CheckStyle plugin

Used to detect the coding standard

Two types of Check/Style/Rules:

1. Google\_check
2. Sun\_check

Goals:

* [checkstyle:checkstyle](https://maven.apache.org/plugins/maven-checkstyle-plugin/checkstyle-mojo.html) is a reporting goal that performs Checkstyle analysis and generates a report on violations.
* [checkstyle:checkstyle-aggregate](https://maven.apache.org/plugins/maven-checkstyle-plugin/checkstyle-aggregate-mojo.html) is a reporting goal that performs Checkstyle analysis and generates an aggregate HTML report on violations in a multi-module reactor build.
* [checkstyle:check](https://maven.apache.org/plugins/maven-checkstyle-plugin/check-mojo.html) is a goal that performs Checkstyle analysis and outputs violations or a count of violations to the console, potentially failing the build. It can also be configured to re-use an earlier analysis.

To configure in maven,

1. In pom.xml, Under build -> plugins, enter the below plugin,

<**plugin**>

<**groupId**>org.apache.maven.plugins</**groupId**>

<**artifactId**>maven-checkstyle-plugin</**artifactId**>

<**version**>3.0.0</**version**>

<**configuration**>

<**configLocation**>google\_checks.xml</**configLocation**>

<**encoding**>UTF-8</**encoding**>

<**consoleOutput**>true</**consoleOutput**>

<**failsOnError**>true</**failsOnError**>

<**linkXRef**>false</**linkXRef**>

</**configuration**>

<**executions**>

<**execution**>

<**goals**>

<**goal**>check</**goal**><!-- any goal you wish -->

</**goals**>

</**execution**>

</**executions**>

</**plugin**>

2. Under build -> reporting -> plugins, enter the below plugin,

<**plugin**>

<**groupId**>org.apache.maven.plugins</**groupId**>

<**artifactId**>maven-checkstyle-plugin</**artifactId**>

<**version**>3.0.0</**version**>

<**configuration**>

<**configLocation**>google\_checks.xml</**configLocation**>

</**configuration**>

<**reportSets**>

<**reportