing").withAtLeas
tOnePermission().build();
  expectPermissions(query, emptyList());
 }

 @Test
 public void countUsersByProjectPermission() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user1 = insertUser(organization);
  UserDto user2 = insertUser(organization);
  ComponentDto project1 = db.components().insertPrivateProject(organization);
  ComponentDto project2 = db.components().insertPrivateProject(organization);
  addGlobalPermission(organization, SYSTEM_ADMIN, user1);
  addProjectPermission(organization, USER, user1, project1);
  addProjectPermission(organization, ISSUE_ADMIN, user1, project1);
  addProjectPermission(organization, ISSUE_ADMIN, user2, project1);
  addProjectPermission(organization, ISSUE_ADMIN, user2, project2);

  // no projects -> return empty list
  assertThat(underTest.countUsersByProjectPermission(dbSession, emptyList())).isEmpty();

  // one project
  expectCount(singletonList(project1.getId()),
    new CountPerProjectPermission(project1.getId(), USER, 1),
    new CountPerProjectPermission(project1.getId(), ISSUE_ADMIN, 2));

  // multiple projects
  expectCount(asList(project1.getId(), project2.getId(), -1L),
```

```java
      new CountPerProjectPermission(project1.getId(), USER, 1),
      new CountPerProjectPermission(project1.getId(), ISSUE_ADMIN, 2),
      new CountPerProjectPermission(project2.getId(), ISSUE_ADMIN, 1));
  }

  @Test
  public void selectUserIdsByQuery() {
    OrganizationDto org1 = db.organizations().insert();
    OrganizationDto org2 = db.organizations().insert();
    UserDto user1 = insertUser(u -> u.setLogin("login1").setName("Marius").setEmail("email1@email.com"), org1,
org2);
    UserDto user2 = insertUser(u -> u.setLogin("login2").setName("Marie").setEmail("email2@email.com"), org1,
org2);
    ComponentDto project1 = db.components().insertPrivateProject(org1);
    ComponentDto project2 = db.components().insertPrivateProject(org2);
    addProjectPermission(org1, USER, user1, project1);
    addProjectPermission(org1, USER, user2, project1);
    addProjectPermission(org2, USER, user1, project2);
    addProjectPermission(org1, ISSUE_ADMIN, user2, project1);
    addProjectPermission(org2, ISSUE_ADMIN, user2, project2);

    // logins are ordered by user name: user2 ("Marie") then user1 ("Marius")
    PermissionQuery query =
PermissionQuery.builder().setOrganizationUuid(project1.getOrganizationUuid()).setComponentUuid(project1.uuid(
)).withAtLeastOnePermission().build();
    assertThat(underTest.selectUserIdsByQuery(dbSession, query)).containsExactly(user2.getId(), user1.getId());
    query =
PermissionQuery.builder().setOrganizationUuid("anotherOrg").setComponentUuid(project1.uuid()).withAtLeastOn
ePermission().build();
    assertThat(underTest.selectUserIdsByQuery(dbSession, query)).isEmpty();

    // on a project without permissions
    query =
PermissionQuery.builder().setOrganizationUuid(org1.getUuid()).setComponentUuid("missing").withAtLeastOnePer
mission().build();
    assertThat(underTest.selectUserIdsByQuery(dbSession, query)).isEmpty();

    // search all users whose name matches "mar", whatever the permissions
    query = PermissionQuery.builder().setOrganizationUuid(org1.getUuid()).setSearchQuery("mar").build();
    assertThat(underTest.selectUserIdsByQuery(dbSession, query)).containsExactly(user2.getId(), user1.getId());

    // search all users whose name matches "mariu", whatever the permissions
    query = PermissionQuery.builder().setOrganizationUuid(org1.getUuid()).setSearchQuery("mariu").build();
    assertThat(underTest.selectUserIdsByQuery(dbSession, query)).containsExactly(user1.getId());

    // search all users whose name matches "mariu", whatever the permissions
    query =
PermissionQuery.builder().setOrganizationUuid(org1.getUuid()).setSearchQuery("mariu").setComponentUuid(proje
```

```java
      ct1.uuid())).build();
      assertThat(underTest.selectUserIdsByQuery(dbSession, query)).containsExactly(user1.getId());

      // search all users whose name matches "mariu", whatever the organization
      query = PermissionQuery.builder().setOrganizationUuid("missingOrg").setSearchQuery("mariu").build();
      assertThat(underTest.selectUserIdsByQuery(dbSession, query)).isEmpty();
  }

  @Test
  public void selectUserIdsByQuery_is_paginated() {
    OrganizationDto organization = db.organizations().insert();
    List<Integer> userIds = new ArrayList<>();
    for (int i = 0; i < 10; i++) {
      String name = "user-" + i;
      UserDto user = insertUser(u -> u.setName(name), organization);
      addGlobalPermission(organization, PROVISIONING, user);
      addGlobalPermission(organization, SYSTEM_ADMIN, user);
      userIds.add(user.getId());
    }

    assertThat(underTest.selectUserIdsByQuery(dbSession,
PermissionQuery.builder().setOrganizationUuid(organization.getUuid())
      .setPageSize(3).setPageIndex(1).build()))
        .containsExactly(userIds.get(0), userIds.get(1), userIds.get(2));
    assertThat(underTest.selectUserIdsByQuery(dbSession,
PermissionQuery.builder().setOrganizationUuid(organization.getUuid())
      .setPageSize(2).setPageIndex(3).build()))
        .containsExactly(userIds.get(4), userIds.get(5));
    assertThat(underTest.selectUserIdsByQuery(dbSession,
PermissionQuery.builder().setOrganizationUuid(organization.getUuid())
      .setPageSize(50).setPageIndex(1).build()))
        .hasSize(10);
  }

  @Test
  public void selectUserIdsByQuery_is_sorted_by_insensitive_name() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = insertUser(u -> u.setName("user1"), organization);
    addGlobalPermission(organization, PROVISIONING, user1);
    UserDto user3 = insertUser(u -> u.setName("user3"), organization);
    addGlobalPermission(organization, SYSTEM_ADMIN, user3);
    UserDto user2 = insertUser(u -> u.setName("User2"), organization);
    addGlobalPermission(organization, PROVISIONING, user2);

    assertThat(underTest.selectUserIdsByQuery(dbSession,
PermissionQuery.builder().setOrganizationUuid(organization.getUuid()).build()))
        .containsExactly(user1.getId(), user2.getId(), user3.getId());
  }
```

```java
  @Test
  public void deleteGlobalPermission() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = insertUser(organization);
    UserDto user2 = insertUser(organization);
    ComponentDto project1 = db.components().insertPrivateProject(organization);
    ComponentDto project2 = db.components().insertPrivateProject(organization);
    addGlobalPermission(organization, "perm1", user1);
    addGlobalPermission(organization, "perm2", user1);
    addProjectPermission(organization, "perm1", user1, project1);
    addProjectPermission(organization, "perm3", user2, project1);
    addProjectPermission(organization, "perm4", user2, project2);

    // user2 does not have global permissions -> do nothing
    underTest.deleteGlobalPermission(dbSession, user2.getId(), "perm1", db.getDefaultOrganization().getUuid());
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(5);

    // global permission is not granted -> do nothing
    underTest.deleteGlobalPermission(dbSession, user1.getId(), "notGranted",
db.getDefaultOrganization().getUuid());
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(5);

    // permission is on project -> do nothing
    underTest.deleteGlobalPermission(dbSession, user1.getId(), "perm3", db.getDefaultOrganization().getUuid());
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(5);

    // global permission on another organization-> do nothing
    underTest.deleteGlobalPermission(dbSession, user1.getId(), "notGranted", "anotherOrg");
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(5);

    // global permission exists -> delete it, but not the project permission with the same name !
    underTest.deleteGlobalPermission(dbSession, user1.getId(), "perm1", organization.getUuid());
    assertThat(db.countSql(dbSession, "select count(id) from user_roles where role='perm1' and resource_id is
null")).isEqualTo(0);
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(4);
  }

  @Test
  public void deleteProjectPermission() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = insertUser(organization);
    UserDto user2 = insertUser(organization);
    ComponentDto project1 = db.components().insertPrivateProject(organization);
    ComponentDto project2 = db.components().insertPrivateProject(organization);
    addGlobalPermission(organization, "perm", user1);
    addProjectPermission(organization, "perm", user1, project1);
    addProjectPermission(organization, "perm", user1, project2);
```

```java
    addProjectPermission(organization, "perm", user2, project1);

    // no such provision -> ignore
    underTest.deleteProjectPermission(dbSession, user1.getId(), "anotherPerm", project1.getId());
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(4);

    underTest.deleteProjectPermission(dbSession, user1.getId(), "perm", project1.getId());
    assertThatProjectPermissionDoesNotExist(user1, "perm", project1);
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(3);
  }

  @Test
  public void deleteProjectPermissions() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = insertUser(organization);
    UserDto user2 = insertUser(organization);
    ComponentDto project1 = db.components().insertPrivateProject(organization);
    ComponentDto project2 = db.components().insertPrivateProject(organization);
    addGlobalPermission(organization, "perm", user1);
    addProjectPermission(organization, "perm", user1, project1);
    addProjectPermission(organization, "perm", user2, project1);
    addProjectPermission(organization, "perm", user1, project2);

    underTest.deleteProjectPermissions(dbSession, project1.getId());
    assertThat(db.countRowsOfTable(dbSession, "user_roles")).isEqualTo(2);
    assertThatProjectHasNoPermissions(project1);
  }

  @Test
  public void selectGlobalPermissionsOfUser() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = insertUser(organization);
    UserDto user2 = insertUser(organization);
    UserDto user3 = insertUser(organization);
    OrganizationDto org = db.organizations().insert();
    ComponentDto project = db.components().insertPrivateProject(organization);
    addGlobalPermission(db.getDefaultOrganization(), "perm1", user1);
    addGlobalPermission(org, "perm2", user2);
    addGlobalPermission(org, "perm3", user1);
    addProjectPermission(organization, "perm4", user1, project);
    addProjectPermission(organization, "perm5", user1, project);

    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user1.getId(),
org.getUuid())).containsOnly("perm3");
    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user1.getId(),
db.getDefaultOrganization().getUuid())).containsOnly("perm1");
    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user1.getId(), "otherOrg")).isEmpty();
    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user3.getId(), org.getUuid())).isEmpty();
```

```java
}

@Test
public void selectProjectPermissionsOfUser() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user1 = insertUser(organization);
  UserDto user2 = insertUser(organization);
  ComponentDto project1 = db.components().insertPrivateProject(organization);
  ComponentDto project2 = db.components().insertPrivateProject(organization);
  ComponentDto project3 = db.components().insertPrivateProject(organization);
  addGlobalPermission(organization, "perm1", user1);
  addProjectPermission(organization, "perm2", user1, project1);
  addProjectPermission(organization, "perm3", user1, project1);
  addProjectPermission(organization, "perm4", user1, project2);
  addProjectPermission(organization, "perm5", user2, project1);

  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user1.getId(),
project1.getId())).containsOnly("perm2", "perm3");
  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user1.getId(),
project2.getId())).containsOnly("perm4");
  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user1.getId(), project3.getId())).isEmpty();
}

@Test
public void selectGroupIdsWithPermissionOnProjectBut_returns_empty_if_project_does_not_exist() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = randomPublicOrPrivateProject(organization);
  UserDto user = insertUser(organization);
  db.users().insertProjectPermissionOnUser(user, "foo", project);

  assertThat(underTest.selectUserIdsWithPermissionOnProjectBut(dbSession, 1234, UserRole.USER))
    .isEmpty();
}

@Test
public void
selectGroupIdsWithPermissionOnProjectBut_returns_only_users_of_projects_which_do_not_have_permission() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = randomPublicOrPrivateProject(organization);
  UserDto user1 = insertUser(organization);
  UserDto user2 = insertUser(organization);
  db.users().insertProjectPermissionOnUser(user1, "p1", project);
  db.users().insertProjectPermissionOnUser(user2, "p2", project);

  assertThat(underTest.selectUserIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p2"))
    .containsOnly(user1.getId());
  assertThat(underTest.selectUserIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p1"))
    .containsOnly(user2.getId());
```

```java
  assertThat(underTest.selectUserIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p3"))
    .containsOnly(user1.getId(), user2.getId());
}

@Test
public void
selectGroupIdsWithPermissionOnProjectBut_does_not_return_groups_which_have_no_permission_at_all_on_speci
fied_project() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = randomPublicOrPrivateProject(organization);
  UserDto user1 = insertUser(organization);
  UserDto user2 = insertUser(organization);
  db.users().insertProjectPermissionOnUser(user1, "p1", project);
  db.users().insertProjectPermissionOnUser(user2, "p2", project);

  assertThat(underTest.selectUserIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p2"))
    .containsOnly(user1.getId());
  assertThat(underTest.selectUserIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p1"))
    .containsOnly(user2.getId());
}

@Test
public void deleteByOrganization_does_not_fail_if_table_is_empty() {
  underTest.deleteByOrganization(dbSession, "some uuid");
  dbSession.commit();
}

@Test
public void deleteByOrganization_does_not_fail_if_organization_has_no_user_permission() {
  OrganizationDto organization = db.organizations().insert();

  underTest.deleteByOrganization(dbSession, organization.getUuid());
  dbSession.commit();
}

@Test
public void deleteByOrganization_deletes_all_user_permission_of_specified_organization() {
  OrganizationDto organization1 = db.organizations().insert();
  OrganizationDto organization2 = db.organizations().insert();
  OrganizationDto organization3 = db.organizations().insert();
  UserDto user1 = insertUser(organization1, organization2, organization3);
  UserDto user2 = insertUser(organization1, organization2, organization3);
  UserDto user3 = insertUser(organization1, organization2, organization3);
  db.users().insertPermissionOnUser(organization1, user1, "foo");
  db.users().insertPermissionOnUser(organization1, user2, "foo");
  db.users().insertPermissionOnUser(organization1, user2, "bar");
  db.users().insertPermissionOnUser(organization2, user2, "foo");
  db.users().insertPermissionOnUser(organization2, user3, "foo");
```

```java
      db.users().insertPermissionOnUser(organization2, user3, "bar");
      db.users().insertPermissionOnUser(organization3, user3, "foo");
      db.users().insertPermissionOnUser(organization3, user1, "foo");
      db.users().insertPermissionOnUser(organization3, user1, "bar");

      underTest.deleteByOrganization(dbSession, organization3.getUuid());
      dbSession.commit();
      verifyOrganizationUuidsInTable(organization1.getUuid(), organization2.getUuid());

      underTest.deleteByOrganization(dbSession, organization2.getUuid());
      dbSession.commit();
      verifyOrganizationUuidsInTable(organization1.getUuid());

      underTest.deleteByOrganization(dbSession, organization1.getUuid());
      dbSession.commit();
      verifyOrganizationUuidsInTable();
    }

    @Test
    public void delete_permissions_of_an_organization_member() {
      OrganizationDto organization1 = db.organizations().insert();
      OrganizationDto organization2 = db.organizations().insert();
      ComponentDto project = db.components().insertPrivateProject(organization1);
      UserDto user1 = insertUser(organization1, organization2);
      UserDto user2 = insertUser(organization1, organization2);
      // user 1 permissions
      db.users().insertPermissionOnUser(organization1, user1, SCAN);
      db.users().insertPermissionOnUser(organization1, user1, ADMINISTER);
      db.users().insertProjectPermissionOnUser(user1, UserRole.CODEVIEWER, project);
      db.users().insertPermissionOnUser(organization2, user1, SCAN);
      // user 2 permission
      db.users().insertPermissionOnUser(organization1, user2, SCAN);
      db.users().insertProjectPermissionOnUser(user2, UserRole.CODEVIEWER, project);

      underTest.deleteOrganizationMemberPermissions(dbSession, organization1.getUuid(), user1.getId());
      dbSession.commit();

      // user 1 permissions
      assertOrgPermissionsOfUser(user1, organization1);
      assertOrgPermissionsOfUser(user1, organization2, SCAN);
      assertProjectPermissionsOfUser(user1, project);
      // user 2 permissions
      assertOrgPermissionsOfUser(user2, organization1, SCAN);
      assertProjectPermissionsOfUser(user2, project, CODEVIEWER);
    }

    @Test
    public void deleteByUserId() {
```

```java
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = insertUser(organization);
    UserDto user2 = insertUser(organization);
    ComponentDto project = db.components().insertPrivateProject(organization);
    db.users().insertPermissionOnUser(user1, SCAN);
    db.users().insertPermissionOnUser(user1, ADMINISTER);
    db.users().insertProjectPermissionOnUser(user1, ADMINISTER_QUALITY_GATES.getKey(), project);
    db.users().insertPermissionOnUser(user2, SCAN);
    db.users().insertProjectPermissionOnUser(user2, ADMINISTER_QUALITY_GATES.getKey(), project);

    underTest.deleteByUserId(dbSession, user1.getId());
    dbSession.commit();

    assertThat(db.select("select user_id as \"userId\", resource_id as \"projectId\", role as \"permission\" from
user_roles"))
      .extracting((row) -> row.get("userId"), (row) -> row.get("projectId"), (row) -> row.get("permission"))
      .containsOnly(tuple(user2.getId().longValue(), null, SCAN.getKey()), tuple(user2.getId().longValue(),
project.getId(), ADMINISTER_QUALITY_GATES.getKey()));
  }

  @Test
  public void deleteProjectPermissionOfAnyUser_has_no_effect_if_specified_component_does_not_exist() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user = insertUser(organization);
    db.users().insertPermissionOnUser(organization, user, SCAN);

    int deletedCount = underTest.deleteProjectPermissionOfAnyUser(dbSession, 124L, SCAN.getKey());

    assertThat(deletedCount).isEqualTo(0);
    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user.getId(),
organization.getUuid())).containsOnly(SCAN.getKey());
  }

  @Test
  public void
deleteProjectPermissionOfAnyUser_has_no_effect_if_specified_component_has_no_permission_at_all() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user = insertUser(organization);
    db.users().insertPermissionOnUser(organization, user, SCAN);
    ComponentDto project = randomPublicOrPrivateProject(organization);

    int deletedCount = underTest.deleteProjectPermissionOfAnyUser(dbSession, project.getId(), SCAN.getKey());

    assertThat(deletedCount).isEqualTo(0);
    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user.getId(),
organization.getUuid())).containsOnly(SCAN.getKey());
  }
```

```java
@Test
public void
deleteProjectPermissionOfAnyUser_has_no_effect_if_specified_component_does_not_have_specified_permission()
{
  OrganizationDto organization = db.organizations().insert();
  UserDto user = insertUser(organization);
  db.users().insertPermissionOnUser(organization, user, SCAN);
  ComponentDto project = randomPublicOrPrivateProject(organization);
  db.users().insertProjectPermissionOnUser(user, SCAN.getKey(), project);

  int deletedCount = underTest.deleteProjectPermissionOfAnyUser(dbSession, project.getId(), "p1");

  assertThat(deletedCount).isEqualTo(0);
  assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user.getId(),
organization.getUuid())).containsOnly(SCAN.getKey());
  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user.getId(),
project.getId())).containsOnly(SCAN.getKey());
 }

 @Test
 public void
deleteProjectPermissionOfAnyUser_deletes_specified_permission_for_any_user_on_the_specified_component() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user1 = insertUser(organization);
  UserDto user2 = insertUser(organization);
  db.users().insertPermissionOnUser(organization, user1, SCAN);
  db.users().insertPermissionOnUser(organization, user2, SCAN);
  ComponentDto project1 = randomPublicOrPrivateProject(organization);
  ComponentDto project2 = randomPublicOrPrivateProject(organization);
  db.users().insertProjectPermissionOnUser(user1, SCAN.getKey(), project1);
  db.users().insertProjectPermissionOnUser(user2, SCAN.getKey(), project1);
  db.users().insertProjectPermissionOnUser(user1, SCAN.getKey(), project2);
  db.users().insertProjectPermissionOnUser(user2, SCAN.getKey(), project2);
  db.users().insertProjectPermissionOnUser(user2, PROVISION_PROJECTS.getKey(), project2);

  int deletedCount = underTest.deleteProjectPermissionOfAnyUser(dbSession, project1.getId(), SCAN.getKey());

  assertThat(deletedCount).isEqualTo(2);
  assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user1.getId(),
organization.getUuid())).containsOnly(SCAN.getKey());
  assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user2.getId(),
organization.getUuid())).containsOnly(SCAN.getKey());
  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user1.getId(), project1.getId())).isEmpty();
  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user2.getId(), project1.getId())).isEmpty();
  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user1.getId(),
project2.getId())).containsOnly(SCAN.getKey());
  assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user2.getId(),
project2.getId())).containsOnly(SCAN.getKey(), PROVISION_PROJECTS.getKey());
```

```java
    deletedCount = underTest.deleteProjectPermissionOfAnyUser(dbSession, project2.getId(), SCAN.getKey());

    assertThat(deletedCount).isEqualTo(2);
    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user1.getId(),
organization.getUuid())).containsOnly(SCAN.getKey());
    assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user2.getId(),
organization.getUuid())).containsOnly(SCAN.getKey());
    assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user1.getId(), project1.getId())).isEmpty();
    assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user2.getId(), project1.getId())).isEmpty();
    assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user1.getId(), project2.getId())).containsOnly();
    assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user2.getId(),
project2.getId())).containsOnly(PROVISION_PROJECTS.getKey());
  }

  private ComponentDto randomPublicOrPrivateProject(OrganizationDto organization) {
    return new Random().nextBoolean() ? db.components().insertPrivateProject(organization) :
db.components().insertPublicProject(organization);
  }

  private UserDto insertUser(Consumer<UserDto> populateUserDto, OrganizationDto... organizations) {
    UserDto user = db.users().insertUser(populateUserDto);
    stream(organizations).forEach(organization -> db.organizations().addMember(organization, user));
    return user;
  }

  private UserDto insertUser(OrganizationDto... organizations) {
    UserDto user = db.users().insertUser();
    stream(organizations).forEach(organization -> db.organizations().addMember(organization, user));
    return user;
  }

  private void verifyOrganizationUuidsInTable(String... organizationUuids) {
    assertThat(db.select("select organization_uuid as \"organizationUuid\" from user_roles"))
      .extracting((row) -> (String) row.get("organizationUuid"))
      .containsOnly(organizationUuids);
  }

  private void expectCount(List<Long> projectIds, CountPerProjectPermission... expected) {
    List<CountPerProjectPermission> got = underTest.countUsersByProjectPermission(dbSession, projectIds);
    assertThat(got).hasSize(expected.length);

    for (CountPerProjectPermission expect : expected) {
      boolean found = got.stream().anyMatch(b -> b.getPermission().equals(expect.getPermission()) &&
        b.getCount() == expect.getCount() &&
        b.getComponentId() == expect.getComponentId());
      assertThat(found).isTrue();
    }
```

```java
  }

  private void expectPermissions(PermissionQuery query, Collection<Integer> expectedUserIds,
UserPermissionDto... expectedPermissions) {
    assertThat(underTest.selectUserIdsByQuery(dbSession, query)).containsExactly(expectedUserIds.toArray(new
Integer[0]));
    List<UserPermissionDto> currentPermissions = underTest.selectUserPermissionsByQuery(dbSession, query,
expectedUserIds);
    assertThat(currentPermissions).hasSize(expectedPermissions.length);
    List<Tuple> expectedPermissionsAsTuple = Arrays.stream(expectedPermissions)
      .map(expectedPermission -> tuple(expectedPermission.getUserId(), expectedPermission.getPermission(),
expectedPermission.getComponentId(),
        expectedPermission.getOrganizationUuid()))
      .collect(Collectors.toList());
    assertThat(currentPermissions)
      .extracting(UserPermissionDto::getUserId, UserPermissionDto::getPermission,
UserPermissionDto::getComponentId, UserPermissionDto::getOrganizationUuid)
      .containsOnly(expectedPermissionsAsTuple.toArray(new Tuple[0]));

    // test method "countUsers()"
    long distinctUsers = stream(expectedPermissions).mapToLong(UserPermissionDto::getUserId).distinct().count();
    assertThat((long) underTest.countUsersByQuery(dbSession, query)).isEqualTo(distinctUsers);
  }

  private UserPermissionDto addGlobalPermission(OrganizationDto org, String permission, UserDto user) {
    UserPermissionDto dto = new UserPermissionDto(org.getUuid(), permission, user.getId(), null);
    underTest.insert(dbSession, dto);
    db.commit();
    return dto;
  }

  private UserPermissionDto addProjectPermission(OrganizationDto org, String permission, UserDto user,
ComponentDto project) {
    UserPermissionDto dto = new UserPermissionDto(org.getUuid(), permission, user.getId(), project.getId());
    underTest.insert(dbSession, dto);
    db.commit();
    return dto;
  }

  private void assertThatProjectPermissionDoesNotExist(UserDto user, String permission, ComponentDto project) {
    assertThat(db.countSql(dbSession, "select count(id) from user_roles where role='" + permission + "' and user_id="
+ user.getId() + " and resource_id=" + project.getId()))
      .isEqualTo(0);
  }

  private void assertThatProjectHasNoPermissions(ComponentDto project) {
    assertThat(db.countSql(dbSession, "select count(id) from user_roles where resource_id=" +
project.getId())).isEqualTo(0);
```

```java
    }

    private void assertOrgPermissionsOfUser(UserDto user, OrganizationDto organization, OrganizationPermission...
permissions) {
      assertThat(underTest.selectGlobalPermissionsOfUser(dbSession, user.getId(), organization.getUuid()).stream()
        .map(OrganizationPermission::fromKey))
          .containsOnly(permissions);
    }

    private void assertProjectPermissionsOfUser(UserDto user, ComponentDto project, String... permissions) {
      assertThat(underTest.selectProjectPermissionsOfUser(dbSession, user.getId(),
project.getId())).containsOnly(permissions);
    }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.List;
import java.util.Random;
import java.util.stream.IntStream;
import java.util.stream.Stream;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
import org.sonar.core.util.stream.MoreCollectors;
```

```java
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.component.ComponentTesting;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;

import static java.util.Arrays.asList;
import static java.util.Collections.singletonList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.tuple;
import static org.sonar.api.security.DefaultGroups.ANYONE;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.PROVISIONING;
import static org.sonar.core.permission.GlobalPermissions.SCAN_EXECUTION;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.PROVISION_PROJECTS;
import static org.sonar.db.permission.OrganizationPermission.SCAN;

public class GroupPermissionDaoTest {

  private static final int ANYONE_ID = 0;
  private static final int MISSING_ID = -1;

  @Rule
  public DbTester db = DbTester.create(System2.INSTANCE);

  private DbSession dbSession = db.getSession();
  private GroupPermissionDao underTest = new GroupPermissionDao();
  private String defaultOrganizationUuid;

  @Before
  public void setUp() throws Exception {
    defaultOrganizationUuid = db.getDefaultOrganization().getUuid();
  }

  @Test
  public void group_count_by_permission_and_component_id_on_private_projects() {
    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    GroupDto group3 = db.users().insertGroup();
    ComponentDto project1 = db.components().insertPrivateProject();
    ComponentDto project2 = db.components().insertPrivateProject();
    ComponentDto project3 = db.components().insertPrivateProject();
```

```java
    db.users().insertProjectPermissionOnGroup(group1, ISSUE_ADMIN, project1);
    db.users().insertProjectPermissionOnGroup(group1, ADMIN, project2);
    db.users().insertProjectPermissionOnGroup(group2, ADMIN, project2);
    db.users().insertProjectPermissionOnGroup(group3, ADMIN, project2);
    db.users().insertProjectPermissionOnGroup(group1, USER, project2);
    db.users().insertProjectPermissionOnGroup(group1, USER, project3);

    final List<CountPerProjectPermission> result = new ArrayList<>();
    underTest.groupsCountByComponentIdAndPermission(dbSession, asList(project2.getId(), project3.getId(),
789L),
      context -> result.add((CountPerProjectPermission) context.getResultObject()));

    assertThat(result).hasSize(3);
    assertThat(result).extracting("permission").containsOnly(ADMIN, USER);
    assertThat(result).extracting("componentId").containsOnly(project2.getId(), project3.getId());
    assertThat(result).extracting("count").containsOnly(3, 1);
  }

  @Test
  public void group_count_by_permission_and_component_id_on_public_projects() {
    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    GroupDto group3 = db.users().insertGroup();
    ComponentDto project1 = db.components().insertPublicProject();
    ComponentDto project2 = db.components().insertPublicProject();
    ComponentDto project3 = db.components().insertPublicProject();

    db.users().insertProjectPermissionOnGroup(group1, "p1", project1);
    db.users().insertProjectPermissionOnGroup(group1, "p2", project2);
    db.users().insertProjectPermissionOnGroup(group2, "p2", project2);
    db.users().insertProjectPermissionOnGroup(group3, "p2", project2);
    // anyone group
    db.users().insertProjectPermissionOnAnyone("p2", project2);
    db.users().insertProjectPermissionOnGroup(group1, "p3", project2);
    db.users().insertProjectPermissionOnGroup(group1, "p3", project3);

    final List<CountPerProjectPermission> result = new ArrayList<>();
    underTest.groupsCountByComponentIdAndPermission(dbSession, asList(project2.getId(), project3.getId(),
789L),
      context -> result.add((CountPerProjectPermission) context.getResultObject()));

    assertThat(result).hasSize(3);
    assertThat(result).extracting("permission").containsOnly("p2", "p3");
    assertThat(result).extracting("componentId").containsOnly(project2.getId(), project3.getId());
    assertThat(result).extracting("count").containsOnly(4, 1);
  }

  @Test
```

```java
 public void selectGroupNamesByQuery_is_ordered_by_group_names() {
   OrganizationDto organizationDto = db.organizations().insert();
   GroupDto group2 = db.users().insertGroup(organizationDto, "Group-2");
   GroupDto group3 = db.users().insertGroup(organizationDto, "Group-3");
   GroupDto group1 = db.users().insertGroup(organizationDto, "Group-1");
   db.users().insertPermissionOnAnyone(organizationDto, SCAN);

   assertThat(underTest.selectGroupNamesByQuery(dbSession,
 newQuery().setOrganizationUuid(organizationDto.getUuid()).build()))
     .containsExactly(ANYONE, group1.getName(), group2.getName(), group3.getName());
 }

 @Test
 public void countGroupsByQuery() {
   OrganizationDto organizationDto = db.getDefaultOrganization();
   GroupDto group1 = db.users().insertGroup(organizationDto, "Group-1");
   db.users().insertGroup(organizationDto, "Group-2");
   db.users().insertGroup(organizationDto, "Group-3");
   db.users().insertPermissionOnAnyone(organizationDto, SCAN);
   db.users().insertPermissionOnGroup(group1, PROVISION_PROJECTS);

   assertThat(underTest.countGroupsByQuery(dbSession,
     newQuery().build())).isEqualTo(4);
   assertThat(underTest.countGroupsByQuery(dbSession,
     newQuery().setPermission(PROVISION_PROJECTS.getKey()).build())).isEqualTo(1);
   assertThat(underTest.countGroupsByQuery(dbSession,
     newQuery().withAtLeastOnePermission().build())).isEqualTo(2);
   assertThat(underTest.countGroupsByQuery(dbSession,
     newQuery().setSearchQuery("Group-").build())).isEqualTo(3);
   assertThat(underTest.countGroupsByQuery(dbSession,
     newQuery().setSearchQuery("Any").build())).isEqualTo(1);
 }

 @Test
 public void selectGroupNamesByQuery_with_global_permission() {
   OrganizationDto organizationDto = db.organizations().insert();
   GroupDto group1 = db.users().insertGroup(organizationDto, "Group-1");
   GroupDto group2 = db.users().insertGroup(organizationDto, "Group-2");
   GroupDto group3 = db.users().insertGroup(organizationDto, "Group-3");

   ComponentDto project =
 db.components().insertComponent(ComponentTesting.newPrivateProjectDto(organizationDto));

   db.users().insertPermissionOnAnyone(organizationDto, SCAN);
   db.users().insertPermissionOnAnyone(organizationDto, PROVISION_PROJECTS);
   db.users().insertPermissionOnGroup(group1, SCAN);
   db.users().insertPermissionOnGroup(group3, ADMINISTER);
   db.users().insertProjectPermissionOnGroup(group2, UserRole.ADMIN, project);
```

```java
    assertThat(underTest.selectGroupNamesByQuery(dbSession,
newQuery().setOrganizationUuid(organizationDto.getUuid()).setPermission(SCAN.getKey()).build())).containsExa
ctly(ANYONE, group1.getName());

    assertThat(underTest.selectGroupNamesByQuery(dbSession,
newQuery().setOrganizationUuid(organizationDto.getUuid()).setPermission(ADMINISTER.getKey()).build())).cont
ainsExactly(group3.getName());

    assertThat(underTest.selectGroupNamesByQuery(dbSession,
newQuery().setOrganizationUuid(organizationDto.getUuid()).setPermission(PROVISION_PROJECTS.getKey()).b
uild())).containsExactly(ANYONE);
  }

  @Test
  public void select_groups_by_query_with_project_permissions_on_public_projects() {
    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    GroupDto group3 = db.users().insertGroup();

    ComponentDto project = db.components().insertPublicProject();
    ComponentDto anotherProject = db.components().insertPublicProject();

    db.users().insertProjectPermissionOnGroup(group1, "p1", project);
    db.users().insertProjectPermissionOnGroup(group1, "p2", project);
    db.users().insertProjectPermissionOnAnyone("p3", project);

    db.users().insertProjectPermissionOnGroup(group1, "p4", anotherProject);
    db.users().insertProjectPermissionOnAnyone("p4", anotherProject);
    db.users().insertProjectPermissionOnGroup(group3, "p1", anotherProject);
    db.users().insertPermissionOnGroup(group2, "p5");

    PermissionQuery.Builder builderOnComponent = newQuery().setComponentUuid(project.uuid());
    assertThat(underTest.selectGroupNamesByQuery(dbSession,
      builderOnComponent.withAtLeastOnePermission().build())).containsOnlyOnce(group1.getName());
    assertThat(underTest.selectGroupNamesByQuery(dbSession,
      builderOnComponent.setPermission("p1").build())).containsOnlyOnce(group1.getName());
    assertThat(underTest.selectGroupNamesByQuery(dbSession,
      builderOnComponent.setPermission("p3").build())).containsOnlyOnce(ANYONE);
  }

  @Test
  public void select_groups_by_query_with_project_permissions_on_private_projects() {
    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    GroupDto group3 = db.users().insertGroup();

    ComponentDto project = db.components().insertPrivateProject();
```

```java
ComponentDto anotherProject = db.components().insertPrivateProject();

db.users().insertProjectPermissionOnGroup(group1, SCAN_EXECUTION, project);
db.users().insertProjectPermissionOnGroup(group1, PROVISIONING, project);

db.users().insertProjectPermissionOnGroup(group1, SYSTEM_ADMIN, anotherProject);
db.users().insertProjectPermissionOnGroup(group3, SCAN_EXECUTION, anotherProject);
db.users().insertPermissionOnGroup(group2, SCAN);

PermissionQuery.Builder builderOnComponent = newQuery().setComponentUuid(project.uuid());
assertThat(underTest.selectGroupNamesByQuery(dbSession,
  builderOnComponent.withAtLeastOnePermission().build())).containsOnlyOnce(group1.getName());
assertThat(underTest.selectGroupNamesByQuery(dbSession,
  builderOnComponent.setPermission(SCAN_EXECUTION).build())).containsOnlyOnce(group1.getName());
assertThat(underTest.selectGroupNamesByQuery(dbSession,
  builderOnComponent.setPermission(USER).build())).isEmpty();
}

@Test
public void selectGroupNamesByQuery_is_paginated() {
  IntStream.rangeClosed(0, 9).forEach(i -> db.users().insertGroup(db.getDefaultOrganization(), i + "-name"));

  List<String> groupNames = underTest.selectGroupNamesByQuery(dbSession,
    newQuery().setPageIndex(2).setPageSize(3).build());
  assertThat(groupNames).containsExactly("3-name", "4-name", "5-name");
}

@Test
public void selectGroupNamesByQuery_with_search_query() {
  GroupDto group = db.users().insertGroup(db.getDefaultOrganization(), "group-anyone");
  db.users().insertGroup(db.getDefaultOrganization(), "unknown");
  db.users().insertPermissionOnGroup(group, SCAN);

  assertThat(underTest.selectGroupNamesByQuery(dbSession,
    newQuery().setSearchQuery("any").build())).containsOnlyOnce(ANYONE, group.getName());
}

@Test
public void selectGroupNamesByQuery_does_not_return_anyone_when_group_roles_is_empty() {
  GroupDto group = db.users().insertGroup();

  assertThat(underTest.selectGroupNamesByQuery(dbSession,
    newQuery().build()))
      .doesNotContain(ANYONE)
      .containsExactly(group.getName());
}

@Test
```

```java
public void selectByGroupIds_on_global_permissions() {
  OrganizationDto organizationDto = db.organizations().insert();

  GroupDto group1 = db.users().insertGroup(organizationDto, "Group-1");
  db.users().insertPermissionOnGroup(group1, SCAN);

  GroupDto group2 = db.users().insertGroup(organizationDto, "Group-2");
  ComponentDto project =
db.components().insertComponent(ComponentTesting.newPrivateProjectDto(organizationDto));
  db.users().insertProjectPermissionOnGroup(group2, UserRole.ADMIN, project);

  GroupDto group3 = db.users().insertGroup(organizationDto, "Group-3");
  db.users().insertPermissionOnGroup(group3, ADMINISTER);

  // Anyone
  db.users().insertPermissionOnAnyone(organizationDto, SCAN);
  db.users().insertPermissionOnAnyone(organizationDto, PROVISION_PROJECTS);

  assertThat(underTest.selectByGroupIds(dbSession, organizationDto.getUuid(), asList(group1.getId()), null))
    .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
    .containsOnly(tuple(group1.getId(), SCAN_EXECUTION, null));

  assertThat(underTest.selectByGroupIds(dbSession, organizationDto.getUuid(), asList(group2.getId()),
null)).isEmpty();

  assertThat(underTest.selectByGroupIds(dbSession, organizationDto.getUuid(), asList(group3.getId()), null))
    .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
    .containsOnly(tuple(group3.getId(), SYSTEM_ADMIN, null));

  assertThat(underTest.selectByGroupIds(dbSession, organizationDto.getUuid(), asList(ANYONE_ID), null))
    .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
    .containsOnly(
      tuple(0, SCAN_EXECUTION, null),
      tuple(0, PROVISIONING, null));

  assertThat(underTest.selectByGroupIds(dbSession, organizationDto.getUuid(), asList(group1.getId(),
group2.getId(), ANYONE_ID), null)).hasSize(3);
  assertThat(underTest.selectByGroupIds(dbSession, organizationDto.getUuid(), asList(MISSING_ID),
null)).isEmpty();
  assertThat(underTest.selectByGroupIds(dbSession, organizationDto.getUuid(), Collections.emptyList(),
null)).isEmpty();
}

@Test
public void selectByGroupIds_on_public_projects() {
```

```java
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org, "Group-1");
    db.users().insertPermissionOnGroup(group1, "p1");

    GroupDto group2 = db.users().insertGroup(org, "Group-2");
    ComponentDto project = db.components().insertPublicProject(org);
    db.users().insertProjectPermissionOnGroup(group2, "p2", project);

    GroupDto group3 = db.users().insertGroup(org, "Group-3");
    db.users().insertProjectPermissionOnGroup(group3, "p2", project);

    // Anyone group
    db.users().insertPermissionOnAnyone(org, "p3");
    db.users().insertProjectPermissionOnAnyone("p4", project);

    assertThat(underTest.selectByGroupIds(dbSession, defaultOrganizationUuid, singletonList(group1.getId()),
project.getId())).isEmpty();

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(group2.getId()), project.getId()))
      .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
      .containsOnly(tuple(group2.getId(), "p2", project.getId()));

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(group3.getId()), project.getId()))
      .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
      .containsOnly(tuple(group3.getId(), "p2", project.getId()));

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(ANYONE_ID), project.getId()))
      .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
      .containsOnly(tuple(0, "p4", project.getId()));

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), asList(group1.getId(), group2.getId(),
ANYONE_ID), project.getId())).hasSize(2);
    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(MISSING_ID),
project.getId())).isEmpty();
    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(group1.getId()),
123L)).isEmpty();
    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), Collections.emptyList(),
project.getId())).isEmpty();
  }

  @Test
  public void selectByGroupIds_on_private_projects() {
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org, "Group-1");
    db.users().insertPermissionOnGroup(group1, PROVISION_PROJECTS);
```

```java
    GroupDto group2 = db.users().insertGroup(org, "Group-2");
    ComponentDto project = db.components().insertPrivateProject(org);
    db.users().insertProjectPermissionOnGroup(group2, USER, project);

    GroupDto group3 = db.users().insertGroup(org, "Group-3");
    db.users().insertProjectPermissionOnGroup(group3, USER, project);

    // Anyone group
    db.users().insertPermissionOnAnyone(org, SCAN);

    assertThat(underTest.selectByGroupIds(dbSession, defaultOrganizationUuid, singletonList(group1.getId()),
project.getId())).isEmpty();

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(group2.getId()), project.getId()))
      .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
      .containsOnly(tuple(group2.getId(), USER, project.getId()));

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(group3.getId()), project.getId()))
      .extracting(GroupPermissionDto::getGroupId, GroupPermissionDto::getRole,
GroupPermissionDto::getResourceId)
      .containsOnly(tuple(group3.getId(), USER, project.getId()));

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(ANYONE_ID), project.getId()))
      .isEmpty();

    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), asList(group1.getId(), group2.getId(),
ANYONE_ID), project.getId())).hasSize(1);
    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(MISSING_ID),
project.getId())).isEmpty();
    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), singletonList(group1.getId()),
123L)).isEmpty();
    assertThat(underTest.selectByGroupIds(dbSession, org.getUuid(), Collections.emptyList(),
project.getId())).isEmpty();
  }

  @Test
  public void selectGlobalPermissionsOfGroup() {
    OrganizationDto org1 = db.organizations().insert();
    OrganizationDto org2 = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org1, "group1");
    GroupDto group2 = db.users().insertGroup(org2, "group2");
    ComponentDto project = db.components().insertPublicProject(org1);

    db.users().insertPermissionOnAnyone(org1, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertPermissionOnGroup(group1, "perm3");
```

```java
      db.users().insertPermissionOnGroup(group2, "perm4");
      db.users().insertProjectPermissionOnGroup(group1, "perm5", project);
      db.users().insertProjectPermissionOnAnyone("perm6", project);

    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, org1.getUuid(),
group1.getId())).containsOnly("perm2", "perm3");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, org2.getUuid(),
group2.getId())).containsOnly("perm4");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, org1.getUuid(), null)).containsOnly("perm1");

    // group1 is not in org2
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, org2.getUuid(), group1.getId())).isEmpty();
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, org2.getUuid(), null)).isEmpty();
  }

  @Test
  public void selectProjectPermissionsOfGroup_on_public_project() {
    OrganizationDto org1 = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org1, "group1");
    ComponentDto project1 = db.components().insertPublicProject(org1);
    ComponentDto project2 = db.components().insertPublicProject(org1);

    db.users().insertPermissionOnAnyone(org1, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
    db.users().insertProjectPermissionOnGroup(group1, "perm4", project1);
    db.users().insertProjectPermissionOnGroup(group1, "perm5", project2);
    db.users().insertProjectPermissionOnAnyone("perm6", project1);

    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), group1.getId(),
project1.getId()))
      .containsOnly("perm3", "perm4");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), group1.getId(),
project2.getId()))
      .containsOnly("perm5");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), null, project1.getId()))
      .containsOnly("perm6");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), null, project2.getId()))
      .isEmpty();
  }

  @Test
  public void selectProjectPermissionsOfGroup_on_private_project() {
    OrganizationDto org1 = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org1, "group1");
    ComponentDto project1 = db.components().insertPrivateProject(org1);
    ComponentDto project2 = db.components().insertPrivateProject(org1);
```

```java
    db.users().insertPermissionOnAnyone(org1, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
    db.users().insertProjectPermissionOnGroup(group1, "perm4", project1);
    db.users().insertProjectPermissionOnGroup(group1, "perm5", project2);

    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), group1.getId(),
  project1.getId()))
      .containsOnly("perm3", "perm4");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), group1.getId(),
  project2.getId()))
      .containsOnly("perm5");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), null, project1.getId()))
      .isEmpty();
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, org1.getUuid(), null, project2.getId()))
      .isEmpty();
  }

  @Test
  public void selectAllPermissionsByGroupId_on_public_project() {
    OrganizationDto org1 = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org1, "group1");
    ComponentDto project1 = db.components().insertPublicProject(org1);
    ComponentDto project2 = db.components().insertPublicProject(org1);
    db.users().insertPermissionOnAnyone(org1, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
    db.users().insertProjectPermissionOnGroup(group1, "perm4", project1);
    db.users().insertProjectPermissionOnGroup(group1, "perm5", project2);
    db.users().insertProjectPermissionOnAnyone("perm6", project1);

    List<GroupPermissionDto> result = new ArrayList<>();
    underTest.selectAllPermissionsByGroupId(dbSession, org1.getUuid(), group1.getId(), context ->
  result.add((GroupPermissionDto) context.getResultObject()));
    assertThat(result).extracting(GroupPermissionDto::getResourceId, GroupPermissionDto::getRole).containsOnly(
      tuple(null, "perm2"),
      tuple(project1.getId(), "perm3"), tuple(project1.getId(), "perm4"), tuple(project2.getId(), "perm5"));
  }

  @Test
  public void selectAllPermissionsByGroupId_on_private_project() {
    OrganizationDto org1 = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org1, "group1");
    ComponentDto project1 = db.components().insertPrivateProject(org1);
    ComponentDto project2 = db.components().insertPrivateProject(org1);
    db.users().insertPermissionOnAnyone(org1, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
```

```java
    db.users().insertProjectPermissionOnGroup(group1, "perm4", project1);
    db.users().insertProjectPermissionOnGroup(group1, "perm5", project2);

  List<GroupPermissionDto> result = new ArrayList<>();
  underTest.selectAllPermissionsByGroupId(dbSession, org1.getUuid(), group1.getId(), context ->
result.add((GroupPermissionDto) context.getResultObject()));
  assertThat(result).extracting(GroupPermissionDto::getResourceId, GroupPermissionDto::getRole).containsOnly(
    tuple(null, "perm2"),
    tuple(project1.getId(), "perm3"), tuple(project1.getId(), "perm4"), tuple(project2.getId(), "perm5"));
}

@Test
public void selectGroupIdsWithPermissionOnProjectBut_returns_empty_if_project_does_not_exist() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = randomPublicOrPrivateProject(organization);
  GroupDto group = db.users().insertGroup(organization);
  db.users().insertProjectPermissionOnGroup(group, "foo", project);

  assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, 1234, UserRole.USER))
    .isEmpty();
}

@Test
public void
selectGroupIdsWithPermissionOnProjectBut_returns_only_groups_of_project_which_do_not_have_permission() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = randomPublicOrPrivateProject(organization);
  GroupDto group1 = db.users().insertGroup(organization);
  GroupDto group2 = db.users().insertGroup(organization);
  db.users().insertProjectPermissionOnGroup(group1, "p1", project);
  db.users().insertProjectPermissionOnGroup(group2, "p2", project);

  assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p2"))
    .containsOnly(group1.getId());
  assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p1"))
    .containsOnly(group2.getId());
  assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p3"))
    .containsOnly(group1.getId(), group2.getId());
}

@Test
public void
selectGroupIdsWithPermissionOnProjectBut_does_not_returns_group_AnyOne_of_project_when_it_does_not_have_permission() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertPublicProject(organization);
  GroupDto group1 = db.users().insertGroup(organization);
  GroupDto group2 = db.users().insertGroup(organization);
```

```java
    db.users().insertProjectPermissionOnGroup(group1, "p1", project);
    db.users().insertProjectPermissionOnGroup(group2, "p2", project);
    db.users().insertProjectPermissionOnAnyone("p2", project);

    assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p2"))
      .containsOnly(group1.getId());
    assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p1"))
      .containsOnly(group2.getId());
  }

  @Test
  public void
selectGroupIdsWithPermissionOnProjectBut_does_not_return_groups_which_have_no_permission_at_all_on_speci
fied_project() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = randomPublicOrPrivateProject(organization);
    GroupDto group1 = db.users().insertGroup(organization);
    GroupDto group2 = db.users().insertGroup(organization);
    GroupDto group3 = db.users().insertGroup(organization);
    db.users().insertProjectPermissionOnGroup(group1, "p1", project);
    db.users().insertProjectPermissionOnGroup(group2, "p2", project);

    assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p2"))
      .containsOnly(group1.getId());
    assertThat(underTest.selectGroupIdsWithPermissionOnProjectBut(dbSession, project.getId(), "p1"))
      .containsOnly(group2.getId());
  }

  @Test
  public void deleteByRootComponentId_on_private_project() {
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org);
    GroupDto group2 = db.users().insertGroup(org);
    ComponentDto project1 = db.components().insertPrivateProject(org);
    ComponentDto project2 = db.components().insertPrivateProject(org);
    db.users().insertPermissionOnGroup(group1, "perm1");
    db.users().insertProjectPermissionOnGroup(group1, "perm2", project1);
    db.users().insertProjectPermissionOnGroup(group2, "perm3", project2);

    underTest.deleteByRootComponentId(dbSession, project1.getId());
    dbSession.commit();

    assertThat(db.countSql("select count(id) from group_roles where resource_id=" + project1.getId())).isEqualTo(0);
    assertThat(db.countRowsOfTable("group_roles")).isEqualTo(2);
  }

  @Test
  public void deleteByRootComponentId_on_public_project() {
```

```java
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org);
    GroupDto group2 = db.users().insertGroup(org);
    ComponentDto project1 = db.components().insertPublicProject(org);
    ComponentDto project2 = db.components().insertPublicProject(org);
    db.users().insertPermissionOnGroup(group1, "perm1");
    db.users().insertProjectPermissionOnGroup(group1, "perm2", project1);
    db.users().insertProjectPermissionOnGroup(group2, "perm3", project2);
    db.users().insertProjectPermissionOnAnyone("perm4", project1);
    db.users().insertProjectPermissionOnAnyone("perm5", project2);

    underTest.deleteByRootComponentId(dbSession, project1.getId());
    dbSession.commit();

    assertThat(db.countSql("select count(id) from group_roles where resource_id=" + project1.getId())).isEqualTo(0);
    assertThat(db.countRowsOfTable("group_roles")).isEqualTo(3);
  }

  @Test
  public void delete_global_permission_from_group_on_public_project() {
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org);
    ComponentDto project1 = db.components().insertPublicProject(org);
    db.users().insertPermissionOnAnyone(org, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
    db.users().insertProjectPermissionOnAnyone("perm4", project1);

    underTest.delete(dbSession, "perm2", group1.getOrganizationUuid(), group1.getId(), null);
    dbSession.commit();

    assertThatNoPermission("perm2");
    assertThat(db.countRowsOfTable("group_roles")).isEqualTo(3);
  }

  @Test
  public void delete_global_permission_from_group_on_private_project() {
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org);
    ComponentDto project1 = db.components().insertPrivateProject(org);
    db.users().insertPermissionOnAnyone(org, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);

    underTest.delete(dbSession, "perm2", group1.getOrganizationUuid(), group1.getId(), null);
    dbSession.commit();

    assertThatNoPermission("perm2");
```

```java
    assertThat(db.countRowsOfTable("group_roles")).isEqualTo(2);
  }

  @Test
  public void delete_global_permission_from_anyone_on_public_project() {
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org);
    ComponentDto project1 = db.components().insertPublicProject(org);
    db.users().insertPermissionOnAnyone(org, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
    db.users().insertProjectPermissionOnAnyone("perm4", project1);

    underTest.delete(dbSession, "perm1", group1.getOrganizationUuid(), null, null);
    dbSession.commit();

    assertThatNoPermission("perm1");
    assertThat(db.countRowsOfTable("group_roles")).isEqualTo(3);
  }

  @Test
  public void delete_project_permission_from_group_on_private_project() {
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org);
    ComponentDto project1 = db.components().insertPrivateProject(org);
    db.users().insertPermissionOnAnyone(org, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);

    underTest.delete(dbSession, "perm3", group1.getOrganizationUuid(), group1.getId(), project1.getId());
    dbSession.commit();

    assertThatNoPermission("perm3");
    assertThat(db.countRowsOfTable("group_roles")).isEqualTo(2);
  }

  @Test
  public void delete_project_permission_from_group_on_public_project() {
    OrganizationDto org = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(org);
    ComponentDto project1 = db.components().insertPublicProject(org);
    db.users().insertPermissionOnAnyone(org, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
    db.users().insertProjectPermissionOnAnyone("perm4", project1);

    underTest.delete(dbSession, "perm3", group1.getOrganizationUuid(), group1.getId(), project1.getId());
    dbSession.commit();
```

```java
      assertThatNoPermission("perm3");
      assertThat(db.countRowsOfTable("group_roles")).isEqualTo(3);
    }

    @Test
    public void delete_project_permission_from_anybody_on_private_project() {
      OrganizationDto org = db.organizations().insert();
      GroupDto group1 = db.users().insertGroup(org);
      ComponentDto project1 = db.components().insertPublicProject(org);
      db.users().insertPermissionOnAnyone(org, "perm1");
      db.users().insertPermissionOnGroup(group1, "perm2");
      db.users().insertProjectPermissionOnGroup(group1, "perm3", project1);
      db.users().insertProjectPermissionOnAnyone("perm4", project1);

      underTest.delete(dbSession, "perm4", group1.getOrganizationUuid(), null, project1.getId());
      dbSession.commit();

      assertThatNoPermission("perm4");
      assertThat(db.countRowsOfTable("group_roles")).isEqualTo(3);
    }

    @Test
    public void deleteByOrganization_does_not_fail_on_empty_db() {
      underTest.deleteByOrganization(dbSession, "some uuid");
      dbSession.commit();
    }

    @Test
    public void deleteByOrganization_does_not_fail_if_organization_has_no_group() {
      OrganizationDto organization = db.organizations().insert();

      underTest.deleteByOrganization(dbSession, organization.getUuid());
      dbSession.commit();
    }

    @Test
    public void deleteByOrganization_deletes_all_groups_of_organization() {
      OrganizationDto organization1 = db.organizations().insert();
      OrganizationDto organization2 = db.organizations().insert();
      OrganizationDto organization3 = db.organizations().insert();
      insertGroupWithPermissions(organization1);
      insertGroupWithPermissions(organization2);
      insertGroupWithPermissions(organization3);
      insertGroupWithPermissions(organization3);
      insertGroupWithPermissions(organization2);
      db.users().insertPermissionOnAnyone(organization1, "pop");
      db.users().insertPermissionOnAnyone(organization2, "pop");
```

```java
    db.users().insertPermissionOnAnyone(organization3, "pop");

    underTest.deleteByOrganization(dbSession, organization2.getUuid());
    dbSession.commit();
    verifyOrganizationUuidsInTable(organization1.getUuid(), organization3.getUuid());

    underTest.deleteByOrganization(dbSession, organization1.getUuid());
    dbSession.commit();
    verifyOrganizationUuidsInTable(organization3.getUuid());

    underTest.deleteByOrganization(dbSession, organization3.getUuid());
    dbSession.commit();
    verifyOrganizationUuidsInTable();
  }

  @Test
  public void
deleteByRootComponentIdAndGroupId_deletes_all_permissions_of_group_AnyOne_of_specified_component_if_g
roupId_is_null() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    GroupDto group = db.users().insertGroup(organization);
    db.users().insertProjectPermissionOnAnyone("p1", project);
    db.users().insertProjectPermissionOnGroup(group, "p2", project);
    db.users().insertPermissionOnAnyone(organization, "p3");
    db.users().insertPermissionOnGroup(group, "p4");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), null, project.getId()))
      .containsOnly("p1");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group.getId(),
project.getId()))
      .containsOnly("p2");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), null))
      .containsOnly("p3");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group.getId()))
      .containsOnly("p4");

    int deletedCount = underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(), null);

    assertThat(deletedCount).isEqualTo(1);
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), null, project.getId()))
      .isEmpty();
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group.getId(),
project.getId()))
      .containsOnly("p2");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), null))
      .containsOnly("p3");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group.getId()))
      .containsOnly("p4");
```

```java
  }

  @Test
  public void
deleteByRootComponentIdAndGroupId_deletes_all_permissions_of_specified_group_of_specified_component_if_
groupId_is_non_null() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    GroupDto group1 = db.users().insertGroup(organization);
    GroupDto group2 = db.users().insertGroup(organization);
    db.users().insertProjectPermissionOnAnyone("p1", project);
    db.users().insertProjectPermissionOnGroup(group1, "p2", project);
    db.users().insertProjectPermissionOnGroup(group2, "p3", project);
    db.users().insertProjectPermissionOnGroup(group2, "p4", project);
    db.users().insertPermissionOnAnyone(organization, "p5");
    db.users().insertPermissionOnGroup(group1, "p6");
    db.users().insertPermissionOnGroup(group2, "p7");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), null, project.getId()))
      .containsOnly("p1");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId(),
project.getId()))
      .containsOnly("p2");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group2.getId(),
project.getId()))
      .containsOnly("p3", "p4");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), null))
      .containsOnly("p5");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId()))
      .containsOnly("p6");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group2.getId()))
      .containsOnly("p7");

    int deletedCount = underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(), group1.getId());

    assertThat(deletedCount).isEqualTo(1);
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), null, project.getId()))
      .containsOnly("p1");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId(),
project.getId()))
      .isEmpty();
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group2.getId(),
project.getId()))
      .containsOnly("p3", "p4");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId()))
      .containsOnly("p6");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group2.getId()))
      .containsOnly("p7");
```

```java
    deletedCount = underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(), group2.getId());

    assertThat(deletedCount).isEqualTo(2);
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), null, project.getId()))
      .containsOnly("p1");
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId(),
project.getId()))
      .isEmpty();
    assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group2.getId(),
project.getId()))
      .isEmpty();
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId()))
      .containsOnly("p6");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group2.getId()))
      .containsOnly("p7");
  }

  @Test
  public void deleteByRootComponentIdAndGroupId_has_no_effect_if_component_does_not_exist() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);

    assertThat(underTest.deleteByRootComponentIdAndGroupId(dbSession, 1234L, null)).isEqualTo(0);
    assertThat(underTest.deleteByRootComponentIdAndGroupId(dbSession, 1234L, group.getId())).isEqualTo(0);
  }

  @Test
  public void
deleteByRootComponentIdAndGroupId_has_no_effect_if_component_has_no_group_permission_at_all() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = randomPublicOrPrivateProject(organization);
    GroupDto group = db.users().insertGroup(organization);

    assertThat(underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(), null)).isEqualTo(0);
    assertThat(underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(),
group.getId())).isEqualTo(0);
  }

  @Test
  public void deleteByRootComponentIdAndGroupId_has_no_effect_if_group_does_not_exist() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = randomPublicOrPrivateProject(organization);

    assertThat(underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(), 5678)).isEqualTo(0);
  }

  @Test
  public void
```

```java
deleteByRootComponentIdAndGroupId_has_no_effect_if_component_has_no_group_permission_for_group_AnyO
ne() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertPrivateProject(organization);
  GroupDto group1 = db.users().insertGroup(organization);
  db.users().insertProjectPermissionOnGroup(group1, "p1", project);
  assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), null, project.getId()))
    .isEmpty();
  assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId(),
project.getId()))
    .containsOnly("p1");
  db.users().insertPermissionOnAnyone(organization, "p2");
  db.users().insertPermissionOnGroup(group1, "p3");

  int deletedCount = underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(), null);

  assertThat(deletedCount).isEqualTo(0);
  assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), null, project.getId()))
    .isEmpty();
  assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId(),
project.getId()))
    .containsOnly("p1");
  assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), null))
    .containsOnly("p2");
  assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId()))
    .containsOnly("p3");
}

@Test
public void
deleteByRootComponentIdAndGroupId_has_no_effect_if_component_has_no_group_permission_for_specified_gr
oup() {
  OrganizationDto organization = db.organizations().insert();
  ComponentDto project = db.components().insertPrivateProject(organization);
  GroupDto group1 = db.users().insertGroup(organization);
  GroupDto group2 = db.users().insertGroup(organization);
  db.users().insertProjectPermissionOnGroup(group1, "p1", project);
  db.users().insertPermissionOnAnyone(organization, "p2");
  db.users().insertPermissionOnGroup(group1, "p3");

  int deletedCount = underTest.deleteByRootComponentIdAndGroupId(dbSession, project.getId(), group2.getId());

  assertThat(deletedCount).isEqualTo(0);
  assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId(),
project.getId()))
    .containsOnly("p1");
  assertThat(underTest.selectProjectPermissionsOfGroup(dbSession, organization.getUuid(), group2.getId(),
project.getId()))
```

```
      .isEmpty();
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), null))
      .containsOnly("p2");
    assertThat(underTest.selectGlobalPermissionsOfGroup(dbSession, organization.getUuid(), group1.getId()))
      .containsOnly("p3");
  }

  @Test
  public void
  deleteByRootComponentIdAndPermission_deletes_all_rows_for_specified_role_of_specified_component() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    GroupDto group = db.users().insertGroup(organization);
    Stream.of("p1", "p2").forEach(permission -> {
      db.users().insertPermissionOnAnyone(organization, permission);
      db.users().insertPermissionOnGroup(group, permission);
      db.users().insertProjectPermissionOnGroup(group, permission, project);
      db.users().insertProjectPermissionOnAnyone(permission, project);
    });
    assertThat(getGlobalPermissionsForAnyone(organization)).containsOnly("p1", "p2");
    assertThat(getGlobalPermissionsForGroup(group)).containsOnly("p1", "p2");
    assertThat(getProjectPermissionsForAnyOne(project)).containsOnly("p1", "p2");
    assertThat(getProjectPermissionsForGroup(project, group)).containsOnly("p1", "p2");

    int deletedRows = underTest.deleteByRootComponentIdAndPermission(dbSession, project.getId(), "p1");

    assertThat(deletedRows).isEqualTo(2);
    assertThat(getGlobalPermissionsForAnyone(organization)).containsOnly("p1", "p2");
    assertThat(getGlobalPermissionsForGroup(group)).containsOnly("p1", "p2");
    assertThat(getProjectPermissionsForAnyOne(project)).containsOnly("p2");
    assertThat(getProjectPermissionsForGroup(project, group)).containsOnly("p2");

    deletedRows = underTest.deleteByRootComponentIdAndPermission(dbSession, project.getId(), "p2");

    assertThat(deletedRows).isEqualTo(2);
    assertThat(getGlobalPermissionsForAnyone(organization)).containsOnly("p1", "p2");
    assertThat(getGlobalPermissionsForGroup(group)).containsOnly("p1", "p2");
    assertThat(getProjectPermissionsForAnyOne(project)).isEmpty();
    assertThat(getProjectPermissionsForGroup(project, group)).isEmpty();
  }

  @Test
  public void
  deleteByRootComponentIdAndPermission_has_no_effect_if_component_has_no_group_permission_at_all() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = randomPublicOrPrivateProject(organization);
    db.users().insertPermissionOnAnyone(organization, "p1");
```

```java
    db.users().insertPermissionOnGroup(group, "p1");

    assertThat(underTest.deleteByRootComponentIdAndPermission(dbSession, project.getId(), "p1")).isEqualTo(0);

    assertThat(getGlobalPermissionsForAnyone(organization)).containsOnly("p1");
    assertThat(getGlobalPermissionsForGroup(group)).containsOnly("p1");
    assertThat(getProjectPermissionsForAnyOne(project)).isEmpty();
    assertThat(getProjectPermissionsForGroup(project, group)).isEmpty();
  }

  @Test
  public void deleteByRootComponentIdAndPermission_has_no_effect_if_component_does_not_exist() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);
    GroupDto group = db.users().insertGroup(organization);
    db.users().insertPermissionOnAnyone(organization, "p1");
    db.users().insertPermissionOnGroup(group, "p1");
    db.users().insertProjectPermissionOnGroup(group, "p1", project);
    db.users().insertProjectPermissionOnAnyone("p1", project);

    assertThat(underTest.deleteByRootComponentIdAndPermission(dbSession, 1324, "p1")).isEqualTo(0);

    assertThat(getGlobalPermissionsForAnyone(organization)).containsOnly("p1");
    assertThat(getGlobalPermissionsForGroup(group)).containsOnly("p1");
    assertThat(getProjectPermissionsForAnyOne(project)).containsOnly("p1");
    assertThat(getProjectPermissionsForGroup(project, group)).containsOnly("p1");
  }

  @Test
  public void
  deleteByRootComponentIdAndPermission_has_no_effect_if_component_does_not_have_specified_permission() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group = db.users().insertGroup(organization);
    ComponentDto project = randomPublicOrPrivateProject(organization);
    db.users().insertPermissionOnAnyone(organization, "p1");
    db.users().insertPermissionOnGroup(group, "p1");

    assertThat(underTest.deleteByRootComponentIdAndPermission(dbSession, project.getId(), "p1")).isEqualTo(0);
  }

  private Collection<String> getGlobalPermissionsForAnyone(OrganizationDto organization) {
    return getPermissions("organization_uuid = '" + organization.getUuid() + "' and group_id is null and resource_id is
null");
  }

  private Collection<String> getGlobalPermissionsForGroup(GroupDto groupDto) {
    return getPermissions("organization_uuid = '" + groupDto.getOrganizationUuid() + "' and group_id = " +
groupDto.getId() + " and resource_id is null");
```

```java
  }

  private Collection<String> getProjectPermissionsForAnyOne(ComponentDto project) {
    return getPermissions("organization_uuid = '" + project.getOrganizationUuid() + "' and group_id is null and
resource_id = " + project.getId());
  }

  private Collection<String> getProjectPermissionsForGroup(ComponentDto project, GroupDto group) {
    return getPermissions("organization_uuid = '" + project.getOrganizationUuid() + "' and group_id = " +
group.getId() + " and resource_id = " + project.getId());
  }

  private Collection<String> getPermissions(String whereClauses) {
    return db
      .select(dbSession, "select role from group_roles where " + whereClauses)
      .stream()
      .flatMap(map -> map.entrySet().stream())
      .map(entry -> (String) entry.getValue())
      .collect(MoreCollectors.toList());
  }

  private ComponentDto randomPublicOrPrivateProject(OrganizationDto organization) {
    return new Random().nextBoolean() ? db.components().insertPublicProject(organization) :
db.components().insertPrivateProject(organization);
  }

  private PermissionQuery.Builder newQuery() {
    return PermissionQuery.builder().setOrganizationUuid(db.getDefaultOrganization().getUuid());
  }

  private void verifyOrganizationUuidsInTable(String... organizationUuids) {
    assertThat(db.select("select distinct organization_uuid as \"organizationUuid\" from group_roles"))
      .extracting((row) -> (String) row.get("organizationUuid"))
      .containsOnly(organizationUuids);
  }

  private int insertGroupWithPermissions(OrganizationDto organization1) {
    GroupDto group = db.users().insertGroup(organization1);
    db.users().insertPermissionOnGroup(group, "foo");
    db.users().insertPermissionOnGroup(group, "bar");
    db.users().insertPermissionOnGroup(group, "doh");
    return group.getId();
  }

  private void assertThatNoPermission(String permission) {
    assertThat(db.countSql("select count(id) from group_roles where role='" + permission + "'")).isEqualTo(0);
  }
```

```
}
```

```
package org.sonar.db.permission.template;

import java.util.Collections;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.UserDto;

import static java.util.Arrays.asList;
import static java.util.Collections.singletonList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.tuple;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.db.permission.PermissionQuery.builder;

public class UserWithPermissionTemplateDaoTest {

  @Rule
  public DbTester db = DbTester.create(System2.INSTANCE);

  private DbSession dbSession = db.getSession();

  private PermissionTemplateDao underTest = db.getDbClient().permissionTemplateDao();
```

```java
@Test
public void select_logins() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user1 = db.users().insertUser();
  UserDto user2 = db.users().insertUser();
  UserDto user3 = db.users().insertUser();
  db.organizations().addMember(organization, user1, user2, user3);
  PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, USER);
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, ADMIN);
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, CODEVIEWER);
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user2, USER);
  PermissionTemplateDto anotherPermissionTemplate = db.permissionTemplates().insertTemplate();
  db.permissionTemplates().addUserToTemplate(anotherPermissionTemplate, user1, USER);

  assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
    builder().setOrganizationUuid(organization.getUuid()).build(),
    permissionTemplate.getId()))
      .containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin(), user3.getLogin());
  assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setPermission(USER).build(),
  permissionTemplate.getId()))
      .containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin());
}

@Test
public void return_no_logins_on_unknown_template_key() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user = db.users().insertUser();
  db.organizations().addMember(organization, user);
  PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user, USER);

  assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
builder().setOrganizationUuid(organization.getUuid()).setPermission(USER).withAtLeastOnePermission().build(),
999L))
      .isEmpty();
}

@Test
public void select_only_logins_with_permission() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user1 = db.users().insertUser();
  UserDto user2 = db.users().insertUser();
  UserDto user3 = db.users().insertUser();
  db.organizations().addMember(organization, user1, user2, user3);
  PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
```

```
      db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, USER);
      db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, ADMIN);
      db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, CODEVIEWER);
      db.permissionTemplates().addUserToTemplate(permissionTemplate, user2, USER);
      PermissionTemplateDto anotherPermissionTemplate = db.permissionTemplates().insertTemplate();
      db.permissionTemplates().addUserToTemplate(anotherPermissionTemplate, user1, USER);

      assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
   builder().setOrganizationUuid(organization.getUuid()).setPermission(USER).withAtLeastOnePermission().build(),
      permissionTemplate.getId()))
         .containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin());
   }

   @Test
   public void select_only_enable_users() {
      OrganizationDto organization = db.organizations().insert();
      UserDto user = db.users().insertUser();
      UserDto disabledUser = db.users().insertUser(u -> u.setActive(false));
      db.organizations().addMember(organization, user, disabledUser);
      PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
      db.permissionTemplates().addUserToTemplate(permissionTemplate, user, USER);
      db.permissionTemplates().addUserToTemplate(permissionTemplate, disabledUser, USER);

      assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
        builder().setOrganizationUuid(organization.getUuid()).setPermission(USER).build(),
   permissionTemplate.getId()))
         .containsExactlyInAnyOrder(user.getLogin());
   }

   @Test
   public void search_by_user_name() {
      OrganizationDto organization = db.organizations().insert();
      UserDto user1 = db.users().insertUser(u -> u.setName("User1"));
      UserDto user2 = db.users().insertUser(u -> u.setName("User2"));
      UserDto user3 = db.users().insertUser(u -> u.setName("User3"));
      db.organizations().addMember(organization, user1, user2, user3);
      PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
      db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, USER);
      db.permissionTemplates().addUserToTemplate(permissionTemplate, user2, USER);

      assertThat(underTest.selectUserLoginsByQueryAndTemplate(
        dbSession,
   builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setPermission(USER).setSearc
   hQuery("SEr1").build(),
        permissionTemplate.getId()))
         .containsExactlyInAnyOrder(user1.getLogin());

      assertThat(underTest.selectUserLoginsByQueryAndTemplate(
```

```java
        dbSession,
    builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setPermission(USER).setSearc
hQuery("user").build(),
      permissionTemplate.getId()))
        .containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin());
  }

  @Test
  public void should_be_sorted_by_user_name() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = db.users().insertUser(u -> u.setName("User3"));
    UserDto user2 = db.users().insertUser(u -> u.setName("User1"));
    UserDto user3 = db.users().insertUser(u -> u.setName("User2"));
    db.organizations().addMember(organization, user1, user2, user3);
    PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
    db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, USER);
    db.permissionTemplates().addUserToTemplate(permissionTemplate, user2, USER);

    assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
      builder().setOrganizationUuid(organization.getUuid()).build(), permissionTemplate.getId()))
        .containsExactly(user2.getLogin(), user3.getLogin(), user1.getLogin());
  }

  @Test
  public void should_be_paginated() {
    OrganizationDto organization = db.organizations().insert();
    UserDto user1 = db.users().insertUser(u -> u.setName("User1"));
    UserDto user2 = db.users().insertUser(u -> u.setName("User2"));
    UserDto user3 = db.users().insertUser(u -> u.setName("User3"));
    db.organizations().addMember(organization, user1, user2, user3);
    PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
    db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, USER);
    db.permissionTemplates().addUserToTemplate(permissionTemplate, user2, USER);

    assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
      builder().setOrganizationUuid(organization.getUuid()).setPageIndex(1).setPageSize(2).build(),
permissionTemplate.getId()))
        .containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin());
    assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
      builder().setOrganizationUuid(organization.getUuid()).setPageIndex(2).setPageSize(2).build(),
permissionTemplate.getId()))
        .containsExactlyInAnyOrder(user3.getLogin());
    assertThat(underTest.selectUserLoginsByQueryAndTemplate(dbSession,
      builder().setOrganizationUuid(organization.getUuid()).setPageIndex(3).setPageSize(1).build(),
permissionTemplate.getId()))
        .containsExactlyInAnyOrder(user3.getLogin());
  }
```

```java
@Test
public void count_users() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user1 = db.users().insertUser();
  UserDto user2 = db.users().insertUser();
  UserDto user3 = db.users().insertUser();
  db.organizations().addMember(organization, user1, user2, user3);
  PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, USER);
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user2, USER);

  assertThat(underTest.countUserLoginsByQueryAndTemplate(dbSession,
    builder().setOrganizationUuid(organization.getUuid()).build(), permissionTemplate.getId()))
      .isEqualTo(3);
  assertThat(underTest.countUserLoginsByQueryAndTemplate(dbSession,
builder().setOrganizationUuid(organization.getUuid()).withAtLeastOnePermission().setPermission("user").build(), p
ermissionTemplate.getId()))
      .isEqualTo(2);
}

@Test
public void select_user_permission_templates_by_template_and_logins() {
  OrganizationDto organization = db.organizations().insert();
  UserDto user1 = db.users().insertUser();
  UserDto user2 = db.users().insertUser();
  UserDto user3 = db.users().insertUser();
  db.organizations().addMember(organization, user1, user2, user3);
  PermissionTemplateDto permissionTemplate = db.permissionTemplates().insertTemplate();
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, USER);
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, ADMIN);
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user1, CODEVIEWER);
  db.permissionTemplates().addUserToTemplate(permissionTemplate, user2, USER);
  PermissionTemplateDto anotherPermissionTemplate = db.permissionTemplates().insertTemplate();
  db.permissionTemplates().addUserToTemplate(anotherPermissionTemplate, user1, USER);

  assertThat(underTest.selectUserPermissionsByTemplateIdAndUserLogins(dbSession,
permissionTemplate.getId(), singletonList(user1.getLogin())))
    .extracting(PermissionTemplateUserDto::getUserLogin, PermissionTemplateUserDto::getPermission)
    .containsExactlyInAnyOrder(
      tuple(user1.getLogin(), USER),
      tuple(user1.getLogin(), ADMIN),
      tuple(user1.getLogin(), CODEVIEWER));

  assertThat(underTest.selectUserPermissionsByTemplateIdAndUserLogins(dbSession,
permissionTemplate.getId(), asList(user1.getLogin(), user2.getLogin(), user2.getLogin())))
    .extracting(PermissionTemplateUserDto::getUserLogin, PermissionTemplateUserDto::getPermission)
    .containsExactlyInAnyOrder(
      tuple(user1.getLogin(), USER),
```

```
        tuple(user1.getLogin(), ADMIN),
        tuple(user1.getLogin(), CODEVIEWER),
        tuple(user2.getLogin(), USER));

    assertThat(underTest.selectUserPermissionsByTemplateIdAndUserLogins(dbSession,
permissionTemplate.getId(), singletonList("unknown"))).isEmpty();
    assertThat(underTest.selectUserPermissionsByTemplateIdAndUserLogins(dbSession,
permissionTemplate.getId(), Collections.emptyList())).isEmpty();
    assertThat(underTest.selectUserPermissionsByTemplateIdAndUserLogins(dbSession, 123L,
singletonList(user1.getLogin()))).isEmpty();
 }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import javax.annotation.Nullable;
import org.sonar.db.DbClient;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;

import static
org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateCharacteristicDto;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateDto;

public class PermissionTemplateDbTester {
 private final DbTester db;
 private final DbClient dbClient;
```

```java
private final DbSession dbSession;

public PermissionTemplateDbTester(DbTester db) {
  this.db = db;
  this.dbClient = db.getDbClient();
  this.dbSession = db.getSession();
}

public PermissionTemplateDto insertTemplate() {
  return insertTemplate(newPermissionTemplateDto());
}

public PermissionTemplateDto insertTemplate(OrganizationDto organizationDto) {
  return insertTemplate(newPermissionTemplateDto().setOrganizationUuid(organizationDto.getUuid()));
}

public PermissionTemplateDto insertTemplate(PermissionTemplateDto template) {
  PermissionTemplateDto templateInDb = dbClient.permissionTemplateDao().insert(dbSession, template);
  db.commit();

  return templateInDb;
}

public void addGroupToTemplate(PermissionTemplateDto permissionTemplate, GroupDto group, String
permission) {
  addGroupToTemplate(permissionTemplate.getId(), group.getId(), permission);
}

public void addGroupToTemplate(long templateId, @Nullable Integer groupId, String permission) {
  dbClient.permissionTemplateDao().insertGroupPermission(dbSession, templateId, groupId, permission);
  db.commit();
}

public void addAnyoneToTemplate(PermissionTemplateDto permissionTemplate, String permission) {
  addGroupToTemplate(permissionTemplate.getId(), null, permission);
}

public void addUserToTemplate(PermissionTemplateDto permissionTemplate, UserDto user, String permission) {
  addUserToTemplate(permissionTemplate.getId(), user.getId(), permission);
}

public void addUserToTemplate(long templateId, int userId, String permission) {
  dbClient.permissionTemplateDao().insertUserPermission(dbSession, templateId, userId, permission);
  db.commit();
}

public void addProjectCreatorToTemplate(PermissionTemplateDto permissionTemplate, String permission) {
  addProjectCreatorToTemplate(permissionTemplate.getId(), permission);
```

```java
  }

  public void addProjectCreatorToTemplate(long templateId, String permission) {
    dbClient.permissionTemplateCharacteristicDao().insert(dbSession, newPermissionTemplateCharacteristicDto()
      .setWithProjectCreator(true)
      .setTemplateId(templateId)
      .setPermission(permission));
    db.commit();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;

import static com.google.common.primitives.Longs.asList;
import static org.assertj.core.api.Assertions.assertThat;
```

```java
import static org.assertj.core.api.Assertions.tuple;
import static org.mockito.Mockito.mock;
import static org.mockito.Mockito.when;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.CODEVIEWER;
import static org.sonar.api.web.UserRole.ISSUE_ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.SCAN_EXECUTION;
import static org.sonar.db.permission.template.PermissionTemplateTesting.newPermissionTemplateDto;
import static org.sonar.db.user.GroupTesting.newGroupDto;

public class PermissionTemplateDaoTest {

  private static final Date PAST = new Date(100_000_000_000L);
  private static final Date NOW = new Date(500_000_000_000L);

  @Rule
  public ExpectedException expectedException = ExpectedException.none();

  @Rule
  public DbTester db = DbTester.create();

  private System2 system2 = mock(System2.class);
  private DbSession dbSession = db.getSession();
  private PermissionTemplateDbTester templateDb = db.permissionTemplates();

  private PermissionTemplateDao underTest = new PermissionTemplateDao(system2);

  @Before
  public void setUp() throws Exception {
    when(system2.now()).thenReturn(NOW.getTime());
  }

  @Test
  public void should_create_permission_template() {
    PermissionTemplateDto permissionTemplate = underTest.insert(db.getSession(), newPermissionTemplateDto()
      .setUuid("ABCD")
      .setName("my template")
      .setDescription("my description")
      .setKeyPattern("myregexp")
      .setOrganizationUuid("org")
      .setCreatedAt(PAST)
      .setUpdatedAt(NOW));
    db.commit();

    assertThat(underTest.selectByUuid(db.getSession(), permissionTemplate.getUuid()))
      .extracting(PermissionTemplateDto::getUuid, PermissionTemplateDto::getName,
PermissionTemplateDto::getDescription, PermissionTemplateDto::getKeyPattern,
```

```java
      PermissionTemplateDto::getOrganizationUuid, PermissionTemplateDto::getCreatedAt,
PermissionTemplateDto::getUpdatedAt)
      .containsOnly("ABCD", "my template", "my description", "myregexp", "org", PAST, NOW);
  }

  @Test
  public void should_select_permission_template_by_uuid() {
    templateDb.insertTemplate(newPermissionTemplateDto()
      .setUuid("ABCD")
      .setName("my template")
      .setDescription("my description")
      .setKeyPattern("myregexp")
      .setOrganizationUuid("org"));

    assertThat(underTest.selectByUuid(db.getSession(), "ABCD"))
      .extracting(PermissionTemplateDto::getUuid, PermissionTemplateDto::getName,
PermissionTemplateDto::getDescription, PermissionTemplateDto::getKeyPattern,
        PermissionTemplateDto::getOrganizationUuid)
      .containsOnly("ABCD", "my template", "my description", "myregexp", "org");
  }

  @Test
  public void selectAll_without_name_filtering() {
    templateDb.insertTemplate(newPermissionTemplateDto()
      .setUuid("tpl1")
      .setName("template1")
      .setDescription("description1")
      .setOrganizationUuid("org"));
    templateDb.insertTemplate(newPermissionTemplateDto()
      .setUuid("tpl2")
      .setName("template2")
      .setDescription("description2")
      .setOrganizationUuid("org"));
    templateDb.insertTemplate(newPermissionTemplateDto()
      .setUuid("tpl3")
      .setName("template3")
      .setDescription("description3")
      .setOrganizationUuid("org"));

    assertThat(underTest.selectAll(dbSession, "org", null))
      .extracting(PermissionTemplateDto::getUuid, PermissionTemplateDto::getName,
PermissionTemplateDto::getDescription)
      .containsOnly(
        tuple("tpl1", "template1", "description1"),
        tuple("tpl2", "template2", "description2"),
        tuple("tpl3", "template3", "description3"));
    assertThat(underTest.selectAll(dbSession, "missingOrg", null)).isEmpty();
  }
```

```java
  @Test
  public void selectAll_with_name_filtering() {
    PermissionTemplateDto t1InOrg1 =
templateDb.insertTemplate(newPermissionTemplateDto().setName("aBcDeF").setOrganizationUuid("org1"));
    PermissionTemplateDto t2InOrg1 =
templateDb.insertTemplate(newPermissionTemplateDto().setName("cdefgh").setOrganizationUuid("org1"));
    PermissionTemplateDto t3InOrg1 =
templateDb.insertTemplate(newPermissionTemplateDto().setName("hijkl").setOrganizationUuid("org2"));
    PermissionTemplateDto t4InOrg2 =
templateDb.insertTemplate(newPermissionTemplateDto().setName("cdefgh").setOrganizationUuid("org2"));

    assertThat(underTest.selectAll(dbSession, "org1",
"def")).extracting(PermissionTemplateDto::getId).containsExactly(t1InOrg1.getId(), t2InOrg1.getId());
    assertThat(underTest.selectAll(dbSession, "org1", "missing")).isEmpty();
  }

  @Test
  public void should_update_permission_template() {
    PermissionTemplateDto permissionTemplateDto = templateDb.insertTemplate(newPermissionTemplateDto()
      .setUuid("ABCD")
      .setName("name")
      .setDescription("description")
      .setKeyPattern("regexp")
      .setOrganizationUuid("org")
      .setCreatedAt(PAST)
      .setUpdatedAt(PAST));

    underTest.update(dbSession, permissionTemplateDto
      .setName("new_name")
      .setDescription("new_description")
      .setKeyPattern("new_regexp")
      .setUpdatedAt(NOW)
      // Invariant fields, should not be updated
      .setUuid("new UUID")
      .setOrganizationUuid("new org")
      .setCreatedAt(NOW));
    db.commit();

    assertThat(underTest.selectByUuid(db.getSession(), "ABCD"))
      .extracting(PermissionTemplateDto::getUuid, PermissionTemplateDto::getName,
PermissionTemplateDto::getDescription, PermissionTemplateDto::getKeyPattern,
       PermissionTemplateDto::getOrganizationUuid, PermissionTemplateDto::getCreatedAt,
PermissionTemplateDto::getUpdatedAt)
      .containsOnly("ABCD", "new_name", "new_description", "new_regexp", "org", PAST, NOW);
  }

  @Test
```

```java
public void should_delete_permission_template() {
  UserDto user1 = db.users().insertUser();
  UserDto user2 = db.users().insertUser();
  GroupDto group1 = db.users().insertGroup();
  GroupDto group2 = db.users().insertGroup();
  PermissionTemplateDto permissionTemplate1 = templateDb.insertTemplate(db.getDefaultOrganization());
  PermissionTemplateDto permissionTemplate2 = templateDb.insertTemplate(db.getDefaultOrganization());
  templateDb.addUserToTemplate(permissionTemplate1, user1, "user");
  templateDb.addUserToTemplate(permissionTemplate1, user2, "user");
  templateDb.addUserToTemplate(permissionTemplate1, user2, "admin");
  templateDb.addUserToTemplate(permissionTemplate2, user2, "admin");
  templateDb.addGroupToTemplate(permissionTemplate1, group1, "user");
  templateDb.addGroupToTemplate(permissionTemplate1, group2, "user");
  templateDb.addAnyoneToTemplate(permissionTemplate1, "admin");
  templateDb.addAnyoneToTemplate(permissionTemplate2, "admin");
  templateDb.addProjectCreatorToTemplate(permissionTemplate1.getId(), "user");
  templateDb.addProjectCreatorToTemplate(permissionTemplate2.getId(), "user");

  underTest.deleteById(dbSession, permissionTemplate1.getId());
  dbSession.commit();

  assertThat(underTest.selectAll(db.getSession(), db.getDefaultOrganization().getUuid(), null))
    .extracting(PermissionTemplateDto::getUuid)
    .containsOnly(permissionTemplate2.getUuid());
  assertThat(db.getDbClient().permissionTemplateDao().selectUserPermissionsByTemplateId(db.getSession(),
permissionTemplate1.getId())).isEmpty();
  assertThat(db.getDbClient().permissionTemplateDao().selectUserPermissionsByTemplateId(db.getSession(),
permissionTemplate2.getId())).hasSize(1);
  assertThat(db.getDbClient().permissionTemplateDao().selectGroupPermissionsByTemplateId(db.getSession(),
permissionTemplate1.getId())).isEmpty();
  assertThat(db.getDbClient().permissionTemplateDao().selectGroupPermissionsByTemplateId(db.getSession(),
permissionTemplate2.getId())).hasSize(1);
  assertThat(db.getDbClient().permissionTemplateCharacteristicDao().selectByTemplateIds(db.getSession(),
asList(permissionTemplate1.getId(), permissionTemplate2.getId())))
    .extracting(PermissionTemplateCharacteristicDto::getTemplateId)
    .containsOnly(permissionTemplate2.getId());
}

@Test
public void should_add_user_permission_to_template() {
  PermissionTemplateDto permissionTemplate = templateDb.insertTemplate(db.getDefaultOrganization());
  UserDto user = db.users().insertUser();

  underTest.insertUserPermission(dbSession, permissionTemplate.getId(), user.getId(), "user");

  assertThat(db.getDbClient().permissionTemplateDao().selectUserPermissionsByTemplateId(db.getSession(),
permissionTemplate.getId()))
    .extracting(PermissionTemplateUserDto::getTemplateId, PermissionTemplateUserDto::getUserId,
```

```java
    PermissionTemplateUserDto::getPermission, PermissionTemplateUserDto::getCreatedAt,
        PermissionTemplateUserDto::getUpdatedAt)
      .containsOnly(tuple(permissionTemplate.getId(), user.getId(), "user", NOW, NOW));
  }

  @Test
  public void should_remove_user_permission_from_template() {
    PermissionTemplateDto permissionTemplate = templateDb.insertTemplate(db.getDefaultOrganization());
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    templateDb.addUserToTemplate(permissionTemplate, user1, "user");
    templateDb.addUserToTemplate(permissionTemplate, user1, "admin");
    templateDb.addUserToTemplate(permissionTemplate, user2, "user");

    underTest.deleteUserPermission(dbSession, permissionTemplate.getId(), user1.getId(), "user");

    assertThat(db.getDbClient().permissionTemplateDao().selectUserPermissionsByTemplateId(db.getSession(),
permissionTemplate.getId()))
      .extracting(PermissionTemplateUserDto::getUserId, PermissionTemplateUserDto::getPermission)
      .containsOnly(tuple(user1.getId(), "admin"), tuple(user2.getId(), "user"));
  }

  @Test
  public void should_add_group_permission_to_template() {
    PermissionTemplateDto permissionTemplate = templateDb.insertTemplate(db.getDefaultOrganization());
    GroupDto group = db.users().insertGroup();

    underTest.insertGroupPermission(dbSession, permissionTemplate.getId(), group.getId(), "user");
    dbSession.commit();

    assertThat(db.getDbClient().permissionTemplateDao().selectGroupPermissionsByTemplateId(db.getSession(),
permissionTemplate.getId()))
      .extracting(PermissionTemplateGroupDto::getTemplateId, PermissionTemplateGroupDto::getGroupId,
PermissionTemplateGroupDto::getPermission,
        PermissionTemplateGroupDto::getCreatedAt,
        PermissionTemplateGroupDto::getUpdatedAt)
      .containsOnly(tuple(permissionTemplate.getId(), group.getId(), "user", NOW, NOW));
  }

  @Test
  public void remove_by_group() {
    PermissionTemplateDto permissionTemplate = templateDb.insertTemplate(db.getDefaultOrganization());
    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    templateDb.addGroupToTemplate(permissionTemplate, group1, "user");
    templateDb.addGroupToTemplate(permissionTemplate, group1, "admin");
    templateDb.addGroupToTemplate(permissionTemplate, group2, "user");
```

```java
    underTest.deleteByGroup(db.getSession(), group1.getId());
    db.getSession().commit();

    assertThat(db.getDbClient().permissionTemplateDao().selectGroupPermissionsByTemplateId(db.getSession(),
permissionTemplate.getId()))
      .extracting(PermissionTemplateGroupDto::getGroupId, PermissionTemplateGroupDto::getPermission)
      .containsOnly(tuple(group2.getId(), "user"));
  }

  @Test
  public void should_add_group_permission_to_anyone() {
    PermissionTemplateDto permissionTemplate = templateDb.insertTemplate(db.getDefaultOrganization());

    underTest.insertGroupPermission(dbSession, permissionTemplate.getId(), null, "user");
    dbSession.commit();

    assertThat(db.getDbClient().permissionTemplateDao().selectGroupPermissionsByTemplateId(db.getSession(),
permissionTemplate.getId()))
      .extracting(PermissionTemplateGroupDto::getTemplateId, PermissionTemplateGroupDto::getGroupId,
PermissionTemplateGroupDto::getGroupName,
        PermissionTemplateGroupDto::getPermission)
      .containsOnly(tuple(permissionTemplate.getId(), 0, "Anyone", "user"));
  }

  @Test
  public void group_count_by_template_and_permission() {
    PermissionTemplateDto template1 = templateDb.insertTemplate();
    PermissionTemplateDto template2 = templateDb.insertTemplate();
    PermissionTemplateDto template3 = templateDb.insertTemplate();
    PermissionTemplateDto template4 = templateDb.insertTemplate();
    GroupDto group1 = db.users().insertGroup(newGroupDto());
    GroupDto group2 = db.users().insertGroup(newGroupDto());
    GroupDto group3 = db.users().insertGroup(newGroupDto());
    templateDb.addGroupToTemplate(template1.getId(), group1.getId(), CODEVIEWER);
    templateDb.addGroupToTemplate(template1.getId(), group2.getId(), CODEVIEWER);
    templateDb.addGroupToTemplate(template1.getId(), group3.getId(), CODEVIEWER);
    templateDb.addGroupToTemplate(template1.getId(), null, CODEVIEWER);
    templateDb.addGroupToTemplate(template1.getId(), group1.getId(), ADMIN);
    templateDb.addGroupToTemplate(template2.getId(), group1.getId(), ADMIN);
    templateDb.addGroupToTemplate(template4.getId(), group1.getId(), ISSUE_ADMIN);

    final List<CountByTemplateAndPermissionDto> result = new ArrayList<>();
    underTest.groupsCountByTemplateIdAndPermission(dbSession, asList(template1.getId(), template2.getId(),
template3.getId()),
      context -> result.add(context.getResultObject()));

    assertThat(result).extracting(CountByTemplateAndPermissionDto::getPermission,
CountByTemplateAndPermissionDto::getTemplateId, CountByTemplateAndPermissionDto::getCount)
```

```java
      .containsOnly(tuple(ADMIN, template1.getId(), 1), tuple(CODEVIEWER, template1.getId(), 4), tuple(ADMIN,
template2.getId(), 1));
 }

 @Test
 public void user_count_by_template_and_permission() {
  PermissionTemplateDto template1 = templateDb.insertTemplate();
  PermissionTemplateDto template2 = templateDb.insertTemplate();
  PermissionTemplateDto template3 = templateDb.insertTemplate();
  PermissionTemplateDto anotherTemplate = templateDb.insertTemplate();

  UserDto user1 = db.users().insertUser();
  UserDto user2 = db.users().insertUser();
  UserDto user3 = db.users().insertUser();

  templateDb.addUserToTemplate(template1.getId(), user1.getId(), ADMIN);
  templateDb.addUserToTemplate(template1.getId(), user2.getId(), ADMIN);
  templateDb.addUserToTemplate(template1.getId(), user3.getId(), ADMIN);
  templateDb.addUserToTemplate(template1.getId(), user1.getId(), USER);
  templateDb.addUserToTemplate(template2.getId(), user1.getId(), USER);
  templateDb.addUserToTemplate(anotherTemplate.getId(), user1.getId(), ISSUE_ADMIN);

  final List<CountByTemplateAndPermissionDto> result = new ArrayList<>();
  underTest.usersCountByTemplateIdAndPermission(dbSession, asList(template1.getId(), template2.getId(),
template3.getId()),
    context -> result.add(context.getResultObject()));
  assertThat(result)
    .extracting(CountByTemplateAndPermissionDto::getPermission,
CountByTemplateAndPermissionDto::getTemplateId, CountByTemplateAndPermissionDto::getCount)
    .containsExactlyInAnyOrder(
      tuple(ADMIN, template1.getId(), 3),
      tuple(USER, template1.getId(), 1),
      tuple(USER, template2.getId(), 1));
 }

 @Test
 public void selectPotentialPermissions_with_unknown_template_and_no_user() {
  List<String> result = underTest.selectPotentialPermissionsByUserIdAndTemplateId(dbSession, null, 42L);

  assertThat(result).isEmpty();
 }

 @Test
 public void selectPotentialPermissions_with_empty_template_and_new_user() {
  UserDto user = db.users().insertUser();
  PermissionTemplateDto template = templateDb.insertTemplate();

  List<String> result = underTest.selectPotentialPermissionsByUserIdAndTemplateId(dbSession, user.getId(),
```

```java
    template.getId());

    assertThat(result).isEmpty();
  }

  @Test
  public void selectPotentialPermission_with_template_users_groups_and_project_creator() {
    UserDto user = db.users().insertUser();
    GroupDto group = db.users().insertGroup(newGroupDto());
    db.users().insertMember(group, user);
    PermissionTemplateDto template = templateDb.insertTemplate();
    templateDb.addProjectCreatorToTemplate(template.getId(), SCAN_EXECUTION);
    templateDb.addProjectCreatorToTemplate(template.getId(), UserRole.ADMIN);
    templateDb.addUserToTemplate(template.getId(), user.getId(), UserRole.USER);
    templateDb.addUserToTemplate(template.getId(), user.getId(), UserRole.ADMIN);
    templateDb.addGroupToTemplate(template.getId(), group.getId(), UserRole.CODEVIEWER);
    templateDb.addGroupToTemplate(template.getId(), group.getId(), UserRole.ADMIN);
    templateDb.addGroupToTemplate(template.getId(), null, UserRole.ISSUE_ADMIN);

    List<String> resultWithUser = underTest.selectPotentialPermissionsByUserIdAndTemplateId(dbSession,
user.getId(), template.getId());
    List<String> resultWithoutUser = underTest.selectPotentialPermissionsByUserIdAndTemplateId(dbSession, null,
template.getId());

    assertThat(resultWithUser).containsOnlyOnce(SCAN_EXECUTION, UserRole.ADMIN, UserRole.USER,
UserRole.CODEVIEWER, UserRole.ISSUE_ADMIN);
    // only permission from anyone group
    assertThat(resultWithoutUser).containsOnly(UserRole.ISSUE_ADMIN);
  }

  @Test
  public void selectAllGroupPermissionTemplatesByGroupId() {
    PermissionTemplateDto permissionTemplate = templateDb.insertTemplate(db.getDefaultOrganization());
    GroupDto group1 = db.users().insertGroup();
    GroupDto group2 = db.users().insertGroup();
    templateDb.addGroupToTemplate(permissionTemplate, group1, "user");
    templateDb.addGroupToTemplate(permissionTemplate, group1, "admin");
    templateDb.addGroupToTemplate(permissionTemplate, group2, "user");

assertThat(db.getDbClient().permissionTemplateDao().selectAllGroupPermissionTemplatesByGroupId(db.getSessi
on(), group1.getId()))
      .extracting(PermissionTemplateGroupDto::getGroupId, PermissionTemplateGroupDto::getPermission)
      .containsOnly(tuple(group1.getId(), "user"), tuple(group1.getId(), "admin"));
  }

  @Test
  public void deleteByOrganization_does_not_fail_on_empty_db() {
    underTest.deleteByOrganization(dbSession, "some uuid");
```

```java
    dbSession.commit();
  }

  @Test
  public void deleteByOrganization_does_not_fail_when_organization_has_no_template() {
    OrganizationDto organization = db.organizations().insert();

    underTest.deleteByOrganization(dbSession, organization.getUuid());
    dbSession.commit();
  }

  @Test
  public void deleteByOrganization_delete_all_templates_of_organization_and_content_of_child_tables() {
    OrganizationDto organization1 = db.organizations().insert();
    OrganizationDto organization2 = db.organizations().insert();
    OrganizationDto organization3 = db.organizations().insert();

    PermissionTemplateDto[] templates = {
      createTemplate(organization1),
      createTemplate(organization2),
      createTemplate(organization3),
      createTemplate(organization1),
      createTemplate(organization2)
    };

    verifyTemplateIdsInDb(templates[0].getId(), templates[1].getId(), templates[2].getId(), templates[3].getId(),
templates[4].getId());

    underTest.deleteByOrganization(dbSession, organization2.getUuid());
    dbSession.commit();
    verifyTemplateIdsInDb(templates[0].getId(), templates[2].getId(), templates[3].getId());

    underTest.deleteByOrganization(dbSession, organization3.getUuid());
    dbSession.commit();
    verifyTemplateIdsInDb(templates[0].getId(), templates[3].getId());

    underTest.deleteByOrganization(dbSession, organization1.getUuid());
    dbSession.commit();
    verifyTemplateIdsInDb();
  }

  @Test
  public void delete_user_permissions_by_organization() {
    OrganizationDto organization = db.organizations().insert();
    OrganizationDto anotherOrganization = db.organizations().insert();
    UserDto user = db.users().insertUser();
    UserDto anotherUser = db.users().insertUser();
    PermissionTemplateDto template = db.permissionTemplates().insertTemplate(organization);
```

```java
    PermissionTemplateDto anotherTemplate = db.permissionTemplates().insertTemplate(anotherOrganization);
    String permission = "PERMISSION";
    db.permissionTemplates().addUserToTemplate(template.getId(), user.getId(), permission);
    db.permissionTemplates().addUserToTemplate(template.getId(), anotherUser.getId(), permission);
    db.permissionTemplates().addUserToTemplate(anotherTemplate.getId(), user.getId(), permission);

    underTest.deleteUserPermissionsByOrganization(dbSession, organization.getUuid(), user.getId());

    assertThat(underTest.selectUserPermissionsByTemplateId(dbSession,
template.getId())).extracting(PermissionTemplateUserDto::getUserId).containsOnly(anotherUser.getId());
    assertThat(underTest.selectUserPermissionsByTemplateId(dbSession,
anotherTemplate.getId())).extracting(PermissionTemplateUserDto::getUserId).containsOnly(user.getId());
  }

  @Test
  public void delete_user_permissions_by_user_id() {
    OrganizationDto organization = db.organizations().insert();
    OrganizationDto anotherOrganization = db.organizations().insert();
    UserDto user = db.users().insertUser();
    UserDto anotherUser = db.users().insertUser();
    PermissionTemplateDto template = db.permissionTemplates().insertTemplate(organization);
    PermissionTemplateDto anotherTemplate = db.permissionTemplates().insertTemplate(anotherOrganization);
    String permission = "PERMISSION";
    db.permissionTemplates().addUserToTemplate(template.getId(), user.getId(), permission);
    db.permissionTemplates().addUserToTemplate(template.getId(), anotherUser.getId(), permission);
    db.permissionTemplates().addUserToTemplate(anotherTemplate.getId(), user.getId(), permission);

    underTest.deleteUserPermissionsByUserId(dbSession, user.getId());
    db.commit();

    assertThat(db.select("select template_id as \"templateId\", user_id as \"userId\", permission_reference as
\"permission\" from perm_templates_users"))
      .extracting((row) -> row.get("templateId"), (row) -> row.get("userId"), (row) -> row.get("permission"))
      .containsOnly(tuple(template.getId(), anotherUser.getId().longValue(), permission));
  }

  private PermissionTemplateDto createTemplate(OrganizationDto organization) {
    UserDto user = db.users().insertUser();
    GroupDto group = db.users().insertGroup();
    db.users().insertMember(group, user);
    PermissionTemplateDto template = templateDb.insertTemplate(organization);
    templateDb.addProjectCreatorToTemplate(template.getId(), SCAN_EXECUTION);
    templateDb.addProjectCreatorToTemplate(template.getId(), UserRole.ADMIN);
    templateDb.addUserToTemplate(template.getId(), user.getId(), UserRole.USER);
    templateDb.addUserToTemplate(template.getId(), user.getId(), UserRole.ADMIN);
    templateDb.addGroupToTemplate(template.getId(), group.getId(), UserRole.CODEVIEWER);
    templateDb.addGroupToTemplate(template.getId(), group.getId(), UserRole.ADMIN);
    templateDb.addGroupToTemplate(template.getId(), null, UserRole.ISSUE_ADMIN);
```

```java
      return template;
    }

    private void verifyTemplateIdsInDb(Long... expectedTemplateIds) {
      assertThat(db.select("select distinct template_id as \"templateId\" from perm_templates_groups"))
        .extracting((row) -> (Long) row.get("templateId"))
        .containsOnly(expectedTemplateIds);
      assertThat(db.select("select distinct template_id as \"templateId\" from perm_templates_users"))
        .extracting((row) -> (Long) row.get("templateId"))
        .containsOnly(expectedTemplateIds);
      assertThat(db.select("select distinct template_id as \"templateId\" from perm_tpl_characteristics"))
        .extracting((row) -> (Long) row.get("templateId"))
        .containsOnly(expectedTemplateIds);
      assertThat(db.select("select distinct id as \"templateId\" from permission_templates"))
        .extracting((row) -> (Long) row.get("templateId"))
        .containsOnly(expectedTemplateIds);
    }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import com.google.common.base.Strings;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;

public class PermissionTemplateCharacteristicDtoTest {
  @Rule
  public ExpectedException expectedException = ExpectedException.none();
```

```java
    PermissionTemplateCharacteristicDto underTest = new PermissionTemplateCharacteristicDto();

    @Test
    public void check_permission_field_length() {
      expectedException.expect(IllegalArgumentException.class);
      expectedException
        .expectMessage("Permission key length (65) is longer than the maximum authorized (64).
'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa' was provided.");

      underTest.setPermission(Strings.repeat("a", 65));
    }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.Date;
import org.apache.commons.lang.math.RandomUtils;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.core.util.Uuids;

import static org.apache.commons.lang.RandomStringUtils.randomAlphanumeric;
import static org.apache.commons.lang.RandomStringUtils.randomAscii;

public class PermissionTemplateTesting {
  public static PermissionTemplateDto newPermissionTemplateDto() {
    return new PermissionTemplateDto()
      .setName(randomAlphanumeric(60))
      .setDescription(randomAscii(500))
      .setOrganizationUuid(randomAlphanumeric(40))
```

```java
      .setUuid(Uuids.create())
      .setCreatedAt(new Date())
      .setUpdatedAt(new Date());
  }

  public static PermissionTemplateUserDto newPermissionTemplateUserDto() {
    return new PermissionTemplateUserDto()
      .setPermission(ProjectPermissions.ALL.get(RandomUtils.nextInt(ProjectPermissions.ALL.size())))
      .setCreatedAt(new Date())
      .setUpdatedAt(new Date());
  }

  public static PermissionTemplateGroupDto newPermissionTemplateGroupDto() {
    return new PermissionTemplateGroupDto()
      .setPermission(ProjectPermissions.ALL.get(RandomUtils.nextInt(ProjectPermissions.ALL.size())))
      .setCreatedAt(new Date())
      .setUpdatedAt(new Date());
  }

  public static PermissionTemplateCharacteristicDto newPermissionTemplateCharacteristicDto() {
    return new PermissionTemplateCharacteristicDto()
      .setPermission(ProjectPermissions.ALL.get(RandomUtils.nextInt(ProjectPermissions.ALL.size())))
      .setWithProjectCreator(RandomUtils.nextBoolean())
      .setCreatedAt(System.currentTimeMillis())
      .setUpdatedAt(System.currentTimeMillis());
  }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;
```

```java
import java.util.Collections;
import java.util.List;
import java.util.stream.IntStream;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.permission.PermissionQuery;
import org.sonar.db.user.GroupDto;

import static java.util.Arrays.asList;
import static org.assertj.core.api.Assertions.assertThat;
import static org.assertj.core.api.Assertions.tuple;
import static org.sonar.api.web.UserRole.ADMIN;
import static org.sonar.api.web.UserRole.USER;
import static org.sonar.core.permission.GlobalPermissions.PROVISIONING;
import static org.sonar.db.permission.PermissionQuery.builder;
import static org.sonar.db.user.GroupTesting.newGroupDto;

public class GroupWithPermissionTemplateDaoTest {

  @Rule
  public DbTester db = DbTester.create(System2.INSTANCE);

  private DbSession session = db.getSession();
  private PermissionTemplateDbTester permissionTemplateDbTester = db.permissionTemplates();
  private PermissionTemplateDao underTest = db.getDbClient().permissionTemplateDao();

  @Test
  public void select_group_names_by_query_and_template() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(organization, "Group-1");
    GroupDto group2 = db.users().insertGroup(organization, "Group-2");
    GroupDto group3 = db.users().insertGroup(organization, "Group-3");

    PermissionTemplateDto template = permissionTemplateDbTester.insertTemplate(organization);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), USER);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), ADMIN);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group2.getId(), PROVISIONING);

    PermissionTemplateDto anotherTemplate = permissionTemplateDbTester.insertTemplate(organization);
    permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), null, USER);
    permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), group1.getId(), PROVISIONING);

    assertThat(selectGroupNamesByQueryAndTemplate(builder(), organization, template))
```

```
    .containsOnly("Group-1", "Group-2", "Group-3", "Anyone");
  assertThat(selectGroupNamesByQueryAndTemplate(builder().withAtLeastOnePermission(), organization,
template))
    .containsOnly("Group-1", "Group-2");
  assertThat(selectGroupNamesByQueryAndTemplate(builder().setPermission(USER), organization, template))
    .containsOnly("Group-1");
  assertThat(selectGroupNamesByQueryAndTemplate(builder().setPermission(USER), organization,
anotherTemplate))
    .containsOnly("Anyone");
  assertThat(selectGroupNamesByQueryAndTemplate(builder().setSearchQuery("groU"), organization, template))
    .containsOnly("Group-1", "Group-2", "Group-3");
  assertThat(selectGroupNamesByQueryAndTemplate(builder().setSearchQuery("nYo"), organization, template))
    .containsOnly("Anyone");
  assertThat(selectGroupNamesByQueryAndTemplate(builder().setSearchQuery("p-2"), organization, template))
    .containsOnly("Group-2");

  assertThat(selectGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).withAt
LeastOnePermission().build(), organization, 123L))
    .isEmpty();
  assertThat(selectGroupNamesByQueryAndTemplate(builder().setSearchQuery("unknown"), organization,
template))
    .isEmpty();
 }

 @Test
 public void select_group_names_by_query_and_template_is_ordered_by_group_names() {
  OrganizationDto organization = db.organizations().insert();
  GroupDto group2 = db.users().insertGroup(organization, "Group-2");
  db.users().insertGroup(organization, "Group-3");
  db.users().insertGroup(organization, "Group-1");

  PermissionTemplateDto template = permissionTemplateDbTester.insertTemplate(organization);
  permissionTemplateDbTester.addGroupToTemplate(template.getId(), group2.getId(), USER);

  assertThat(selectGroupNamesByQueryAndTemplate(builder(), organization, template))
    .containsExactly("Anyone", "Group-1", "Group-2", "Group-3");
 }

 @Test
 public void select_group_names_by_query_and_template_is_paginated() {
  OrganizationDto organization = db.organizations().insert();
  IntStream.rangeClosed(0, 9).forEach(i -> db.users().insertGroup(organization, i + "-name"));

  PermissionTemplateDto template = permissionTemplateDbTester.insertTemplate(organization);

  assertThat(selectGroupNamesByQueryAndTemplate(builder().setPageIndex(1).setPageSize(1), organization,
template))
    .containsExactly("0-name");
```

```java
    assertThat(selectGroupNamesByQueryAndTemplate(builder().setPageIndex(2).setPageSize(3), organization,
template))
      .containsExactly("3-name", "4-name", "5-name");
  }

  @Test
  public void select_group_names_by_query_and_template_returns_anyone() {
    OrganizationDto organization = db.organizations().insert();
    PermissionTemplateDto template = permissionTemplateDbTester.insertTemplate(organization);

    GroupDto group = db.users().insertGroup(newGroupDto().setName("Group"));
    PermissionTemplateDto otherTemplate = permissionTemplateDbTester.insertTemplate(organization);
    permissionTemplateDbTester.addGroupToTemplate(otherTemplate.getId(), group.getId(), USER);

    assertThat(selectGroupNamesByQueryAndTemplate(builder().setSearchQuery("nyo"), organization, template))
      .containsExactly("Anyone");
  }

  @Test
  public void count_group_names_by_query_and_template() {
    OrganizationDto organization = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(organization, "Group-1");
    GroupDto group2 = db.users().insertGroup(organization, "Group-2");
    GroupDto group3 = db.users().insertGroup(organization, "Group-3");

    PermissionTemplateDto template = permissionTemplateDbTester.insertTemplate(organization);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), USER);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), ADMIN);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group2.getId(), PROVISIONING);

    PermissionTemplateDto anotherTemplate = permissionTemplateDbTester.insertTemplate(organization);
    permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), null, USER);
    permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), group1.getId(), PROVISIONING);

    assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()),
organization, template))
      .isEqualTo(4);
assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).withAt
LeastOnePermission(), organization, template))
      .isEqualTo(2);
assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).setPer
mission(USER), organization, template)).isEqualTo(1);
assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).setPer
mission(USER), organization, anotherTemplate))
      .isEqualTo(1);
assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).setSear
chQuery("groU"), organization, template))
      .isEqualTo(3);
```

```
      assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).setSear
chQuery("nYo"), organization, template))
        .isEqualTo(1);
      assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).setSear
chQuery("p-2"), organization, template))
        .isEqualTo(1);

      assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).withAt
LeastOnePermission().build(), organization, 123L))
        .isZero();
      assertThat(countGroupNamesByQueryAndTemplate(builder().setOrganizationUuid(organization.getUuid()).setSear
chQuery("unknown"), organization, template))
        .isZero();
  }

  @Test
  public void select_group_permissions_by_template_id_and_group_names() {
    GroupDto group1 = db.users().insertGroup(newGroupDto().setName("Group-1"));
    GroupDto group2 = db.users().insertGroup(newGroupDto().setName("Group-2"));
    GroupDto group3 = db.users().insertGroup(newGroupDto().setName("Group-3"));

    PermissionTemplateDto template = permissionTemplateDbTester.insertTemplate();
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), USER);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), ADMIN);
    permissionTemplateDbTester.addGroupToTemplate(template.getId(), group2.getId(), PROVISIONING);

    PermissionTemplateDto anotherTemplate = permissionTemplateDbTester.insertTemplate();
    permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), null, USER);
    permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), group1.getId(), PROVISIONING);

    assertThat(underTest.selectGroupPermissionsByTemplateIdAndGroupNames(session, template.getId(),
asList("Group-1")))
      .extracting(PermissionTemplateGroupDto::getGroupId, PermissionTemplateGroupDto::getGroupName,
PermissionTemplateGroupDto::getPermission)
      .containsOnly(
        tuple(group1.getId(), "Group-1", USER),
        tuple(group1.getId(), "Group-1", ADMIN));

    assertThat(underTest.selectGroupPermissionsByTemplateIdAndGroupNames(session, anotherTemplate.getId(),
asList("Group-1")))
      .extracting(PermissionTemplateGroupDto::getGroupId, PermissionTemplateGroupDto::getGroupName,
PermissionTemplateGroupDto::getPermission)
      .containsOnly(
        tuple(group1.getId(), "Group-1", PROVISIONING));

    assertThat(underTest.selectGroupPermissionsByTemplateIdAndGroupNames(session, anotherTemplate.getId(),
asList("Anyone")))
      .extracting(PermissionTemplateGroupDto::getGroupId, PermissionTemplateGroupDto::getGroupName,
```

```
PermissionTemplateGroupDto::getPermission)
    .containsOnly(
      tuple(0, "Anyone", USER));

  assertThat(underTest.selectGroupPermissionsByTemplateIdAndGroupNames(session, template.getId(),
asList("Group-1", "Group-2", "Anyone"))).hasSize(3);
  assertThat(underTest.selectGroupPermissionsByTemplateIdAndGroupNames(session, template.getId(),
asList("Unknown"))).isEmpty();
  assertThat(underTest.selectGroupPermissionsByTemplateIdAndGroupNames(session, template.getId(),
Collections.emptyList())).isEmpty();
}

@Test
public void select_group_permissions_by_template_id() {
  GroupDto group1 = db.users().insertGroup(newGroupDto().setName("Group-1"));
  GroupDto group2 = db.users().insertGroup(newGroupDto().setName("Group-2"));
  GroupDto group3 = db.users().insertGroup(newGroupDto().setName("Group-3"));

  PermissionTemplateDto template = permissionTemplateDbTester.insertTemplate();
  permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), USER);
  permissionTemplateDbTester.addGroupToTemplate(template.getId(), group1.getId(), ADMIN);
  permissionTemplateDbTester.addGroupToTemplate(template.getId(), group2.getId(), PROVISIONING);

  PermissionTemplateDto anotherTemplate = permissionTemplateDbTester.insertTemplate();
  permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), null, USER);
  permissionTemplateDbTester.addGroupToTemplate(anotherTemplate.getId(), group1.getId(), PROVISIONING);

  assertThat(underTest.selectGroupPermissionsByTemplateId(session, template.getId()))
    .extracting(PermissionTemplateGroupDto::getGroupId, PermissionTemplateGroupDto::getGroupName,
PermissionTemplateGroupDto::getPermission)
    .containsOnly(
      tuple(group1.getId(), "Group-1", USER),
      tuple(group1.getId(), "Group-1", ADMIN),
      tuple(group2.getId(), "Group-2", PROVISIONING));
  assertThat(underTest.selectGroupPermissionsByTemplateId(session, anotherTemplate.getId()))
    .extracting(PermissionTemplateGroupDto::getGroupId, PermissionTemplateGroupDto::getGroupName,
PermissionTemplateGroupDto::getPermission)
    .containsOnly(
      tuple(group1.getId(), "Group-1", PROVISIONING),
      tuple(0, "Anyone", USER));

  assertThat(underTest.selectGroupPermissionsByTemplateId(session, 321L)).isEmpty();
}

private List<String> selectGroupNamesByQueryAndTemplate(PermissionQuery.Builder queryBuilder,
OrganizationDto organization, PermissionTemplateDto permissionTemplateDto) {
  return
selectGroupNamesByQueryAndTemplate(queryBuilder.setOrganizationUuid(organization.getUuid()).build(),
```

```java
    organization, permissionTemplateDto.getId());
  }

  private List<String> selectGroupNamesByQueryAndTemplate(PermissionQuery query, OrganizationDto
organization, long templateId) {
    return underTest.selectGroupNamesByQueryAndTemplate(session, query, templateId);
  }

  private int countGroupNamesByQueryAndTemplate(PermissionQuery.Builder queryBuilder, OrganizationDto
organization, PermissionTemplateDto permissionTemplateDto) {
    return countGroupNamesByQueryAndTemplate(queryBuilder.build(), organization,
permissionTemplateDto.getId());
  }

  private int countGroupNamesByQueryAndTemplate(PermissionQuery query, OrganizationDto organization, long
templateId) {
    return underTest.countGroupNamesByQueryAndTemplate(session, query, organization.getUuid(), templateId);
  }

}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission.template;

import java.util.List;
import java.util.Optional;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
```

```java
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;

import static com.google.common.collect.Lists.newArrayList;
import static com.google.common.primitives.Longs.asList;
import static java.util.Collections.emptyList;
import static org.assertj.core.api.Assertions.assertThat;

public class PermissionTemplateCharacteristicDaoTest {
 @Rule
 public ExpectedException expectedException = ExpectedException.none();
 @Rule
 public DbTester db = DbTester.create(System2.INSTANCE);
 private DbSession dbSession = db.getSession();
 private PermissionTemplateCharacteristicDao underTest = new PermissionTemplateCharacteristicDao();

 @Test
 public void selectByTemplateId_filter_by_template_id() {
   PermissionTemplateCharacteristicDto templatePermission1 = underTest.insert(dbSession, new
PermissionTemplateCharacteristicDto()
     .setPermission(UserRole.ADMIN)
     .setTemplateId(1L)
     .setWithProjectCreator(true)
     .setCreatedAt(1_000_000_000L)
     .setUpdatedAt(2_000_000_000L));
   PermissionTemplateCharacteristicDto templatePermission2 = underTest.insert(dbSession, new
PermissionTemplateCharacteristicDto()
     .setPermission(UserRole.USER)
     .setTemplateId(2L)
     .setWithProjectCreator(false)
     .setCreatedAt(1_000_000_000L)
     .setUpdatedAt(2_000_000_000L));
   PermissionTemplateCharacteristicDto templatePermissionForAnotherTemplate = underTest.insert(dbSession, new
PermissionTemplateCharacteristicDto()
     .setPermission(UserRole.ADMIN)
     .setTemplateId(42L)
     .setWithProjectCreator(true)
     .setCreatedAt(1_000_000_000L)
     .setUpdatedAt(2_000_000_000L));

   List<PermissionTemplateCharacteristicDto> result = underTest.selectByTemplateIds(dbSession,
newArrayList(1L, 2L));
   assertThat(result)
     .hasSize(2)
     .extracting("id")
     .doesNotContain(templatePermissionForAnotherTemplate.getId())
     .containsOnly(templatePermission1.getId(), templatePermission2.getId());
   assertThat(result.get(0))
```

```java
      .isEqualToComparingFieldByField(templatePermission1);
  }

  @Test
  public void selectByTemplateId_for_empty_list_of_template_id() {
    List<PermissionTemplateCharacteristicDto> result = underTest.selectByTemplateIds(dbSession, emptyList());

    assertThat(result).isEmpty();
  }

  @Test
  public void selectByPermissionAndTemplateId() {
    PermissionTemplateCharacteristicDto templatePermission1 = underTest.insert(dbSession, new
PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.ADMIN)
      .setTemplateId(1L)
      .setWithProjectCreator(true)
      .setCreatedAt(1_000_000_000L)
      .setUpdatedAt(2_000_000_000L));
    underTest.insert(dbSession, new PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(1L)
      .setWithProjectCreator(false)
      .setCreatedAt(1_000_000_000L)
      .setUpdatedAt(2_000_000_000L));
    underTest.insert(dbSession, new PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.ADMIN)
      .setTemplateId(42L)
      .setWithProjectCreator(true)
      .setCreatedAt(1_000_000_000L)
      .setUpdatedAt(2_000_000_000L));

    Optional<PermissionTemplateCharacteristicDto> result =
underTest.selectByPermissionAndTemplateId(dbSession, UserRole.ADMIN, 1L);

    assertThat(result).isPresent();
    assertThat(result.get()).isEqualToComparingFieldByField(templatePermission1);
  }

  @Test
  public void insert() {
    PermissionTemplateCharacteristicDto expectedResult = underTest.insert(dbSession, new
PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(1L)
      .setWithProjectCreator(true)
      .setCreatedAt(123_456_789L)
      .setUpdatedAt(2_000_000_000L));
```

```java
    PermissionTemplateCharacteristicDto result =
dbSession.getMapper(PermissionTemplateCharacteristicMapper.class).selectById(expectedResult.getId());
    assertThat(result.getId()).isNotNull();
    assertThat(result).isEqualToComparingFieldByField(expectedResult);
  }

  @Test
  public void update_only_change_with_project_creator_and_updated_at() {
    PermissionTemplateCharacteristicDto insertedDto = underTest.insert(dbSession, new
PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(1L)
      .setWithProjectCreator(true)
      .setCreatedAt(123_456_789L)
      .setUpdatedAt(2_000_000_000L));

    underTest.update(dbSession, new PermissionTemplateCharacteristicDto()
      .setId(insertedDto.getId())
      .setPermission("PERMISSION_ARE_NOT_UPDATABLE")
      .setTemplateId(42L)
      .setCreatedAt(42L)
      .setWithProjectCreator(false)
      .setUpdatedAt(3_000_000_000L));

    PermissionTemplateCharacteristicDto result = underTest.selectByPermissionAndTemplateId(dbSession,
insertedDto.getPermission(), insertedDto.getTemplateId()).get();
    assertThat(result).extracting("id", "permission", "templateId", "createdAt")
      .containsExactly(insertedDto.getId(), insertedDto.getPermission(), insertedDto.getTemplateId(),
insertedDto.getCreatedAt());
    assertThat(result).extracting("withProjectCreator", "updatedAt")
      .containsExactly(false, 3_000_000_000L);
  }

  @Test
  public void fail_insert_if_created_at_is_equal_to_0() {
    expectedException.expect(IllegalArgumentException.class);

    underTest.insert(dbSession, new PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(1L)
      .setWithProjectCreator(true)
      .setUpdatedAt(2_000_000_000L));
  }

  @Test
  public void fail_insert_if_updated_at_is_equal_to_0() {
    expectedException.expect(IllegalArgumentException.class);
```

```java
    underTest.insert(dbSession, new PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(1L)
      .setWithProjectCreator(true)
      .setCreatedAt(2_000_000_000L));
  }

  @Test
  public void fail_update_if_id_is_null() {
    expectedException.expect(NullPointerException.class);

    underTest.update(dbSession, new PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(1L)
      .setWithProjectCreator(true)
      .setCreatedAt(123_456_789L)
      .setUpdatedAt(2_000_000_000L));
  }

  @Test
  public void delete_by_permission_template_id() {
    underTest.insert(dbSession, new PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(1L)
      .setWithProjectCreator(true)
      .setCreatedAt(123_456_789L)
      .setUpdatedAt(2_000_000_000L));
    underTest.insert(dbSession, new PermissionTemplateCharacteristicDto()
      .setPermission(UserRole.USER)
      .setTemplateId(2L)
      .setWithProjectCreator(true)
      .setCreatedAt(123_456_789L)
      .setUpdatedAt(2_000_000_000L));

    assertThat(underTest.selectByTemplateIds(dbSession, asList(1L))).hasSize(1);
    assertThat(underTest.selectByTemplateIds(dbSession, asList(1L, 2L))).hasSize(2);

    dbSession.getMapper(PermissionTemplateCharacteristicMapper.class).deleteByTemplateId(1L);

    assertThat(underTest.selectByTemplateIds(dbSession, asList(1L))).hasSize(0);
    assertThat(underTest.selectByTemplateIds(dbSession, asList(1L, 2L))).hasSize(1);
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
```

```
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.ExpectedException;

import static org.assertj.core.api.Assertions.assertThat;

public class PermissionQueryTest {

 @Rule
 public ExpectedException expectedException = ExpectedException.none();

 @Test
 public void create_query() {
  PermissionQuery quey = PermissionQuery.builder()
    .setComponentUuid("COMPONENT_UUID")
    .setOrganizationUuid("ORGANIZATION_UUID")
    .setPermission("user")
    .setSearchQuery("sonar")
    .build();

  assertThat(quey.getComponentUuid()).isEqualTo("COMPONENT_UUID");
  assertThat(quey.getOrganizationUuid()).isEqualTo("ORGANIZATION_UUID");
  assertThat(quey.getPermission()).isEqualTo("user");
  assertThat(quey.getSearchQuery()).isEqualTo("sonar");
 }

 @Test
 public void create_query_with_pagination() {
  PermissionQuery quey = PermissionQuery.builder()
    .setOrganizationUuid("ORGANIZATION_UUID")
    .setPageSize(10)
```

```java
      .setPageIndex(5)
      .build();

    assertThat(quey.getPageOffset()).isEqualTo(40);
    assertThat(quey.getPageSize()).isEqualTo(10);
  }

  @Test
  public void create_query_with_default_pagination() {
    PermissionQuery quey = PermissionQuery.builder()
      .setOrganizationUuid("ORGANIZATION_UUID")
      .build();

    assertThat(quey.getPageOffset()).isEqualTo(0);
    assertThat(quey.getPageSize()).isEqualTo(20);
  }

  @Test
  public void fail_when_no_organization() {
    expectedException.expect(NullPointerException.class);
    expectedException.expectMessage("Organization UUID cannot be null");

    PermissionQuery.builder().setOrganizationUuid(null).build();
  }

  @Test
  public void fail_when_search_query_length_is_less_than_3_characters() {
    expectedException.expect(IllegalArgumentException.class);
    expectedException.expectMessage("Search query should contains at least 3 characters");

    PermissionQuery.builder()
      .setOrganizationUuid("ORGANIZATION_UUID")
      .setSearchQuery("so")
      .build();
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
```

```java
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.db.permission;

import java.util.Collection;
import java.util.Collections;
import java.util.List;
import java.util.Random;
import java.util.Set;
import java.util.stream.Collectors;
import java.util.stream.IntStream;
import org.junit.Before;
import org.junit.Rule;
import org.junit.Test;
import org.sonar.api.utils.System2;
import org.sonar.api.web.UserRole;
import org.sonar.core.permission.ProjectPermissions;
import org.sonar.core.util.stream.MoreCollectors;
import org.sonar.db.DbSession;
import org.sonar.db.DbTester;
import org.sonar.db.component.BranchType;
import org.sonar.db.component.ComponentDto;
import org.sonar.db.organization.OrganizationDto;
import org.sonar.db.user.GroupDto;
import org.sonar.db.user.UserDto;

import static com.google.common.collect.Sets.newHashSet;
import static java.util.Collections.singleton;
import static org.assertj.core.api.Assertions.assertThat;
import static org.sonar.core.permission.GlobalPermissions.QUALITY_GATE_ADMIN;
import static org.sonar.core.permission.GlobalPermissions.SCAN_EXECUTION;
import static org.sonar.core.permission.GlobalPermissions.SYSTEM_ADMIN;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_GATES;
import static org.sonar.db.permission.OrganizationPermission.ADMINISTER_QUALITY_PROFILES;
import static org.sonar.db.permission.OrganizationPermission.SCAN;

public class AuthorizationDaoTest {

  private static final Long PROJECT_ID = 300L;
  private static final int MISSING_ID = -1;
  private static final String A_PERMISSION = "a-permission";
  private static final String DOES_NOT_EXIST = "does-not-exist";
```

```java
@Rule
public DbTester db = DbTester.create(System2.INSTANCE);

private final Random random = new Random();
private DbSession dbSession = db.getSession();
private AuthorizationDao underTest = new AuthorizationDao();
private OrganizationDto organization;
private UserDto user;
private GroupDto group1;
private GroupDto group2;
private Set<Long> randomPublicProjectIds;
private Set<Long> randomPrivateProjectIds;
private Set<Integer> randomExistingUserIds;
private String randomPermission = "p" + random.nextInt();

@Before
public void setUp() throws Exception {
  organization = db.organizations().insert();
  user = db.users().insertUser();
  group1 = db.users().insertGroup(organization, "group1");
  group2 = db.users().insertGroup(organization, "group2");
  randomExistingUserIds = IntStream.range(0, 1 + Math.abs(random.nextInt(5)))
    .map(i -> db.users().insertUser().getId())
    .boxed()
    .collect(MoreCollectors.toSet());
  randomPublicProjectIds = IntStream.range(0, 1 + Math.abs(random.nextInt(5)))
    .mapToLong(i -> db.components().insertPublicProject(organization).getId())
    .boxed()
    .collect(MoreCollectors.toSet());
  randomPrivateProjectIds = IntStream.range(0, 1 + Math.abs(random.nextInt(5)))
    .mapToLong(i -> db.components().insertPrivateProject(organization).getId())
    .boxed()
    .collect(MoreCollectors.toSet());
}

/**
 * Union of the permissions granted to:
 * - the user
 * - the groups which user is member
 * - anyone
 */
@Test
public void selectOrganizationPermissions_for_logged_in_user() {
  ComponentDto project = db.components().insertPrivateProject(organization);
  db.users().insertMember(group1, user);
  db.users().insertPermissionOnUser(organization, user, "perm1");
  db.users().insertProjectPermissionOnUser(user, "perm42", project);
```

```
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertPermissionOnAnyone(organization, "perm3");

    // ignored permissions, user is not member of this group
    db.users().insertPermissionOnGroup(group2, "ignored");

    Set<String> permissions = underTest.selectOrganizationPermissions(dbSession, organization.getUuid(),
  user.getId());

    assertThat(permissions).containsOnly("perm1", "perm2", "perm3");
  }

  /**
   * Anonymous user only benefits from the permissions granted to
   * "Anyone"
   */
  @Test
  public void selectOrganizationPermissions_for_anonymous_user() {
    db.users().insertPermissionOnAnyone(organization, "perm1");

    // ignored permissions
    db.users().insertPermissionOnUser(organization, user, "ignored");
    db.users().insertPermissionOnGroup(group1, "ignored");

    Set<String> permissions = underTest.selectOrganizationPermissionsOfAnonymous(dbSession,
  organization.getUuid());

    assertThat(permissions).containsOnly("perm1");
  }

  @Test
  public void countUsersWithGlobalPermissionExcludingGroup() {
    // users with global permission "perm1" :
    // - "u1" and "u2" through group "g1"
    // - "u1" and "u3" through group "g2"
    // - "u4"

    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    UserDto user3 = db.users().insertUser();
    UserDto user4 = db.users().insertUser();
    UserDto user5 = db.users().insertUser();

    OrganizationDto organization = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(organization, "g1");
    db.users().insertPermissionOnGroup(group1, "perm1");
    db.users().insertPermissionOnGroup(group1, "perm2");
    db.users().insertMember(group1, user1);
```

```java
    db.users().insertMember(group1, user2);

    GroupDto group2 = db.users().insertGroup(organization, "g2");
    db.users().insertPermissionOnGroup(group2, "perm1");
    db.users().insertPermissionOnGroup(group2, "perm2");
    db.users().insertMember(group2, user1);
    db.users().insertMember(group2, user3);

    // group3 has the permission "perm1" but has no users
    GroupDto group3 = db.users().insertGroup(organization, "g2");
    db.users().insertPermissionOnGroup(group3, "perm1");

    db.users().insertPermissionOnUser(organization, user4, "perm1");
    db.users().insertPermissionOnUser(organization, user4, "perm2");
    db.users().insertPermissionOnAnyone(organization, "perm1");

    // other organizations are ignored
    OrganizationDto org2 = db.organizations().insert();
    db.users().insertPermissionOnUser(org2, user1, "perm1");

    // excluding group "g1" -> remain u1, u3 and u4
    assertThat(underTest.countUsersWithGlobalPermissionExcludingGroup(db.getSession(),
      organization.getUuid(), "perm1", group1.getId())).isEqualTo(3);

    // excluding group "g2" -> remain u1, u2 and u4
    assertThat(underTest.countUsersWithGlobalPermissionExcludingGroup(db.getSession(),
      organization.getUuid(), "perm1", group2.getId())).isEqualTo(3);

    // excluding group "g3" -> remain u1, u2, u3 and u4
    assertThat(underTest.countUsersWithGlobalPermissionExcludingGroup(db.getSession(),
      organization.getUuid(), "perm1", group3.getId())).isEqualTo(4);

    // nobody has the permission
    assertThat(underTest.countUsersWithGlobalPermissionExcludingGroup(db.getSession(),
      organization.getUuid(), "missingPermission", group1.getId())).isEqualTo(0);
  }

  @Test
  public void countUsersWithGlobalPermissionExcludingUser() {
    // group g1 has the permission p1 and has members user1 and user2
    // user3 has the permission
    UserDto user1 = db.users().insertUser();
    UserDto user2 = db.users().insertUser();
    UserDto user3 = db.users().insertUser();

    OrganizationDto organization = db.organizations().insert();
    GroupDto group1 = db.users().insertGroup(organization, "g1");
    db.users().insertPermissionOnGroup(group1, "p1");
```

```java
    db.users().insertPermissionOnGroup(group1, "p2");
    db.users().insertMember(group1, user1);
    db.users().insertMember(group1, user2);
    db.users().insertPermissionOnUser(organization, user3, "p1");
    db.users().insertPermissionOnAnyone(organization, "p1");

    // other organizations are ignored
    OrganizationDto org2 = db.organizations().insert();
    db.users().insertPermissionOnUser(org2, user1, "p1");

    // excluding user1 -> remain user2 and user3
    assertThat(underTest.countUsersWithGlobalPermissionExcludingUser(db.getSession(),
      organization.getUuid(), "p1", user1.getId())).isEqualTo(2);

    // excluding user3 -> remain the members of group g1
    assertThat(underTest.countUsersWithGlobalPermissionExcludingUser(db.getSession(),
      organization.getUuid(), "p1", user3.getId())).isEqualTo(2);

    // excluding unknown user
    assertThat(underTest.countUsersWithGlobalPermissionExcludingUser(db.getSession(),
      organization.getUuid(), "p1", -1)).isEqualTo(3);

    // nobody has the permission
    assertThat(underTest.countUsersWithGlobalPermissionExcludingUser(db.getSession(),
      organization.getUuid(), "missingPermission", group1.getId())).isEqualTo(0);
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_empty_for_group_AnyOne_if_project_set_is_empty_on_public_project() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, Collections.emptySet(), null, UserRole.USER))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectIds_returns_empty_for_user_if_project_set_is_empty_on_public_project() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, Collections.emptySet(), user.getId(),
UserRole.USER))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectIds_returns_empty_for_group_AnyOne_for_non_existent_projects() {
    Set<Long> randomNonProjectsSet = IntStream.range(0, 1 + Math.abs(random.nextInt(5)))
      .mapToLong(i -> 3_562 + i)
      .boxed()
      .collect(MoreCollectors.toSet());
```

```java
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomNonProjectsSet, null, UserRole.USER))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectIds_returns_empty_for_user_for_non_existent_projects() {
    Set<Long> randomNonProjectsSet = IntStream.range(0, 1 + Math.abs(random.nextInt(5)))
      .mapToLong(i -> 9_666 + i)
      .boxed()
      .collect(MoreCollectors.toSet());

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomNonProjectsSet, user.getId(),
UserRole.USER))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_any_public_project_for_group_AnyOne_without_any_permission_in_DB_and_
permission_USER() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPublicProjectIds, null, UserRole.USER))
      .containsAll(randomPublicProjectIds);
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_any_public_project_for_user_without_any_permission_in_DB_and_permission_
USER() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPublicProjectIds, user.getId(),
UserRole.USER))
      .containsAll(randomPublicProjectIds);
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_any_public_project_for_group_AnyOne_without_any_permission_in_DB_and_
permission_CODEVIEWER() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPublicProjectIds, null,
UserRole.CODEVIEWER))
      .containsAll(randomPublicProjectIds);
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_any_public_project_for_user_without_any_permission_in_DB_and_permission_
CODEVIEWER() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPublicProjectIds, user.getId(),
UserRole.CODEVIEWER))
```

```
      .containsAll(randomPublicProjectIds);
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_empty_for_other_permission_for_group_AnyOne_on_public_project_without_a
ny_permission_in_DB() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPublicProjectIds, null, randomPermission))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_empty_for_any_permission_for_user_on_public_project_without_any_permissio
n_in_DB() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPublicProjectIds, user.getId(),
randomPermission))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectIds_returns_public_project_if_user_is_granted_project_permission_directly() {
    ComponentDto project = db.components().insertPublicProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user, randomPermission, project);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), otherUser.getId(),
randomPermission))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(otherProject.getId()), user.getId(),
randomPermission))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(),
randomPermission))
      .containsOnly(project.getId());
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(), "another
perm"))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectIds_returns_public_project_if_user_is_granted_project_permission_by_group()
{
    ComponentDto project = db.components().insertPublicProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertMember(group1, user);
```

```java
    db.users().insertProjectPermissionOnGroup(group1, randomPermission, project);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(),
randomPermission))
      .containsOnly(project.getId());
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(otherProject.getId()), user.getId(),
randomPermission))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), otherUser.getId(),
randomPermission))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(), "another
perm"))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_public_project_if_group_AnyOne_is_granted_project_permission_directly() {
    ComponentDto project = db.components().insertPublicProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    db.users().insertProjectPermissionOnAnyone(randomPermission, project);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), null, randomPermission))
      .containsOnly(project.getId());
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), null, "another perm"))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(otherProject.getId()), null,
randomPermission))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_empty_for_user_on_private_project_without_any_permission_in_DB_and_perm
ission_USER() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, user.getId(),
UserRole.USER))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedProjectIds_returns_empty_for_group_AnyOne_on_private_project_without_any_permission_in_DB
_and_permission_USER() {
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, null, UserRole.USER))
      .isEmpty();
  }
```

```java
@Test
public void
keepAuthorizedProjectIds_returns_empty_for_user_on_private_project_without_any_permission_in_DB_and_perm
ission_CODEVIEWER() {
  assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, user.getId(),
UserRole.CODEVIEWER))
    .isEmpty();
}

@Test
public void
keepAuthorizedProjectIds_returns_empty_for_group_AnyOne_on_private_project_without_any_permission_in_DB
_and_permission_CODEVIEWER() {
  assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, null,
UserRole.CODEVIEWER))
    .isEmpty();
}

@Test
public void
keepAuthorizedProjectIds_returns_empty_for_user_and_any_permission_on_private_project_without_any_permissi
on_in_DB() {
  ProjectPermissions.ALL
    .forEach(perm -> {
      assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, user.getId(), perm))
        .isEmpty();
    });
  assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, user.getId(),
randomPermission))
    .isEmpty();
}

@Test
public void
keepAuthorizedProjectIds_returns_empty_for_group_AnyOne_and_any_permission_on_private_project_without_a
ny_permission_in_DB() {
  ProjectPermissions.ALL
    .forEach(perm -> {
      assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, null, perm))
        .isEmpty();
    });
  assertThat(underTest.keepAuthorizedProjectIds(dbSession, randomPrivateProjectIds, null, randomPermission))
    .isEmpty();
}

@Test
public void keepAuthorizedProjectIds_returns_private_project_if_user_is_granted_project_permission_directly() {
```

```java
    ComponentDto project = db.components().insertPrivateProject(organization);
    ComponentDto otherProject = db.components().insertPrivateProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user, randomPermission, project);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(),
randomPermission))
      .containsOnly(project.getId());
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(), "another
perm"))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(otherProject.getId()), user.getId(),
randomPermission))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), otherUser.getId(),
randomPermission))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectIds_returns_private_project_if_user_is_granted_project_permission_by_group()
{
    ComponentDto project = db.components().insertPrivateProject(organization);
    ComponentDto otherProject = db.components().insertPrivateProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertMember(group1, user);
    db.users().insertProjectPermissionOnGroup(group1, randomPermission, project);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(),
randomPermission))
      .containsOnly(project.getId());
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), user.getId(), "another
perm"))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(otherProject.getId()), user.getId(),
randomPermission))
      .isEmpty();
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, singleton(project.getId()), otherUser.getId(),
randomPermission))
      .isEmpty();
  }

  @Test
  public void user_should_be_authorized() {
    ComponentDto project1 = db.components().insertPrivateProject(organization);
    ComponentDto project2 = db.components().insertPrivateProject(organization);
    ComponentDto project3 = db.components().insertPrivateProject(organization);
    UserDto user = db.users().insertUser("u1");
```

```java
    GroupDto group = db.users().insertGroup(organization);
    db.users().insertProjectPermissionOnUser(user, UserRole.USER, project2);
    db.users().insertProjectPermissionOnUser(user, UserRole.USER, project3);
    db.users().insertMember(group, user);
    db.users().insertProjectPermissionOnGroup(group, UserRole.USER, project1);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, newHashSet(project2.getId(), project3.getId()),
user.getId(), UserRole.USER))
      .containsOnly(project2.getId(), project3.getId());

    // user does not have the role "admin"
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, newHashSet(project2.getId()), user.getId(),
UserRole.ADMIN))
      .isEmpty();

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, Collections.emptySet(), user.getId(),
UserRole.ADMIN))
      .isEmpty();
  }

  @Test
  public void group_should_be_authorized() {
    ComponentDto project1 = db.components().insertPrivateProject(organization);
    ComponentDto project2 = db.components().insertPrivateProject(organization);
    ComponentDto project3 = db.components().insertPrivateProject(organization);
    UserDto user1 = db.users().insertUser("u1");
    GroupDto group = db.users().insertGroup(organization);
    db.users().insertMembers(group, user1);
    db.users().insertProjectPermissionOnUser(user1, UserRole.USER, project1);
    db.users().insertProjectPermissionOnGroup(group, UserRole.USER, project2);
    db.users().insertProjectPermissionOnGroup(group, UserRole.USER, project3);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, newHashSet(project2.getId(), project3.getId()),
user1.getId(), UserRole.USER))
      .containsOnly(project2.getId(), project3.getId());

    // group does not have the role "admin"
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, newHashSet(project2.getId(), project3.getId()),
user1.getId(), UserRole.ADMIN))
      .isEmpty();
  }

  @Test
  public void anonymous_should_be_authorized() {
    ComponentDto project1 = db.components().insertPublicProject(organization);
    ComponentDto project2 = db.components().insertPublicProject(organization);
    UserDto user1 = db.users().insertUser("u1");
    GroupDto group = db.users().insertGroup(organization);
```

```java
    db.users().insertMembers(group, user1);

    assertThat(underTest.keepAuthorizedProjectIds(dbSession, newHashSet(project1.getId(), project2.getId()), null,
UserRole.USER))
      .containsOnly(project1.getId(), project2.getId());

    // group does not have the role "admin"
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, newHashSet(project1.getId()), null, "admin"))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectIds_should_be_able_to_handle_lots_of_projects() {
    List<ComponentDto> projects = IntStream.range(0, 2000).mapToObj(i ->
db.components().insertPublicProject(organization)).collect(Collectors.toList());

    Collection<Long> ids = projects.stream().map(ComponentDto::getId).collect(Collectors.toSet());
    assertThat(underTest.keepAuthorizedProjectIds(dbSession, ids, null, UserRole.USER))
      .containsOnly(ids.toArray(new Long[0]));
  }

  @Test
  public void keepAuthorizedProjectUuids_should_be_able_to_handle_lots_of_projects() {
    List<ComponentDto> projects = IntStream.range(0, 2000).mapToObj(i ->
db.components().insertPublicProject(organization)).collect(Collectors.toList());

    Collection<String> uuids = projects.stream().map(ComponentDto::uuid).collect(Collectors.toSet());
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, uuids, null, UserRole.USER))
      .containsOnly(uuids.toArray(new String[0]));
  }

  @Test
  public void keepAuthorizedUsersForRoleAndProject_returns_empty_if_user_set_is_empty_on_public_project() {
    OrganizationDto organization = db.organizations().insert();
    ComponentDto project = db.components().insertPublicProject(organization);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, Collections.emptySet(),
UserRole.USER, project.getId()))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedUsersForRoleAndProject_returns_empty_for_non_existent_users() {
    ComponentDto project = random.nextBoolean() ? db.components().insertPublicProject(organization) :
db.components().insertPrivateProject(organization);
    Set<Integer> randomNonExistingUserIdsSet = IntStream.range(0, 1 + Math.abs(random.nextInt(5)))
      .map(i -> i + 1_990)
      .boxed()
```

```
      .collect(MoreCollectors.toSet());

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomNonExistingUserIdsSet,
  UserRole.USER, project.getId()))
      .isEmpty();
  }

  @Test
  public void
  keepAuthorizedUsersForRoleAndProject_returns_any_users_for_public_project_without_any_permission_in_DB_a
  nd_permission_USER() {
    ComponentDto project = db.components().insertPublicProject(organization);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomExistingUserIds,
  UserRole.USER, project.getId()))
      .containsAll(randomExistingUserIds);
  }

  @Test
  public void
  keepAuthorizedUsersForRoleAndProject_returns_any_users_for_public_project_without_any_permission_in_DB_a
  nd_permission_CODEVIEWER() {
    ComponentDto project = db.components().insertPublicProject(organization);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomExistingUserIds,
  UserRole.CODEVIEWER, project.getId()))
      .containsAll(randomExistingUserIds);
  }

  @Test
  public void
  keepAuthorizedUsersForRoleAndProject_returns_empty_for_any_users_on_public_project_without_any_permissio
  n_in_DB() {
    ComponentDto project = db.components().insertPublicProject(organization);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomExistingUserIds,
  randomPermission, project.getId()))
      .isEmpty();
  }

  @Test
  public void
  keepAuthorizedUsersForRoleAndProject_returns_user_if_granted_project_permission_directly_on_public_project()
  {
    ComponentDto project = db.components().insertPublicProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user, randomPermission, project);
```

```java
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, project.getId()))
      .containsOnly(user.getId());
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()), "another perm",
project.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(otherUser.getId()),
randomPermission, project.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, otherProject.getId()))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedUsersForRoleAndProject_returns_user_if_granted_project_permission_by_group_on_public_projec
t() {
    ComponentDto project = db.components().insertPublicProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertMember(group1, user);
    db.users().insertProjectPermissionOnGroup(group1, randomPermission, project);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, project.getId()))
      .containsOnly(user.getId());
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()), "another perm",
project.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, otherProject.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(otherUser.getId()),
randomPermission, project.getId()))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedUsersForRoleAndProject_does_not_return_user_if_granted_project_permission_by_AnyOne_on_p
ublic_project() {
    ComponentDto project = db.components().insertPublicProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertProjectPermissionOnAnyone(randomPermission, project);
```

```java
      assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
  randomPermission, project.getId()))
        .isEmpty();
      assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()), "another perm",
  project.getId()))
        .isEmpty();
      assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
  randomPermission, otherProject.getId()))
        .isEmpty();
      assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(otherUser.getId()),
  randomPermission, project.getId()))
        .isEmpty();
  }

  @Test
  public void
  keepAuthorizedUsersForRoleAndProject_returns_empty_for_any_user_on_private_project_without_any_permissio
  n_in_DB_and_permission_USER() {
    ComponentDto project = db.components().insertPrivateProject(organization);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomExistingUserIds,
  UserRole.USER, project.getId()))
      .isEmpty();
  }

  @Test
  public void
  keepAuthorizedUsersForRoleAndProject_returns_empty_for_any_user_on_private_project_without_any_permissio
  n_in_DB_and_permission_CODEVIEWER() {
    ComponentDto project = db.components().insertPrivateProject(organization);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomExistingUserIds,
  UserRole.CODEVIEWER, project.getId()))
      .isEmpty();
  }

  @Test
  public void
  keepAuthorizedUsersForRoleAndProject_returns_empty_for_any_users_and_any_permission_on_private_project_
  without_any_permission_in_DB() {
    ComponentDto project = db.components().insertPrivateProject(organization);

    ProjectPermissions.ALL
      .forEach(perm -> {
        assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomExistingUserIds, perm,
  project.getId()))
          .isEmpty();
      });
```

```java
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, randomExistingUserIds,
randomPermission, project.getId()))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedUsersForRoleAndProject_returns_user_if_granted_project_permission_directly_on_private_project(
) {
    ComponentDto project = db.components().insertPrivateProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user, randomPermission, project);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, project.getId()))
      .containsOnly(user.getId());
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()), "another perm",
project.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(otherUser.getId()),
randomPermission, project.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, otherProject.getId()))
      .isEmpty();
  }

  @Test
  public void
keepAuthorizedUsersForRoleAndProject_returns_user_if_granted_project_permission_by_group_on_private_proje
ct() {
    ComponentDto project = db.components().insertPrivateProject(organization);
    ComponentDto otherProject = db.components().insertPublicProject(organization);
    UserDto otherUser = db.users().insertUser();
    db.users().insertMember(group1, user);
    db.users().insertProjectPermissionOnGroup(group1, randomPermission, project);

    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, project.getId()))
      .containsOnly(user.getId());
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()), "another perm",
project.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(user.getId()),
randomPermission, otherProject.getId()))
      .isEmpty();
    assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession, singleton(otherUser.getId()),
```

```
randomPermission, project.getId()))
    .isEmpty();
}

@Test
public void keep_authorized_users_returns_empty_list_for_role_and_project_for_anonymous() {
  ComponentDto project1 = db.components().insertPrivateProject(organization);
  ComponentDto project2 = db.components().insertPrivateProject(organization);
  ComponentDto project3 = db.components().insertPrivateProject(organization);
  UserDto user1 = db.users().insertUser("u1");
  UserDto user2 = db.users().insertUser("u2");
  UserDto user3 = db.users().insertUser("u3");
  GroupDto group1 = db.users().insertGroup(organization);
  GroupDto group2 = db.users().insertGroup(organization);
  db.users().insertMembers(group1, user1, user2);
  db.users().insertMembers(group2, user3);
  db.users().insertProjectPermissionOnUser(user1, UserRole.USER, project1);
  db.users().insertProjectPermissionOnUser(user2, UserRole.USER, project1);
  db.users().insertProjectPermissionOnUser(user3, UserRole.USER, project1);
  db.users().insertProjectPermissionOnGroup(group2, UserRole.USER, project3);

  assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession,
    // Only 100 and 101 has 'user' role on project
    newHashSet(100, 101, 102), "user", PROJECT_ID)).isEmpty();
}

@Test
public void keepAuthorizedUsersForRoleAndProject_should_be_able_to_handle_lots_of_users() {
  List<UserDto> users = IntStream.range(0, 2000).mapToObj(i ->
db.users().insertUser()).collect(Collectors.toList());

  assertThat(underTest.keepAuthorizedUsersForRoleAndProject(dbSession,
    users.stream().map(UserDto::getId).collect(Collectors.toSet()), "user", PROJECT_ID)).isEmpty();
}

@Test
public void countUsersWithGlobalPermissionExcludingGroupMember() {
  // u1 has the direct permission, u2 and u3 have the permission through their group
  UserDto u1 = db.users().insertUser();
  db.users().insertPermissionOnUser(organization, u1, A_PERMISSION);
  db.users().insertPermissionOnGroup(group1, A_PERMISSION);
  db.users().insertPermissionOnGroup(group1, "another-permission");
  UserDto u2 = db.users().insertUser();
  db.users().insertMember(group1, u2);
  UserDto u3 = db.users().insertUser();
  db.users().insertMember(group1, u3);

  // excluding u2 membership --> remain u1 and u3
```

```java
    int count = underTest.countUsersWithGlobalPermissionExcludingGroupMember(dbSession,
organization.getUuid(), A_PERMISSION, group1.getId(), u2.getId());
    assertThat(count).isEqualTo(2);

    // excluding unknown memberships
    count = underTest.countUsersWithGlobalPermissionExcludingGroupMember(dbSession, organization.getUuid(),
A_PERMISSION, group1.getId(), MISSING_ID);
    assertThat(count).isEqualTo(3);
    count = underTest.countUsersWithGlobalPermissionExcludingGroupMember(dbSession, organization.getUuid(),
A_PERMISSION, MISSING_ID, u2.getId());
    assertThat(count).isEqualTo(3);

    // another organization
    count = underTest.countUsersWithGlobalPermissionExcludingGroupMember(dbSession, DOES_NOT_EXIST,
A_PERMISSION, group1.getId(), u2.getId());
    assertThat(count).isEqualTo(0);

    // another permission
    count = underTest.countUsersWithGlobalPermissionExcludingGroupMember(dbSession, organization.getUuid(),
DOES_NOT_EXIST, group1.getId(), u2.getId());
    assertThat(count).isEqualTo(0);
  }

  @Test
  public void countUsersWithGlobalPermissionExcludingUserPermission() {
    // u1 and u2 have the direct permission, u3 has the permission through his group
    UserDto u1 = db.users().insertUser();
    db.users().insertPermissionOnUser(organization, u1, A_PERMISSION);
    UserDto u2 = db.users().insertUser();
    db.users().insertPermissionOnUser(organization, u2, A_PERMISSION);
    db.users().insertPermissionOnGroup(group1, A_PERMISSION);
    UserDto u3 = db.users().insertUser();
    db.users().insertMember(group1, u3);

    // excluding u2 permission --> remain u1 and u3
    int count = underTest.countUsersWithGlobalPermissionExcludingUserPermission(dbSession,
organization.getUuid(), A_PERMISSION, u2.getId());
    assertThat(count).isEqualTo(2);

    // excluding unknown user
    count = underTest.countUsersWithGlobalPermissionExcludingUserPermission(dbSession, organization.getUuid(),
A_PERMISSION, MISSING_ID);
    assertThat(count).isEqualTo(3);

    // another organization
    count = underTest.countUsersWithGlobalPermissionExcludingUserPermission(dbSession, DOES_NOT_EXIST,
A_PERMISSION, u2.getId());
    assertThat(count).isEqualTo(0);
```

```java
  // another permission
  count = underTest.countUsersWithGlobalPermissionExcludingUserPermission(dbSession, organization.getUuid(),
DOES_NOT_EXIST, u2.getId());
  assertThat(count).isEqualTo(0);
}

@Test
public void selectOrganizationUuidsOfUserWithGlobalPermission_returns_empty_set_if_user_does_not_exist() {
  // another user
  db.users().insertPermissionOnUser(user, ADMINISTER_QUALITY_GATES);

  Set<String> orgUuids = underTest.selectOrganizationUuidsOfUserWithGlobalPermission(dbSession,
MISSING_ID, SYSTEM_ADMIN);

  assertThat(orgUuids).isEmpty();
}

@Test
public void
selectOrganizationUuidsOfUserWithGlobalPermission_returns_empty_set_if_user_does_not_have_permission_at_a
ll() {
  db.users().insertPermissionOnUser(user, ADMINISTER_QUALITY_GATES);
  // user is not part of this group
  db.users().insertPermissionOnGroup(group1, SCAN_EXECUTION);

  Set<String> orgUuids = underTest.selectOrganizationUuidsOfUserWithGlobalPermission(dbSession, user.getId(),
SCAN_EXECUTION);

  assertThat(orgUuids).isEmpty();
}

@Test
public void
selectOrganizationUuidsOfUserWithGlobalPermission_returns_organizations_on_which_user_has_permission() {
  db.users().insertPermissionOnGroup(group1, SCAN_EXECUTION);
  db.users().insertPermissionOnGroup(group2, QUALITY_GATE_ADMIN);
  db.users().insertMember(group1, user);
  db.users().insertMember(group2, user);

  Set<String> orgUuids = underTest.selectOrganizationUuidsOfUserWithGlobalPermission(dbSession, user.getId(),
SCAN_EXECUTION);

  assertThat(orgUuids).containsExactly(group1.getOrganizationUuid());
}

@Test
public void
```

```java
selectOrganizationUuidsOfUserWithGlobalPermission_handles_user_permissions_and_group_permissions() {
  // organization: through group membership
  db.users().insertPermissionOnGroup(group1, SCAN_EXECUTION);
  db.users().insertMember(group1, user);

  // org2 : direct user permission
  OrganizationDto org2 = db.organizations().insert();
  db.users().insertPermissionOnUser(org2, user, SCAN_EXECUTION);

  // org3 : another permission QUALITY_GATE_ADMIN
  OrganizationDto org3 = db.organizations().insert();
  db.users().insertPermissionOnUser(org3, user, QUALITY_GATE_ADMIN);

  // exclude project permission
  db.users().insertProjectPermissionOnUser(user, UserRole.ADMIN, db.components().insertPrivateProject());

  Set<String> orgUuids = underTest.selectOrganizationUuidsOfUserWithGlobalPermission(dbSession, user.getId(),
SCAN_EXECUTION);

  assertThat(orgUuids).containsOnly(organization.getUuid(), org2.getUuid());
}

@Test
public void selectOrganizationUuidsOfUserWithGlobalPermission_ignores_anonymous_permissions() {
  db.users().insertPermissionOnAnyone(organization, SCAN);
  db.users().insertPermissionOnUser(organization, user, ADMINISTER_QUALITY_GATES);

  Set<String> orgUuids = underTest.selectOrganizationUuidsOfUserWithGlobalPermission(dbSession, user.getId(),
SCAN.getKey());

  assertThat(orgUuids).isEmpty();
}

@Test
public void
selectProjectPermissionsOfAnonymous_returns_permissions_of_anonymous_user_on_specified_public_project() {
  ComponentDto project = db.components().insertPublicProject(organization);
  db.users().insertProjectPermissionOnAnyone("p1", project);
  db.users().insertProjectPermissionOnUser(db.users().insertUser(), "p2", project);
  ComponentDto otherProject = db.components().insertPublicProject();
  db.users().insertProjectPermissionOnAnyone("p3", otherProject);

  assertThat(underTest.selectProjectPermissionsOfAnonymous(dbSession, project.uuid())).containsOnly("p1");
}

@Test
public void selectProjectPermissionsOfAnonymous_returns_empty_set_when_project_does_not_exist() {
  assertThat(underTest.selectProjectPermissionsOfAnonymous(dbSession, "does_not_exist")).isEmpty();
```

```java
}

@Test
public void selectProjectPermissions_returns_empty_set_when_logged_in_user_and_project_does_not_exist() {
  assertThat(underTest.selectProjectPermissions(dbSession, "does_not_exist", user.getId())).isEmpty();
}

@Test
public void
selectProjectPermissions_returns_permissions_of_logged_in_user_on_specified_public_project_through_anonymou
s_permissions() {
  ComponentDto project = db.components().insertPublicProject(organization);
  db.users().insertProjectPermissionOnAnyone("p1", project);
  db.users().insertProjectPermissionOnAnyone("p2", project);

  assertThat(underTest.selectProjectPermissions(dbSession, project.uuid(), user.getId())).containsOnly("p1", "p2");
}

@Test
public void selectProjectPermissions_returns_permissions_of_logged_in_user_on_specified_project() {
  ComponentDto project = db.components().insertPrivateProject(organization);
  db.users().insertProjectPermissionOnUser(user, UserRole.CODEVIEWER, project);
  db.users().insertProjectPermissionOnUser(db.users().insertUser(), UserRole.ISSUE_ADMIN, project);

  assertThat(underTest.selectProjectPermissions(dbSession, project.uuid(),
user.getId())).containsOnly(UserRole.CODEVIEWER);
}

@Test
public void
selectProjectPermissions_returns_permissions_of_logged_in_user_on_specified_project_through_group_membershi
p() {
  ComponentDto project = db.components().insertPrivateProject(organization);
  db.users().insertProjectPermissionOnGroup(group1, UserRole.CODEVIEWER, project);
  db.users().insertProjectPermissionOnGroup(group2, UserRole.ISSUE_ADMIN, project);
  db.users().insertMember(group1, user);

  assertThat(underTest.selectProjectPermissions(dbSession, project.uuid(),
user.getId())).containsOnly(UserRole.CODEVIEWER);
}

@Test
public void
selectProjectPermissions_returns_permissions_of_logged_in_user_on_specified_private_project_through_all_possi
ble_configurations() {
  ComponentDto project = db.components().insertPrivateProject(organization);
  db.users().insertProjectPermissionOnUser(user, UserRole.CODEVIEWER, project);
  db.users().insertProjectPermissionOnGroup(group1, UserRole.USER, project);
```

```java
    db.users().insertMember(group1, user);

    assertThat(underTest.selectProjectPermissions(dbSession, project.uuid(),
user.getId())).containsOnly(UserRole.CODEVIEWER, UserRole.USER);
  }

  @Test
  public void
selectProjectPermissions_returns_permissions_of_logged_in_user_on_specified_public_project_through_all_possible_configurations() {
    ComponentDto project = db.components().insertPublicProject(organization);
    db.users().insertProjectPermissionOnUser(user, "p1", project);
    db.users().insertProjectPermissionOnAnyone("p2", project);
    db.users().insertProjectPermissionOnGroup(group1, "p3", project);
    db.users().insertMember(group1, user);

    assertThat(underTest.selectProjectPermissions(dbSession, project.uuid(), user.getId())).containsOnly("p1", "p2",
"p3");
  }

  @Test
  public void keepAuthorizedProjectUuids_filters_projects_authorized_to_logged_in_user_by_direct_permission() {
    ComponentDto privateProject = db.components().insertPrivateProject(organization);
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    UserDto user = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user, UserRole.ADMIN, privateProject);

    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(privateProject.uuid(),
publicProject.uuid()), user.getId(), UserRole.ADMIN))
      .containsOnly(privateProject.uuid());
    // user does not have the permission "issueadmin"
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(privateProject.uuid(),
publicProject.uuid()), user.getId(), UserRole.ISSUE_ADMIN))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectUuids_filters_projects_authorized_to_logged_in_user_by_group_permission() {
    ComponentDto privateProject = db.components().insertPrivateProject(organization);
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    UserDto user = db.users().insertUser();
    GroupDto group = db.users().insertGroup(organization);
    db.users().insertMember(group, user);
    db.users().insertProjectPermissionOnGroup(group, UserRole.ADMIN, privateProject);

    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(privateProject.uuid(),
publicProject.uuid()), user.getId(), UserRole.ADMIN))
      .containsOnly(privateProject.uuid());
```

```java
    // user does not have the permission "issueadmin"
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(privateProject.uuid(),
publicProject.uuid()), user.getId(), UserRole.ISSUE_ADMIN))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectUuids_returns_empty_list_if_input_is_empty() {
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    UserDto user = db.users().insertUser();

    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, Collections.emptySet(), user.getId(),
UserRole.USER))
      .isEmpty();

    // projects do not exist
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet("does_not_exist"), user.getId(),
UserRole.USER))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectUuids_returns_empty_list_if_input_does_not_reference_existing_projects() {
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    UserDto user = db.users().insertUser();

    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet("does_not_exist"), user.getId(),
UserRole.USER))
      .isEmpty();
  }

  @Test
  public void keepAuthorizedProjectUuids_returns_public_projects_if_permission_USER_or_CODEVIEWER() {
    ComponentDto publicProject = db.components().insertPublicProject(organization);
    UserDto user = db.users().insertUser();

    // logged-in user
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(publicProject.uuid()), user.getId(),
UserRole.CODEVIEWER))
      .containsOnly(publicProject.uuid());
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(publicProject.uuid()), user.getId(),
UserRole.USER))
      .containsOnly(publicProject.uuid());
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(publicProject.uuid()), user.getId(),
UserRole.ADMIN))
      .isEmpty();

    // anonymous
```

```java
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(publicProject.uuid()), null,
UserRole.CODEVIEWER))
      .containsOnly(publicProject.uuid());
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(publicProject.uuid()), null,
UserRole.USER))
      .containsOnly(publicProject.uuid());
    assertThat(underTest.keepAuthorizedProjectUuids(dbSession, newHashSet(publicProject.uuid()), null,
UserRole.ADMIN))
      .isEmpty();
  }

  @Test
  public void
selectQualityProfileAdministratorLogins_return_users_with_quality_profile_administrator_permission() {
    OrganizationDto organization1 = db.organizations().insert();
    UserDto user1 = db.users().insertUser();
    db.users().insertPermissionOnUser(organization1, user1, ADMINISTER_QUALITY_PROFILES);
    OrganizationDto organization2 = db.organizations().insert();
    UserDto user2 = db.users().insertUser();
    db.users().insertPermissionOnUser(organization2, user2, ADMINISTER_QUALITY_PROFILES);

    List<String> logins = underTest.selectQualityProfileAdministratorLogins(dbSession);

    assertThat(logins).containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin());
  }

  @Test
  public void selectQualityProfileAdministratorLogins_return_users_within_quality_profile_administrator_group() {
    OrganizationDto organization1 = db.organizations().insert();
    GroupDto qualityProfileAdministratorGroup1 = db.users().insertGroup(organization1);
    db.users().insertPermissionOnGroup(qualityProfileAdministratorGroup1,
ADMINISTER_QUALITY_PROFILES);
    UserDto user1 = db.users().insertUser();
    db.users().insertMember(qualityProfileAdministratorGroup1, user1);
    OrganizationDto organization2 = db.organizations().insert();
    GroupDto qualityProfileAdministratorGroup2 = db.users().insertGroup(organization2);
    db.users().insertPermissionOnGroup(qualityProfileAdministratorGroup2,
ADMINISTER_QUALITY_PROFILES);
    UserDto user2 = db.users().insertUser();
    db.users().insertMember(qualityProfileAdministratorGroup2, user2);

    List<String> logins = underTest.selectQualityProfileAdministratorLogins(dbSession);

    assertThat(logins).containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin());
  }

  @Test
  public void selectQualityProfileAdministratorLogins_does_not_return_non_quality_profile_administrator_logins()
```

```java
{
  OrganizationDto organization1 = db.organizations().insert();
  UserDto user1 = db.users().insertUser();
  db.users().insertPermissionOnUser(organization1, user1, ADMINISTER);
  db.users().insertUser();

  List<String> logins = underTest.selectQualityProfileAdministratorLogins(dbSession);

  assertThat(logins).isEmpty();
}

@Test
public void selectGlobalAdministratorLogins() {
  OrganizationDto organization1 = db.organizations().insert();
  UserDto user1 = db.users().insertUser();
  db.users().insertPermissionOnUser(organization1, user1, ADMINISTER);
  OrganizationDto organization2 = db.organizations().insert();
  UserDto user2 = db.users().insertUser();
  db.users().insertPermissionOnUser(organization2, user2, ADMINISTER);

  GroupDto administratorGroup2 = db.users().insertGroup(organization2);
  db.users().insertPermissionOnGroup(administratorGroup2, ADMINISTER);
  UserDto user3 = db.users().insertUser();
  db.users().insertMember(administratorGroup2, user3);

  ComponentDto project = db.components().insertPrivateProject();

  UserDto user4 = db.users().insertUser();
  db.users().insertPermissionOnUser(organization1, user4, ADMINISTER_QUALITY_PROFILES);
  db.users().insertProjectPermissionOnUser(user4, "admin", project);
  db.users().insertUser();

  List<String> logins = underTest.selectGlobalAdministratorLogins(dbSession);

  assertThat(logins).containsExactlyInAnyOrder(user1.getLogin(), user2.getLogin(), user3.getLogin());
}

@Test
public void keepAuthorizedLoginsOnProject_return_correct_users_on_public_project() {
  ComponentDto project = db.components().insertPublicProject(organization);

  UserDto user1 = db.users().insertUser();

  // admin with "direct" ADMIN role
  UserDto admin1 = db.users().insertUser();
  db.users().insertProjectPermissionOnUser(admin1, UserRole.ADMIN, project);

  // admin2 with ADMIN role through group
```

```java
    UserDto admin2 = db.users().insertUser();
    GroupDto adminGroup = db.users().insertGroup(organization, "ADMIN");
    db.users().insertMember(adminGroup, admin2);
    db.users().insertProjectPermissionOnGroup(adminGroup, UserRole.ADMIN, project);

    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, newHashSet(user1.getLogin()),
project.getKey(), UserRole.USER))
      .containsOnly(user1.getLogin());
    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, newHashSet(user1.getLogin(),
admin1.getLogin(), admin2.getLogin()), project.getKey(), UserRole.USER))
      .containsOnly(user1.getLogin(), admin1.getLogin(), admin2.getLogin());
    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, newHashSet(user1.getLogin(),
admin1.getLogin(), admin2.getLogin()), project.getKey(), UserRole.ADMIN))
      .containsOnly(admin1.getLogin(), admin2.getLogin());
  }

  @Test
  public void keepAuthorizedLoginsOnProject_return_correct_users_on_private_project() {
    ComponentDto project = db.components().insertPrivateProject(organization);

    GroupDto userGroup = db.users().insertGroup(organization, "USERS");
    GroupDto adminGroup = db.users().insertGroup(organization, "ADMIN");
    db.users().insertProjectPermissionOnGroup(userGroup, UserRole.USER, project);
    db.users().insertProjectPermissionOnGroup(adminGroup, UserRole.ADMIN, project);

    // admin with "direct" ADMIN role
    UserDto admin1 = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(admin1, UserRole.ADMIN, project);

    // admin2 with ADMIN role through group
    UserDto admin2 = db.users().insertUser();
    db.users().insertMember(adminGroup, admin2);

    // user1 with "direct" USER role
    UserDto user1 = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user1, UserRole.USER, project);

    // user2 with USER role through group
    UserDto user2 = db.users().insertUser();
    db.users().insertMember(userGroup, user2);

    // user without role
    UserDto userWithNoRole = db.users().insertUser();

    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, newHashSet(userWithNoRole.getLogin()),
project.getKey(), UserRole.USER))
      .isEmpty();
    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, newHashSet(user1.getLogin()),
```

```
    project.getKey(), UserRole.USER))
      .containsOnly(user1.getLogin());

    Set<String> allLogins = newHashSet(admin1.getLogin(), admin2.getLogin(), user1.getLogin(), user2.getLogin(),
  userWithNoRole.getLogin());

    // Admin does not have the USER permission set
    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, allLogins, project.getKey(), UserRole.USER))
      .containsOnly(user1.getLogin(), user2.getLogin());
    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, allLogins, project.getKey(),
  UserRole.ADMIN))
      .containsOnly(admin1.getLogin(), admin2.getLogin());
  }

  @Test
  public void keepAuthorizedLoginsOnProject_return_correct_users_on_branch() {
    ComponentDto project = db.components().insertPrivateProject(organization);
    ComponentDto branch = db.components().insertProjectBranch(project, c ->
  c.setBranchType(random.nextBoolean() ? BranchType.SHORT : BranchType.LONG));

    GroupDto userGroup = db.users().insertGroup(organization, "USERS");
    GroupDto adminGroup = db.users().insertGroup(organization, "ADMIN");
    db.users().insertProjectPermissionOnGroup(userGroup, UserRole.USER, project);
    db.users().insertProjectPermissionOnGroup(adminGroup, UserRole.ADMIN, project);

    // admin with "direct" ADMIN role
    UserDto admin1 = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(admin1, UserRole.ADMIN, project);

    // admin2 with ADMIN role through group
    UserDto admin2 = db.users().insertUser();
    db.users().insertMember(adminGroup, admin2);

    // user1 with "direct" USER role
    UserDto user1 = db.users().insertUser();
    db.users().insertProjectPermissionOnUser(user1, UserRole.USER, project);

    // user2 with USER role through group
    UserDto user2 = db.users().insertUser();
    db.users().insertMember(userGroup, user2);

    // user without role
    UserDto userWithNoRole = db.users().insertUser();

    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, newHashSet(userWithNoRole.getLogin()),
  branch.getKey(), UserRole.USER))
      .isEmpty();
    assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, newHashSet(user1.getLogin()),
```

```java
      branch.getKey(), UserRole.USER))
        .containsOnly(user1.getLogin());

      Set<String> allLogins = newHashSet(admin1.getLogin(), admin2.getLogin(), user1.getLogin(), user2.getLogin(),
userWithNoRole.getLogin());

      // Admin does not have the USER permission set
      assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, allLogins, branch.getKey(), UserRole.USER))
        .containsOnly(user1.getLogin(), user2.getLogin());
      assertThat(underTest.keepAuthorizedLoginsOnProject(dbSession, allLogins, branch.getKey(),
UserRole.ADMIN))
        .containsOnly(admin1.getLogin(), admin2.getLogin());
  }
}
/*
 * SonarQube
 * Copyright (C) 2009-2018 SonarSource SA
 * mailto:info AT sonarsource DOT com
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
 */
package org.sonar.server.platform.db.migration.version.v67;

import java.sql.SQLException;
import org.sonar.db.Database;
import org.sonar.server.platform.db.migration.step.DataChange;
import org.sonar.server.platform.db.migration.step.MassUpdate;

public class DropOldLicenses extends DataChange {

  private static final String LICENSE_HASH_SECURED_SUFFIX = ".licenseHash.secured";
  private static final String LICENSE_SECURED_SUFFIX = ".license.secured";

  public DropOldLicenses(Database db) {
    super(db);
  }
```

```java
  @Override
  protected void execute(Context context) throws SQLException {
    MassUpdate massUpdate = context.prepareMassUpdate();
    massUpdate.select("select prop_key from properties where prop_key like ?")
      .setString(1, "%" + LICENSE_HASH_SECURED_SUFFIX);
    massUpdate.update("delete from properties where prop_key = ? or prop_key = ?");
    massUpdate.rowPluralName("old license properties");
    massUpdate.execute((row, update) -> {
     String licenseHashKey = row.getString(1);
     String licenseKey = licenseHashKey.replace(LICENSE_HASH_SECURED_SUFFIX, "") +
LICENSE_SECURED_SUFFIX;
     update.setString(1, licenseHashKey);
     update.setString(2, licenseKey);
     return true;
    });
  }
}
```

<div align="center">

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

</div>

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

 This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

 0. Additional Definitions.

 As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

 "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

 An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

 A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

   a) under this License, provided that you make a good faith effort to
   ensure that, in the event an Application does not supply the
   function or data, the facility still operates, and performs
   whatever part of its purpose remains meaningful, or

   b) under the GNU GPL, with none of the additional permissions of
   this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

   a) Give prominent notice with each copy of the object code that the
   Library is used in it and that the Library and its use are
   covered by this License.

   b) Accompany the object code with a copy of the GNU GPL and this license
   document.

4. Combined Works.

  You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

  a) Give prominent notice with each copy of the Combined Work that
  the Library is used in it and that the Library and its use are
  covered by this License.

  b) Accompany the Combined Work with a copy of the GNU GPL and this license
  document.

  c) For a Combined Work that displays copyright notices during
  execution, include the copyright notice for the Library among
  these notices, as well as a reference directing the user to the
  copies of the GNU GPL and this license document.

  d) Do one of the following:

    0) Convey the Minimal Corresponding Source under the terms of this
    License, and the Corresponding Application Code in a form
    suitable for, and under terms that permit, the user to
    recombine or relink the Application with a modified version of
    the Linked Version to produce a modified Combined Work, in the
    manner specified by section 6 of the GNU GPL for conveying
    Corresponding Source.

    1) Use a suitable shared library mechanism for linking with the
    Library. A suitable mechanism is one that (a) uses at run time
    a copy of the Library already present on the user's computer
    system, and (b) will operate properly with a modified version
    of the Library that is interface-compatible with the Linked
    Version.

  e) Provide Installation Information, but only if you would otherwise
  be required to provide such information under section 6 of the
  GNU GPL, and only to the extent that such information is
  necessary to install and execute a modified version of the
  Combined Work produced by recombining or relinking the
  Application with a modified version of the Linked Version. (If
  you use option 4d0, the Installation Information must accompany
  the Minimal Corresponding Source and Corresponding Application
  Code. If you use option 4d1, you must provide the Installation
  Information in the manner specified by section 6 of the GNU GPL
  for conveying Corresponding Source.)

5. Combined Libraries.

 You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

  a) Accompany the combined library with a copy of the same work based
  on the Library, uncombined with any other library facilities,
  conveyed under the terms of this License.

  b) Give prominent notice with the combined library that part of it
  is a work based on the Library, and explaining where to find the
  accompanying uncombined form of the same work.

 6. Revised Versions of the GNU Lesser General Public License.

 The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

 Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

 If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.
The person or persons who have associated work with this document (the
"Dedicator" or "Certifier") hereby either (a) certifies that, to the best of
his knowledge, the work of authorship identified is in the public domain of
the country from which the work is published, or (b) hereby dedicates whatever
copyright the dedicators holds in the work of authorship identified below (the
"Work") to the public domain. A certifier, moreover, dedicates any copyright
interest he may have in the associated work, and for these purposes, is
described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this work. Certifier recognizes that his good faith efforts may not shield him from liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to the detriment of the Dedicator's heirs and successors. Dedicator intends this dedication to be an overt act of relinquishment in perpetuity of all present and future rights under copyright law, whether vested or contingent, in the Work. Dedicator understands that such relinquishment of all rights includes the relinquishment of all rights to enforce (by lawsuit or otherwise) those copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not

standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

```
/*
 * Copyright (c) 2004-2007 QOS.ch
 * All rights reserved.
 *
 * Permission is hereby granted, free  of charge, to any person obtaining
 * a  copy  of this  software  and  associated  documentation files  (the
 * "Software"), to  deal in  the Software without  restriction, including
 * without limitation  the rights to  use, copy, modify,  merge, publish,
 * distribute,  sublicense, and/or sell  copies of  the Software,  and to
 * permit persons to whom the Software  is furnished to do so, subject to
 * the following conditions:
 *
 * The  above  copyright  notice  and  this permission  notice  shall  be
 * included in all copies or substantial portions of the Software.
 *
 * THE  SOFTWARE IS  PROVIDED  "AS  IS", WITHOUT  WARRANTY  OF ANY  KIND,
 * EXPRESS OR  IMPLIED, INCLUDING  BUT NOT LIMITED  TO THE  WARRANTIES OF
 * MERCHANTABILITY,   FITNESS   FOR   A   PARTICULAR   PURPOSE   AND
 * NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
 * LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
 * OF CONTRACT, TORT OR OTHERWISE,  ARISING FROM, OUT OF OR IN CONNECTION
 * WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
 */
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided

that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

 END OF TERMS AND CONDITIONS
The MIT License (MIT)

Copyright (c) 2000 - 2013 The Legion of the Bouncy Castle Inc.
                    (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
(BSD License: http://www.opensource.org/licenses/bsd-license)

Copyright (c) 2011, Joe Walnes, Aslak Hellesy and contributors
All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, are permitted provided that the
following conditions are met:

* Redistributions of source code must retain the above
 copyright notice, this list of conditions and the
 following disclaimer.

* Redistributions in binary form must reproduce the above
 copyright notice, this list of conditions and the
 following disclaimer in the documentation and/or other
 materials provided with the distribution.

* Neither the name of the Webbit nor the names of
 its contributors may be used to endorse or promote products
 derived from this software without specific prior written
 permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
The person or persons who have associated work with this document (the
"Dedicator" or "Certifier") hereby either (a) certifies that, to the best of
his knowledge, the work of authorship identified is in the public domain of
the country from which the work is published, or (b) hereby dedicates whatever
copyright the dedicators holds in the work of authorship identified below (the
"Work") to the public domain. A certifier, moreover, dedicates any copyright
interest he may have in the associated work, and for these purposes, is
described as a "dedicator" below.

A certifier has taken reasonable steps to verify the copyright status of this
work. Certifier recognizes that his good faith efforts may not shield him from
liability if in fact the work certified is not in the public domain.

Dedicator makes this dedication for the benefit of the public at large and to
the detriment of the Dedicator's heirs and successors. Dedicator intends this
dedication to be an overt act of relinquishment in perpetuate of all present
and future rights under copyright law, whether vested or contingent, in the
Work. Dedicator understands that such relinquishment of all rights includes
the relinquishment of all rights to enforce (by lawsuit or otherwise) those
copyrights in the Work.

Dedicator recognizes that, once placed in the public domain, the Work may be
freely reproduced, distributed, transmitted, used, modified, built upon, or
otherwise exploited by anyone for any purpose, commercial or non-commercial,
and in any way, including by methods that have not yet been invented or
conceived.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

 The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

 This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

 When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

 To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

 For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

 We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal

permission to copy, distribute and/or modify the library.

 To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.

 Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

 Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

 When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

 We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

 For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard. To achieve this, non-free programs must be
allowed to use the library. A more frequent case is that a free
library does the same job as widely used non-free libraries. In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

 In other cases, permission to use a particular library in non-free

programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

 Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

 The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

  GNU LESSER GENERAL PUBLIC LICENSE
 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

 0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

 A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

 The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

 "Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

 Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based

on the Library (independent of the use of the Library in a tool for
writing it). Whether that is true depends on what the Library does
and what the program that uses the Library does.

 1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

 You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

 2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

  a) The modified work must itself be a software library.

  b) You must cause the files modified to carry prominent notices
  stating that you changed the files and the date of any change.

  c) You must cause the whole of the work to be licensed at no
  charge to all third parties under the terms of this License.

  d) If a facility in the modified Library refers to a function or a
  table of data to be supplied by an application program that uses
  the facility, other than as an argument passed when the facility
  is invoked, then you must make a good faith effort to ensure that,
  in the event an application does not supply such function or
  table, the facility still operates, and performs whatever part of
  its purpose remains meaningful.

  (For example, a function in a library to compute square roots has
  a purpose that is entirely well-defined independent of the
  application. Therefore, Subsection 2d requires that any
  application-supplied function or table used by this function must
  be optional: if the application does not supply it, the square
  root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those

sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

 3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

 Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

 This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

 4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

 If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

 5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or

linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

 However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

 When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

 If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

 Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

 6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

 You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

  a) Accompany the work with the complete corresponding
  machine-readable source code for the Library including whatever
  changes were used in the work (which must be distributed under
  Sections 1 and 2 above); and, if the work is an executable linked

with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

 For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

 It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

 7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise

permitted, and provided that you do these two things:

   a) Accompany the combined library with a copy of the same work
   based on the Library, uncombined with any other library
   facilities. This must be distributed under the terms of the
   Sections above.

   b) Give prominent notice with the combined library of the fact
   that part of it is a work based on the Library, and explaining
   where to find the accompanying uncombined form of the same work.

 8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License. Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

 9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

 10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

 11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded. In such case, this License incorporates the limitation as if
written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation. If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission. For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, 5th Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

Apache Log4j API

Copyright 1999-2017 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).
ANTLR 2 License

We reserve no legal rights to the ANTLR--it is fully in the public domain. An individual or company may do whatever they wish with source code distributed with ANTLR or the code generated by ANTLR, including the incorporation of ANTLR, or its output, into commerical software.

We encourage users to develop software with ANTLR. However, we do ask that credit is given to us for developing ANTLR. By "credit", we mean that if you use ANTLR or incorporate any source code into one of your programs (commercial product, research project, or otherwise) that you acknowledge this fact somewhere in the documentation, research report, etc... If you like ANTLR and have developed a nice tool with the output, please mention that you developed it using ANTLR. In addition, we ask that the headers remain intact in our source code. As long as these guidelines are kept, we expect to continue enhancing this system and expect to make other tools available as they are completed.

In countries where the Public Domain status of the work may not be valid, the author grants a copyright licence to the general public to deal in the work without restriction and permission to sublicence derivates under the terms of any (OSI approved) Open Source licence.

The Python parser generator code under antlr/actions/python/ is covered by the 3-clause BSD licence (this part is included in the binary JAR files); the run-time part under lib/python/ is covered by the GNU GPL, version 3 or later (this part is not included in the binary JAR files). See [1] for the full details.

https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=750643#80%22
ASM 4 License

Copyright (c) 2000-2011 INRIA, France Telecom
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"

AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Apache Lucene
Copyright 2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/
License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
 http://snowball.tartarus.org/

The KStem stemmer in
 analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, and Hindi analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides

in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

 mecab-ipadic-2.7.0-20070801

which can be obtained from

 http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

 http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801 Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages

whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out
of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification, copying and distribution of the program and the

production of secondary products from the program.

In the case where the program, whether in its original form or modified, was distributed or delivered to or received by a user from any person, organization or entity other than ICOT, unless it makes or grants independently of ICOT any specific warranty to the user in writing, such person, organization or entity, will also be exempted from and not be held liable to the user for any such damages as noted above as far as the program is concerned.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate

as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify

the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include

the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a

file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier

identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was

derived from unicode conversion examples available at

http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright

from those sources:

/*

* Copyright 2001-2004 Unicode, Inc.

*

* Disclaimer

*

* This source code is provided as is by Unicode, Inc. No claims are

* made as to fitness for any particular purpose. No warranties of any

* kind are expressed or implied. The recipient agrees to determine

* applicability of information provided. If this file has been

* purchased on magnetic or optical media from Unicode, Inc., the

* sole remedy for any claim will be exchange of defective media

* within 90 days of receipt.

*

* Limitations on Rights to Redistribute This Code

*

* Unicode, Inc. hereby grants the right to freely use the information

* supplied in this file in the creation of products supporting the

* Unicode Standard, and to make copies of this file in any form

* for internal or external distribution as long as this notice

* remains attached.

*/

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was derived from Python 2.4.2 sources available at http://www.python.org. Full license is here:

 http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was derived from Python 3.1.2 sources available at http://www.python.org. Full license is here:

 http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was derived from Brics automaton sources available at www.brics.dk/automaton/. Here is the copyright from those sources:

/*
* Copyright (c) 2001-2009 Anders Moeller
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. The name of the author may not be used to endorse or promote products
*   derived from this software without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
* DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
* THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
* (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
* THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton were automatically generated with the moman/finenight FSA package. Here is the copyright for those sources:

The following license applies to the Snowball stemmers:

The following license applies to the KStemmer:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and
"University of Massachusetts" must not be used to endorse or promote products
derived from this software without prior written permission. To obtain
permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

  * Neither the name of Morfologik nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski, Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Sonar, open source software quality management tool.
Copyright (C) 2008-2012 SonarSource
mailto:contact AT sonarsource DOT com

Sonar is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 3 of the License, or (at your option) any later version.

Sonar is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with Sonar; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02
This bundle contains Java Service Wrapper scripts and binaries version 3.2.3 from Tanukisoftware published
under the following license:

  Copyright (c) 2001 Silver Egg Technology

  Permission is hereby granted, free of charge, to any person
  obtaining a copy of this software and associated documentation
  files (the "Software"), to deal in the Software without
  restriction, including without limitation the rights to use,
  copy, modify, merge, publish, distribute, sub-license, and/or
  sell copies of the Software, and to permit persons to whom the
  Software is furnished to do so, subject to the following
  conditions:

  The above copyright notice and this permission notice shall be
  included in all copies or substantial portions of the Software.

  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
  EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
  OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
  NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
  HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
  WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
  FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
  OTHER DEALINGS IN THE SOFTWARE.

  http://web.archive.org/web/20070622164715/wrapper.tanukisoftware.org/doc/english/license.html

Elasticsearch
Copyright 2009-2017 Elasticsearch

This product includes software developed by The Apache Software Foundation
(http://www.apache.org/).

===
HdrHistogram LICENSE
===
The code in this repository code was Written by Gil Tene, Michael Barker,
and Matt Warren, and released to the public domain, as explained at
http://creativecommons.org/publicdomain/zero/1.0/

For users of this code who wish to consume it under the "BSD" license
rather than under the public domain or CC0 contribution text mentioned
above, the code found under this directory is *also* provided under the
following license (commonly referred to as the BSD 2-Clause License). This
license does not detract from the above stated release of the code into
the public domain, and simply represents an additional license granted by
the Author.

** Beginning of "BSD 2-Clause License" text. **

CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

==
compiler LICENSE
==
Copyright 2010 RightTime, Inc.

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.

==
groovy NOTICE
==
Apache Groovy
Copyright 2003-2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==

groovy LICENSE

==

/*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

*   http://www.apache.org/licenses/LICENSE-2.0

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*

*/

==

hppc NOTICE

==

ACKNOWLEDGEMENT

===============

HPPC borrowed code, ideas or both from:

* Apache Lucene, http://lucene.apache.org/

  (Apache license)

* Fastutil, http://fastutil.di.unimi.it/

  (Apache license)

* Koloboke, https://github.com/OpenHFT/Koloboke

  (Apache license)

==

hppc LICENSE

==

                    Apache License
                 Version 2.0, January 2004
              http://www.apache.org/licenses/

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

    "License" shall mean the terms and conditions for use, reproduction,
    and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright 2010-2013, Carrot Search s.c., Boznicza 11/56, Poznan, Poland

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
jcodings NOTICE
==
JCodings is released under the MIT License.

==
jcodings LICENSE
==
Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

==
jna LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

===
joda-time NOTICE
===
==
= NOTICE file corresponding to section 4d of the Apache License Version 2.0 =
==
This product includes software developed by
Joda.org (http://www.joda.org/).

===
joda-time LICENSE
===

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common

control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

===
joni NOTICE
===
Joni is released under the MIT License.

==
joni LICENSE
==

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

==
jopt-simple LICENSE
==
/*
The MIT License

Copyright (c) 2004-2015 Paul R. Holser, Jr.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
*/
```

==
jts LICENSE

==

GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

 This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

 0. Additional Definitions.

 As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

 "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

 An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

 A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

 The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

 The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.

b) Accompany the Combined Work with a copy of the GNU GPL and this license document.

c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the
Library.

===
log4j NOTICE
===
Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

===
log4j LICENSE
===

                        Apache License
                  Version 2.0, January 2004
                http://www.apache.org/licenses/

    TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
log4j-api NOTICE
==
Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

==
log4j-api LICENSE
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,

reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

===
log4j-core NOTICE
===
Apache log4j
Copyright 2007 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

===
log4j-core LICENSE
===

                 Apache License
               Version 2.0, January 2004
             http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent

to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work,

excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any

risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!) The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
   file or class name and description of purpose be included on the
   same "printed page" as the copyright notice for easier
   identification within third-party archives.

Copyright 1999-2005 The Apache Software Foundation

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
===============================================================================
lucene NOTICE
===============================================================================
```

Apache Lucene
Copyright 2014 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Includes software from other Apache Software Foundation projects,
including, but not limited to:
- Apache Ant
- Apache Jakarta Regexp
- Apache Commons
- Apache Xerces

ICU4J, (under analysis/icu) is licensed under an MIT styles license
and Copyright (c) 1995-2008 International Business Machines Corporation and others

Some data files (under analysis/icu/src/data) are derived from Unicode data such
as the Unicode Character Database. See http://unicode.org/copyright.html for more
details.

Brics Automaton (under core/src/java/org/apache/lucene/util/automaton) is
BSD-licensed, created by Anders Mller. See http://www.brics.dk/automaton/

The levenshtein automata tables (under core/src/java/org/apache/lucene/util/automaton) were
automatically generated with the moman/finenight FSA library, created by
Jean-Philippe Barrette-LaPierre. This library is available under an MIT license,
see http://sites.google.com/site/rrettesite/moman and
http://bitbucket.org/jpbarrette/moman/overview/

The class org.apache.lucene.util.WeakIdentityMap was derived from
the Apache CXF project and is Apache License 2.0.

The Google Code Prettify is Apache License 2.0.
See http://code.google.com/p/google-code-prettify/

JUnit (junit-4.10) is licensed under the Common Public License v. 1.0
See http://junit.sourceforge.net/cpl-v10.html

This product includes code (JaspellTernarySearchTrie) from Java Spelling Checkin
g Package (jaspell): http://jaspell.sourceforge.net/

License: The BSD License (http://www.opensource.org/licenses/bsd-license.php)

The snowball stemmers in
 analysis/common/src/java/net/sf/snowball
were developed by Martin Porter and Richard Boulton.
The snowball stopword lists in
 analysis/common/src/resources/org/apache/lucene/analysis/snowball
were developed by Martin Porter and Richard Boulton.
The full snowball package is available from
 http://snowball.tartarus.org/

The KStem stemmer in
 analysis/common/src/org/apache/lucene/analysis/en
was developed by Bob Krovetz and Sergio Guzman-Lara (CIIR-UMass Amherst)
under the BSD-license.

The Arabic,Persian,Romanian,Bulgarian, and Hindi analyzers (common) come with a default
stopword list that is BSD-licensed created by Jacques Savoy. These files reside in:
analysis/common/src/resources/org/apache/lucene/analysis/ar/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/fa/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/ro/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/bg/stopwords.txt,
analysis/common/src/resources/org/apache/lucene/analysis/hi/stopwords.txt
See http://members.unine.ch/jacques.savoy/clef/index.html.

The German,Spanish,Finnish,French,Hungarian,Italian,Portuguese,Russian and Swedish light stemmers
(common) are based on BSD-licensed reference implementations created by Jacques Savoy and
Ljiljana Dolamic. These files reside in:
analysis/common/src/java/org/apache/lucene/analysis/de/GermanLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/de/GermanMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/es/SpanishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fi/FinnishLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/fr/FrenchMinimalStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/hu/HungarianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/it/ItalianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/pt/PortugueseLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/ru/RussianLightStemmer.java
analysis/common/src/java/org/apache/lucene/analysis/sv/SwedishLightStemmer.java

The Stempel analyzer (stempel) includes BSD-licensed software developed
by the Egothor project http://egothor.sf.net/, created by Leo Galambos, Martin Kvapil,
and Edmond Nolan.

The Polish analyzer (stempel) comes with a default
stopword list that is BSD-licensed created by the Carrot2 project. The file resides
in stempel/src/resources/org/apache/lucene/analysis/pl/stopwords.txt.
See http://project.carrot2.org/license.html.

The SmartChineseAnalyzer source code (smartcn) was
provided by Xiaoping Gao and copyright 2009 by www.imdict.net.

WordBreakTestUnicode_*.java (under modules/analysis/common/src/test/)
is derived from Unicode data such as the Unicode Character Database.
See http://unicode.org/copyright.html for more details.

The Morfologik analyzer (morfologik) includes BSD-licensed software
developed by Dawid Weiss and Marcin Mikowski (http://morfologik.blogspot.com/).

Morfologik uses data from Polish ispell/myspell dictionary
(http://www.sjp.pl/slownik/en/) licenced on the terms of (inter alia)
LGPL and Creative Commons ShareAlike.

Morfologic includes data from BSD-licensed dictionary of Polish (SGJP)
(http://sgjp.pl/morfeusz/)

Servlet-api.jar and javax.servlet-*.jar are under the CDDL license, the original
source code for this can be found at http://www.eclipse.org/jetty/downloads.php

===
Kuromoji Japanese Morphological Analyzer - Apache Lucene Integration
===

This software includes a binary and/or source version of data from

 mecab-ipadic-2.7.0-20070801

which can be obtained from

 http://atilika.com/releases/mecab-ipadic/mecab-ipadic-2.7.0-20070801.tar.gz

or

 http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz

===
mecab-ipadic-2.7.0-20070801 Notice
===

Nara Institute of Science and Technology (NAIST),
the copyright holders, disclaims all warranties with regard to this
software, including all implied warranties of merchantability and
fitness, in no event shall NAIST be liable for
any special, indirect or consequential damages or any damages
whatsoever resulting from loss of use, data or profits, whether in an
action of contract, negligence or other tortuous action, arising out

of or in connection with the use or performance of this software.

A large portion of the dictionary entries
originate from ICOT Free Software. The following conditions for ICOT
Free Software applies to the current dictionary as well.

Each User may also freely distribute the Program, whether in its
original form or modified, to any third party or parties, PROVIDED
that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
on, or be attached to, the Program, which is distributed substantially
in the same form as set out herein and that such intended
distribution, if actually made, will neither violate or otherwise
contravene any of the laws and regulations of the countries having
jurisdiction over the User or the intended distribution itself.

NO WARRANTY

The program was produced on an experimental basis in the course of the
research and development conducted during the project and is provided
to users as so produced on an experimental basis. Accordingly, the
program is provided without any warranty whatsoever, whether express,
implied, statutory or otherwise. The term "warranty" used herein
includes, but is not limited to, any warranty of the quality,
performance, merchantability and fitness for a particular purpose of
the program and the nonexistence of any infringement or violation of
any right of any third party.

Each user of the program will agree and understand, and be deemed to
have agreed and understood, that there is no warranty whatsoever for
the program and, accordingly, the entire risk arising from or
otherwise connected with the program is assumed by the user.

Therefore, neither ICOT, the copyright holder, or any other
organization that participated in or was otherwise related to the
development of the program and their respective officials, directors,
officers and other employees shall be held liable for any and all
damages, including, without limitation, general, special, incidental
and consequential damages, arising out of or otherwise in connection
with the use or inability to use the program or any product, material
or result produced or otherwise obtained by using the program,
regardless of whether they have been advised of, or otherwise had
knowledge of, the possibility of such damages at any time during the
project or thereafter. Each user will be deemed to have agreed to the
foregoing by his or her commencement of use of the program. The term
"use" as used herein includes, but is not limited to, the use,
modification, copying and distribution of the program and the
production of secondary products from the program.

In the case where the program, whether in its original form or
modified, was distributed or delivered to or received by a user from
any person, organization or entity other than ICOT, unless it makes or
grants independently of ICOT any specific warranty to the user in
writing, such person, organization or entity, will also be exempted
from and not be held liable to the user for any such damages as noted
above as far as the program is concerned.

===
lucene LICENSE
===

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work

or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work

by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from unicode conversion examples available at
http://www.unicode.org/Public/PROGRAMS/CVTUTF. Here is the copyright
from those sources:

Some code in core/src/java/org/apache/lucene/util/ArrayUtil.java was
derived from Python 2.4.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/2.4.2/license/

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from Python 3.1.2 sources available at
http://www.python.org. Full license is here:

 http://www.python.org/download/releases/3.1.2/license/

Some code in core/src/java/org/apache/lucene/util/automaton was
derived from Brics automaton sources available at
www.brics.dk/automaton/. Here is the copyright from those sources:

The levenshtein automata tables in core/src/java/org/apache/lucene/util/automaton
were automatically generated with the moman/finenight FSA package.
Here is the copyright for those sources:

Copyright (c) 2010, Jean-Philippe Barrette-LaPierre, <jpb@rrette.com>
#
Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:
#
The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.
#
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

Some code in core/src/java/org/apache/lucene/util/UnicodeUtil.java was
derived from ICU (http://www.icu-project.org)
The full license is available here:
 http://source.icu-project.org/repos/icu/icu/trunk/license.html

/*
* Copyright (C) 1999-2010, International Business Machines
* Corporation and others. All Rights Reserved.
*
* Permission is hereby granted, free of charge, to any person obtaining a copy
* of this software and associated documentation files (the "Software"), to deal
* in the Software without restriction, including without limitation the rights
* to use, copy, modify, merge, publish, distribute, and/or sell copies of the
* Software, and to permit persons to whom the Software is furnished to do so,
* provided that the above copyright notice(s) and this permission notice appear
* in all copies of the Software and that both the above copyright notice(s) and
* this permission notice appear in supporting documentation.
*
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
* IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
* FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. The names "Center for Intelligent Information Retrieval" and
"University of Massachusetts" must not be used to endorse or promote products
derived from this software without prior written permission. To obtain
permission, contact info@ciir.cs.umass.edu.

THIS SOFTWARE IS PROVIDED BY UNIVERSITY OF MASSACHUSETTS AND OTHER CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The following license applies to the Morfologik project:

Copyright (c) 2006 Dawid Weiss
Copyright (c) 2007-2011 Dawid Weiss, Marcin Mikowski
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

  * Neither the name of Morfologik nor the names of its contributors
  may be used to endorse or promote products derived from this software
  without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The dictionary comes from Morfologik project. Morfologik uses data from Polish ispell/myspell dictionary hosted at http://www.sjp.pl/slownik/en/ and is licenced on the terms of (inter alia) LGPL and Creative Commons ShareAlike. The part-of-speech tags were added in Morfologik project and are not found in the data from sjp.pl. The tagset is similar to IPI PAN tagset.

The following license applies to the Morfeusz project, used by org.apache.lucene.analysis.morfologik.

BSD-licensed dictionary of Polish (SGJP)
http://sgjp.pl/morfeusz/

Copyright 2011 Zygmunt Saloni, Wodzimierz Gruszczyski,
        Marcin Woliski, Robert Woosz

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the
   distribution.

THIS SOFTWARE IS PROVIDED BY COPYRIGHT HOLDERS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL COPYRIGHT HOLDERS OR CONTRIBUTORS BE

LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==
netty NOTICE
==

                    The Netty Project
                    ==================

Please visit the Netty web site for more information:

 * http://netty.io/

Copyright 2011 The Netty Project

The Netty Project licenses this file to you under the Apache License,
version 2.0 (the "License"); you may not use this file except in compliance
with the License. You may obtain a copy of the License at:

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS, WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the
License for the specific language governing permissions and limitations
under the License.

Also, please refer to each LICENSE.<component>.txt file, which is located in
the 'license' directory of the distribution file, for the license terms of the
components that this product depends on.

This product contains the extensions to Java Collections Framework which has
been derived from the works by JSR-166 EG, Doug Lea, and Jason T. Greene:

 * LICENSE:
   * license/LICENSE.jsr166y.txt (Public Domain)
 * HOMEPAGE:
   * http://gee.cs.oswego.edu/cgi-bin/viewcvs.cgi/jsr166/
   * http://viewvc.jboss.org/cgi-bin/viewvc.cgi/jbosscache/experimental/jsr166/

This product contains a modified version of Robert Harder's Public Domain
Base64 Encoder and Decoder, which can be obtained at:

 * LICENSE:
   * license/LICENSE.base64.txt (Public Domain)
 * HOMEPAGE:
   * http://iharder.sourceforge.net/current/java/base64/

This product contains a modified version of 'JZlib', a re-implementation of
zlib in pure Java, which can be obtained at:

 * LICENSE:
   * license/LICENSE.jzlib.txt (BSD Style License)
 * HOMEPAGE:
   * http://www.jcraft.com/jzlib/

This product contains a modified version of 'Webbit', a Java event based
WebSocket and HTTP server:

 * LICENSE:
   * license/LICENSE.webbit.txt (BSD License)
 * HOMEPAGE:
   * https://github.com/joewalnes/webbit

This product optionally depends on 'Protocol Buffers', Google's data
interchange format, which can be obtained at:

 * LICENSE:
   * license/LICENSE.protobuf.txt (New BSD License)
 * HOMEPAGE:
   * http://code.google.com/p/protobuf/

This product optionally depends on 'Bouncy Castle Crypto APIs' to generate
a temporary self-signed X.509 certificate when the JVM does not provide the
equivalent functionality. It can be obtained at:

 * LICENSE:
   * license/LICENSE.bouncycastle.txt (MIT License)
 * HOMEPAGE:
   * http://www.bouncycastle.org/

This product optionally depends on 'SLF4J', a simple logging facade for Java,
which can be obtained at:

 * LICENSE:
   * license/LICENSE.slf4j.txt (MIT License)
 * HOMEPAGE:
   * http://www.slf4j.org/

This product optionally depends on 'Apache Commons Logging', a logging
framework, which can be obtained at:

 * LICENSE:
   * license/LICENSE.commons-logging.txt (Apache License 2.0)
 * HOMEPAGE:
   * http://commons.apache.org/logging/

This product optionally depends on 'Apache Log4J', a logging framework,
which can be obtained at:

 * LICENSE:
   * license/LICENSE.log4j.txt (Apache License 2.0)
 * HOMEPAGE:
   * http://logging.apache.org/log4j/

This product optionally depends on 'JBoss Logging', a logging framework,
which can be obtained at:

 * LICENSE:
   * license/LICENSE.jboss-logging.txt (GNU LGPL 2.1)
 * HOMEPAGE:
   * http://anonsvn.jboss.org/repos/common/common-logging-spi/

This product optionally depends on 'Apache Felix', an open source OSGi
framework implementation, which can be obtained at:

 * LICENSE:
   * license/LICENSE.felix.txt (Apache License 2.0)
 * HOMEPAGE:
   * http://felix.apache.org/

==
netty LICENSE
==

                    Apache License
                 Version 2.0, January 2004
               http://www.apache.org/licenses/

  TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

  1. Definitions.

     "License" shall mean the terms and conditions for use, reproduction,
     and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its

distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,

unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

```
================================================================================
securesm LICENSE
================================================================================
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or
    Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
    stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
    that You distribute, all copyright, patent, trademark, and
    attribution notices from the Source form of the Work,
    excluding those notices that do not pertain to any part of
    the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
    distribution, then any Derivative Works that You distribute must
    include a readable copy of the attribution notices contained
    within such NOTICE file, excluding those notices that do not
    pertain to any part of the Derivative Works, in at least one
    of the following places: within a NOTICE text file distributed
    as part of the Derivative Works; within the Source form or
    documentation, if provided along with the Derivative Works; or,
    within a display generated by the Derivative Works, if and
    wherever such third-party notices normally appear. The contents
    of the NOTICE file are for informational purposes only and
    do not modify the License. You may add Your own attribution
    notices within Derivative Works that You distribute, alongside
    or as an addendum to the NOTICE text from the Work, provided
    that such additional attribution notices cannot be construed
    as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,

except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
snakeyaml NOTICE
==
The art of simplicity is a puzzle of complexity.

Overview
[YAML](http://yaml.org) is a data serialization format designed for human readability and interaction with scripting languages.

SnakeYAML is a YAML processor for the Java Virtual Machine.

SnakeYAML features

* a **complete** [YAML 1.1 processor](http://yaml.org/spec/1.1/current.html). In particular, SnakeYAML can parse all examples from the specification.
* Unicode support including UTF-8/UTF-16 input/output.
* high-level API for serializing and deserializing native Java objects.
* support for all types from the [YAML types repository](http://yaml.org/type/index.html).
* relatively sensible error messages.

Info
* [Changes](https://bitbucket.org/asomov/snakeyaml/wiki/Changes)
* [Documentation](https://bitbucket.org/asomov/snakeyaml/wiki/Documentation)

Contribute
* Mercurial DVCS is used to dance with the [source code](https://bitbucket.org/asomov/snakeyaml/src).
* If you find a bug in SnakeYAML, please [file a bug
report](https://bitbucket.org/asomov/snakeyaml/issues?status=new&status=open).
* You may discuss SnakeYAML at
[the mailing list](http://groups.google.com/group/snakeyaml-core).

==

snakeyaml LICENSE

===

                            Apache License
                       Version 2.0, January 2004
                     http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

      "Work" shall mean the work of authorship, whether in Source or
      Object form, made available under the License, as indicated by a
      copyright notice that is included in or attached to the work
      (an example is provided in the Appendix below).

      "Derivative Works" shall mean any work, whether in Source or Object
      form, that is based on (or derived from) the Work and for which the
      editorial revisions, annotations, elaborations, or other modifications
      represent, as a whole, an original work of authorship. For the purposes
      of this License, Derivative Works shall not include works that remain
      separable from, or merely link (or bind by name) to the interfaces of,
      the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or

Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

   END OF TERMS AND CONDITIONS

==
spatial4j NOTICE
==
Eclipse Foundation Software User Agreement

April 9, 2014

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION
AND/OR OTHER MATERIALS FOR OPEN SOURCE
PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND

CONDITIONS OF THIS AGREEMENT AND/OR
THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR
REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE
THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND
CONDITIONS OF ANY APPLICABLE LICENSE
AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE
TERMS AND CONDITIONS OF THIS AGREEMENT
AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES
INDICATED OR REFERENCED BELOW, THEN YOU MAY
NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and
conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is
also available at http://www.eclipse.org/legal/epl-v10.html. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the
Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made available as
downloadable archives ("Downloads").

*   Content may be structured and packaged into modules to facilitate delivering, extending, and upgrading the Content.
    Typical modules may include plug-ins ("Plug-ins"), plug-in fragments ("Fragments"), and features ("Features").
*   Each Plug-in or Fragment may be packaged as a sub-directory or JAR (Java ARchive) in a directory named "plugins".
*   A Feature is a bundle of one or more Plug-ins and/or Fragments and associated material. Each Feature may be packaged
    as a sub-directory in a directory named "features". Within a Feature, files named "feature.xml" may contain a list
    of the names and version numbers of the Plug-ins and/or Fragments associated with that Feature.
*   Features may also include other Features ("Included Features"). Within a Feature, files named "feature.xml" may
    contain a list of the names and version numbers of Included Features.

The terms and conditions governing Plug-ins and Fragments should be contained in files named "about.html" ("Abouts").
The terms and conditions governing Features and Included Features should be contained in files named "license.html"
("Feature Licenses"). Abouts and Feature Licenses may be located in any directory of a Download or Module including, but
not limited to the following locations:

*   The top-level (root) directory
*   Plug-in and Fragment directories
*   Inside Plug-ins and Fragments packaged as JARs
*   Sub-directories of the directory named "src" of certain Plug-ins

*   Feature directories

Note: if a Feature made available by the Eclipse Foundation is installed using the Provisioning Technology (as defined
below), you must agree to a license ("Feature Update License") during the installation process. If the Feature contains
Included Features, the Feature Update License should either provide you with the terms and conditions governing the
Included Features or inform you where you can locate them. Feature Update Licenses may be found in the "license"
property of files named "feature.properties" found within a Feature. Such Abouts, Feature Licenses, and Feature Update
Licenses contain the terms and conditions (or references to such terms and conditions) that govern your use of the
associated Content in that directory.

THE ABOUTS, FEATURE LICENSES, AND FEATURE UPDATE LICENSES MAY REFER TO THE EPL OR OTHER LICENSE AGREEMENTS, NOTICES OR
TERMS AND CONDITIONS. SOME OF THESE OTHER LICENSE AGREEMENTS MAY INCLUDE (BUT ARE NOT LIMITED TO):

*   Eclipse Distribution License Version 1.0 (available at http://www.eclipse.org/licenses/edl-v10.html)
*   Common Public License Version 1.0 (available at http://www.eclipse.org/legal/cpl-v10.html)
*   Apache Software License 1.1 (available at http://www.apache.org/licenses/LICENSE)
*   Apache Software License 2.0 (available at http://www.apache.org/licenses/LICENSE-2.0)
*   Mozilla Public License Version 1.1 (available at http://www.mozilla.org/MPL/MPL-1.1.html)

IT IS YOUR OBLIGATION TO READ AND ACCEPT ALL SUCH TERMS AND CONDITIONS PRIOR TO USE OF THE CONTENT. If no About, Feature
License, or Feature Update License is provided, please contact the Eclipse Foundation to determine what terms and
conditions govern that particular Content.

Use of Provisioning Technology

The Eclipse Foundation makes available provisioning software, examples of which include, but are not limited to, p2 and
the Eclipse Update Manager ("Provisioning Technology") for the purpose of allowing users to install software,
documentation, information and/or other materials (collectively "Installable Software"). This capability is provided
with the intent of allowing such users to install, extend and update Eclipse-based products. Information about packaging
Installable Software is available at http://eclipse.org/equinox/p2/repository_packaging.html ("Specification").

You may use Provisioning Technology to allow other parties to install Installable Software. You shall be responsible for
enabling the applicable license agreements relating to the Installable Software to be presented to, and accepted by, the
users of the Provisioning Technology in accordance with the Specification. By using Provisioning Technology in such a
manner and making it available in accordance with the Specification, you further acknowledge your agreement to, and the

acquisition of all necessary rights to permit the following:

1. A series of actions may occur ("Provisioning Process") in which a user may execute the Provisioning Technology
on a
   machine ("Target Machine") with the intent of installing, extending or updating the functionality of an
   Eclipse-based product.
2. During the Provisioning Process, the Provisioning Technology may cause third party Installable Software or a
portion
   thereof to be accessed and copied to the Target Machine.
3. Pursuant to the Specification, you will provide to the user the terms and conditions that govern the use of the
   Installable Software ("Installable Software Agreement") and such Installable Software Agreement shall be
accessed
   from the Target Machine in accordance with the Specification. Such Installable Software Agreement must inform
the
   user of the terms and conditions that govern the Installable Software and must solicit acceptance by the end user in
   the manner prescribed in such Installable Software Agreement. Upon such indication of agreement by the user, the
   provisioning Technology will complete installation of the Installable Software.

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the
import,
possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption
software,
please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of
encryption software, to see if this is permitted.

Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries,
or both.

==
spatial4j LICENSE
==

                         Apache License
                    Version 2.0, January 2004
                  http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity

on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

  (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

  (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

  (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

  (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one

of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a

result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

==
t-digest NOTICE
==

The code for the t-digest was originally authored by Ted Dunning

A number of small but very helpful changes have been contributed by Adrien Grand (https://github.com/jpountz)

==
t-digest LICENSE
==
Apache License
                Version 2.0, January 2004
                http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

     "License" shall mean the terms and conditions for use, reproduction,
     and distribution as defined by Sections 1 through 9 of this document.

     "Licensor" shall mean the copyright owner or entity authorized by
     the copyright owner that is granting the License.

     "Legal Entity" shall mean the union of the acting entity and all
     other entities that control, are controlled by, or are under common
     control with that entity. For the purposes of this definition,
     "control" means (i) the power, direct or indirect, to cause the
     direction or management of such entity, whether by contract or
     otherwise, or (ii) ownership of fifty percent (50%) or more of the
     outstanding shares, or (iii) beneficial ownership of such entity.

     "You" (or "Your") shall mean an individual or Legal Entity
     exercising permissions granted by this License.

     "Source" form shall mean the preferred form for making modifications,
     including but not limited to software source code, documentation
     source, and configuration files.

     "Object" form shall mean any form resulting from mechanical
     transformation or translation of a Source form, including but
     not limited to compiled object code, generated documentation,
     and conversions to other media types.

     "Work" shall mean the work of authorship, whether in Source or
     Object form, made available under the License, as indicated by a
     copyright notice that is included in or attached to the work
     (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed

with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "{}"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate

comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

                    Apache License
              Version 2.0, January 2004
            http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

  "License" shall mean the terms and conditions for use, reproduction,
  and distribution as defined by Sections 1 through 9 of this document.

  "Licensor" shall mean the copyright owner or entity authorized by
  the copyright owner that is granting the License.

  "Legal Entity" shall mean the union of the acting entity and all
  other entities that control, are controlled by, or are under common
  control with that entity. For the purposes of this definition,
  "control" means (i) the power, direct or indirect, to cause the
  direction or management of such entity, whether by contract or
  otherwise, or (ii) ownership of fifty percent (50%) or more of the
  outstanding shares, or (iii) beneficial ownership of such entity.

  "You" (or "Your") shall mean an individual or Legal Entity
  exercising permissions granted by this License.

  "Source" form shall mean the preferred form for making modifications,
  including but not limited to software source code, documentation
  source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

   http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

                    GNU LESSER GENERAL PUBLIC LICENSE
                        Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

 This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
License, supplemented by the additional permissions listed below.

 0. Additional Definitions.

 As used herein, "this License" refers to version 3 of the GNU Lesser
General Public License, and the "GNU GPL" refers to version 3 of the GNU
General Public License.

 "The Library" refers to a covered work governed by this License,
other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided
by the Library, but which is not otherwise based on the Library.
Defining a subclass of a class defined by the Library is deemed a mode
of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an
Application with the Library. The particular version of the Library
with which the Combined Work was made is also called the "Linked
Version".

The "Minimal Corresponding Source" for a Combined Work means the
Corresponding Source for the Combined Work, excluding any source code
for portions of the Combined Work that, considered in isolation, are
based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the
object code and/or source code for the Application, including any data
and utility programs needed for reproducing the Combined Work from the
Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License
without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a
facility refers to a function or data to be supplied by an Application
that uses the facility (other than as an argument passed when the
facility is invoked), then you may convey a copy of the modified
version:

   a) under this License, provided that you make a good faith effort to
   ensure that, in the event an Application does not supply the
   function or data, the facility still operates, and performs
   whatever part of its purpose remains meaningful, or

   b) under the GNU GPL, with none of the additional permissions of
   this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from
a header file that is part of the Library. You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure

layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

  a) Give prominent notice with each copy of the object code that the
  Library is used in it and that the Library and its use are
  covered by this License.

  b) Accompany the object code with a copy of the GNU GPL and this license
  document.

 4. Combined Works.

 You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

  a) Give prominent notice with each copy of the Combined Work that
  the Library is used in it and that the Library and its use are
  covered by this License.

  b) Accompany the Combined Work with a copy of the GNU GPL and this license
  document.

  c) For a Combined Work that displays copyright notices during
  execution, include the copyright notice for the Library among
  these notices, as well as a reference directing the user to the
  copies of the GNU GPL and this license document.

  d) Do one of the following:

    0) Convey the Minimal Corresponding Source under the terms of this
    License, and the Corresponding Application Code in a form
    suitable for, and under terms that permit, the user to
    recombine or relink the Application with a modified version of
    the Linked Version to produce a modified Combined Work, in the
    manner specified by section 6 of the GNU GPL for conveying
    Corresponding Source.

    1) Use a suitable shared library mechanism for linking with the
    Library. A suitable mechanism is one that (a) uses at run time
    a copy of the Library already present on the user's computer
    system, and (b) will operate properly with a modified version
    of the Library that is interface-compatible with the Linked
    Version.

  e) Provide Installation Information, but only if you would otherwise

be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the
Library side by side in a single library together with other library
facilities that are not Applications and are not covered by this
License, and convey such a combined library under terms of your
choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based
on the Library, uncombined with any other library facilities,
conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it
is a work based on the Library, and explaining where to find the
accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions
of the GNU Lesser General Public License from time to time. Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the
Library as you received it specifies that a certain numbered version
of the GNU Lesser General Public License "or any later version"
applies to it, you have the option of following the terms and
conditions either of that published version or of any later version
published by the Free Software Foundation. If the Library as you
received it does not specify a version number of the GNU Lesser
General Public License, you may choose any version of the GNU Lesser
General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide
whether future versions of the GNU Lesser General Public License shall
apply, that proxy's public statement of acceptance of any version is
permanent authorization for you to choose that version for the

Library.
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "mybatis-3-mapper.dtd">

<mapper namespace="org.sonar.db.permission.UserPermissionMapper">

 <select id="selectUserPermissionsByQueryAndUserIds" parameterType="map"
resultType="org.sonar.db.permission.UserPermissionDto">
   select
     u.id as userId,
     ur.organization_uuid as organizationUuid,
     ur.resource_id as componentId,
     ur.role as permission
   <include refid="sqlQueryJoins" />
   <where>
     u.id in <foreach collection="userIds" open="(" close=")" item="userId"
separator=",">#{userId,jdbcType=INTEGER}</foreach>
     <include refid="sqlQueryFilters" />
   </where>
 </select>

 <select id="selectUserIdsByQuery" parameterType="map" resultType="int">
   select
     distinct u.id, lower(u.name) as lowerName
     <include refid="sqlQueryJoins" />
     <where>
       <include refid="sqlQueryFilters" />
     </where>
   order by lowerName asc
 </select>

 <select id="countUsersByQuery" parameterType="map" resultType="int">
   select count(distinct(u.id))
   <include refid="sqlQueryJoins" />
   <where>
     <include refid="sqlQueryFilters" />
   </where>
 </select>

 <sql id="sqlQueryJoins">
   from users u
   left join user_roles ur on ur.user_id = u.id
   left join projects p on ur.resource_id = p.id
   inner join organization_members om on u.id=om.user_id and
om.organization_uuid=#{query.organizationUuid,jdbcType=VARCHAR}
 </sql>

 <sql id="sqlQueryFilters">
```

```
      and u.active = ${_true}
  <if test="query.searchQueryToSql != null">
    and (
    lower(u.name) like #{query.searchQueryToSqlLowercase,jdbcType=VARCHAR} ESCAPE '/'
    or u.email like #{query.searchQueryToSql,jdbcType=VARCHAR} ESCAPE '/'
    or u.login like #{query.searchQueryToSql,jdbcType=VARCHAR} ESCAPE '/')
  </if>
  <!-- filter rows with user permissions -->
  <if test="query.withAtLeastOnePermission()">
    and ur.organization_uuid = om.organization_uuid
    and ur.role is not null
    <if test="query.componentUuid==null">
      and ur.resource_id is null
    </if>
    <if test="query.componentUuid!=null">
      and p.uuid = #{query.componentUuid,jdbcType=VARCHAR}
    </if>
    <if test="query.permission!=null">
      and ur.role = #{query.permission,jdbcType=VARCHAR}
    </if>
  </if>
</sql>

<select id="selectGlobalPermissionsOfUser" parameterType="map" resultType="string">
  select ur.role
  from user_roles ur
  where
  ur.organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
  ur.user_id = #{userId,jdbcType=INTEGER} and
  ur.resource_id is null
</select>

<select id="selectProjectPermissionsOfUser" parameterType="map" resultType="string">
  select ur.role
  from user_roles ur
  where
  ur.user_id = #{userId,jdbcType=INTEGER} and
  ur.resource_id = #{projectId,jdbcType=BIGINT}
</select>

<select id="countUsersByProjectPermission" resultType="org.sonar.db.permission.CountPerProjectPermission">
  select ur.resource_id as componentId, ur.role as permission, count(u.login) as count
  from users u
  inner join user_roles ur on ur.user_id = u.id
  inner join projects p on p.id = ur.resource_id
  where u.active = ${_true}
  and p.id in <foreach collection="projectIds" open="(" close=")" item="projectId"
separator=",">#{projectId}</foreach>
```

```xml
    group by ur.resource_id, ur.role
  </select>

  <select id="selectUserIdsWithPermissionOnProjectBut" resultType="Integer">
    select
      distinct ur1.user_id
    from
      user_roles ur1
    where
      ur1.resource_id = #{projectId,jdbcType=BIGINT}
      and role <> #{permission,jdbcType=VARCHAR}
      and not exists (
        select
          1
        from
          user_roles ur2
        where
          ur2.resource_id = ur1.resource_id
          and ur2.user_id = ur1.user_id
          and role = #{permission,jdbcType=VARCHAR}
      )
  </select>

  <insert id="insert" parameterType="org.sonar.db.permission.UserPermissionDto" useGeneratedKeys="false">
    insert into user_roles (
    organization_uuid,
    user_id,
    resource_id,
    role
    ) values (
    #{organizationUuid,jdbcType=VARCHAR},
    #{userId,jdbcType=INTEGER},
    #{componentId,jdbcType=BIGINT},
    #{permission,jdbcType=VARCHAR}
    )
  </insert>

  <delete id="deleteGlobalPermission" parameterType="map">
    delete from user_roles
    where
    role = #{permission,jdbcType=VARCHAR} and
    user_id = #{userId,jdbcType=INTEGER} and
    organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
    resource_id is null
  </delete>

  <delete id="deleteProjectPermission" parameterType="map">
    delete from user_roles
```

```xml
        where
          role = #{permission,jdbcType=VARCHAR} and
          user_id = #{userId,jdbcType=INTEGER} and
          resource_id = #{projectId,jdbcType=BIGINT}
        </delete>

        <delete id="deleteProjectPermissions" parameterType="map">
          delete from user_roles
          where
          resource_id = #{projectId,jdbcType=BIGINT}
        </delete>

        <delete id="deleteProjectPermissionOfAnyUser" parameterType="map">
          delete from
            user_roles
          where
            resource_id = #{projectId,jdbcType=BIGINT}
            and role = #{permission,jdbcType=VARCHAR}
        </delete>

        <delete id="deleteByOrganization" parameterType="String">
          delete from
            user_roles
          where
            organization_uuid = #{organizationUuid,jdbcType=VARCHAR}
        </delete>

        <delete id="deleteOrganizationMemberPermissions" parameterType="map">
          delete from
            user_roles
          where
            organization_uuid = #{organizationUuid,jdbcType=VARCHAR} and
            user_id = #{userId,jdbcType=INTEGER}
        </delete>

        <delete id="deleteByUserId" parameterType="int">
          DELETE FROM user_roles WHERE user_id=#{userId,jdbcType=INTEGER}
        </delete>
      </mapper>
```

1.131 summernote 0.8.10

1.131.1 Available under license :

The MIT License (MIT)

Copyright (c) 2015~ Summernote Team (https://github.com/orgs/summernote/people)

1.132 swagger-core 1.5.19

1.132.1 Available under license :

```
/**
* Copyright 2017 SmartBear Software
* <p>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p>
* http://www.apache.org/licenses/LICENSE-2.0
* <p>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package io.swagger.v3.oas.models.info;

import java.util.Objects;

/**
* License
*
* @see "https://github.com/OAI/OpenAPI-Specification/blob/3.0.1/versions/3.0.1.md#licenseObject"
*/
```

```java
public class License {
  private String name = null;
  private String url = null;
  private java.util.Map<String, Object> extensions = null;

  /**
   * returns the name property from a License instance.
   *
   * @return String name
   **/

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }

  public License name(String name) {
    this.name = name;
    return this;
  }

  /**
   * returns the url property from a License instance.
   *
   * @return String url
   **/

  public String getUrl() {
    return url;
  }

  public void setUrl(String url) {
    this.url = url;
  }

  public License url(String url) {
    this.url = url;
    return this;
  }

  @Override
  public boolean equals(java.lang.Object o) {
    if (this == o) {
      return true;
    }
```

```java
        if (o == null || getClass() != o.getClass()) {
            return false;
        }
        License license = (License) o;
        return Objects.equals(this.name, license.name) &&
                Objects.equals(this.url, license.url) &&
                Objects.equals(this.extensions, license.extensions);
    }

    @Override
    public int hashCode() {
        return Objects.hash(name, url, extensions);
    }

    public java.util.Map<String, Object> getExtensions() {
        return extensions;
    }

    public void addExtension(String name, Object value) {
        if (name == null || name.isEmpty() || !name.startsWith("x-")) {
            return;
        }
        if (this.extensions == null) {
            this.extensions = new java.util.HashMap<>();
        }
        this.extensions.put(name, value);
    }

    public void setExtensions(java.util.Map<String, Object> extensions) {
        this.extensions = extensions;
    }

    public License extensions(java.util.Map<String, Object> extensions) {
        this.extensions = extensions;
        return this;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("class License {\n");

        sb.append("    name: ").append(toIndentedString(name)).append("\n");
        sb.append("    url: ").append(toIndentedString(url)).append("\n");
        sb.append("}");
        return sb.toString();
    }
```

```java
    /**
     * Convert the given object to string with each line indented by 4 spaces
     * (except the first line).
     */
    private String toIndentedString(java.lang.Object o) {
        if (o == null) {
            return "null";
        }
        return o.toString().replace("\n", "\n    ");
    }


}
/**
* Copyright 2017 SmartBear Software
* <p>
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* <p>
* http://www.apache.org/licenses/LICENSE-2.0
* <p>
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

package io.swagger.v3.oas.annotations.info;

import io.swagger.v3.oas.annotations.extensions.Extension;

import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

/**
* The annotation may be used in {@link Info#license()} to define a license for the OpenAPI spec.
*
* @see <a target="_new" href="https://github.com/OAI/OpenAPI-
Specification/blob/3.0.1/versions/3.0.1.md#licenseObject">License (OpenAPI specification)</a>
* @see io.swagger.v3.oas.annotations.OpenAPIDefinition
* @see Info
**/
@Target({})
@Retention(RetentionPolicy.RUNTIME)
public @interface License {
    /**
```

```
   * The license name used for the API.
   *
   * @return the name of the license
   **/
   String name() default "";

   /**
   * A URL to the license used for the API. MUST be in the format of a URL.
   *
   * @return the URL of the license
   **/
   String url() default "";

   /**
   * The list of optional extensions
   *
   * @return an optional array of extensions
   */
   Extension[] extensions() default {};


}
```
Copyright 2018 SmartBear Software

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at [apache.org/licenses/LICENSE-
2.0](http://www.apache.org/licenses/LICENSE-2.0)

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.133 swagger-jaxrs2 2.0.7

1.134 swagger-parser 1.0.35
1.134.1 Available under license :

Copyright 2017 SmartBear Software

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at [apache.org/licenses/LICENSE-

1.135 webpack-merge 4.1.2

1.135.1 Available under license :

Contributors

* [Fernando Montoya](https://github.com/montogeek) - Use separate lodash functions instead of the core package. Faster to install this way.
* [Jonathan Felchlin](https://github.com/GreenGremlin) - Smart merging for loaders.
* [David Gmez](https://github.com/davegomez) - Performance and cosmetic improvements.
* [siready](https://github.com/siready) - Extend `merge.smart` to support `include/exclude`.
* [C.J. Winslow](https://github.com/Whoaa512) - Make `merge.smart` `include/exclude` to work correctly with `loader`.
* [Artem Zakharchenko](https://github.com/blackrabbit99) - Fix `merge.smart` duplication so that if `include` exists, it will merge.
* [Matt Shwery](https://github.com/mshwery) - If `exclude` is the same while using `merge.smart`, merge `loaders`.
* [Lucretiel](https://github.com/Lucretiel) - Added a more generic test to describe merge behavior better.
* [Christian Hoffmeister](https://github.com/choffmeister) - Fix `merge.smart` behavior so that it checks against full loader names instead of just the first letter.
* [Ken Powers](https://github.com/knpwrs) - Changed Travis icon to use SVG (scales better).
* [Kyle Herock](https://github.com/rockmacaca) - Improved webpack 2 support, avoided concatenating loaders if the first matching entry's include/exclude doesn't match. #41
* [Steven Haddix](https://github.com/steven-haddix) - Clarify description. #42
* [Artem Sapegin](https://github.com/sapegin) - Tweaked customizeArray/customizeObject example. #60
* [Dan Kirkham](https://github.com/herecydev) - `merge.multiple`. #74
* [Ahmed Elmehri](https://github.com/ahmehri) - Add `customizeArray` and `customizeObject` examples. #93
* [Flvio](https://github.com/flaviorocks) - Maintenance - Update lodash, #97, #98
* [Amy Lynn](https://github.com/Amy-Lynn) - Smart merge should maintain existing loader order. #79, #101
Copyright (c) 2015 Juho Vepsalainen

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.136 yup 0.25.1

1.136.1 Available under license :

The MIT License (MIT)

Copyright (c) 2014 Jason Quense

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.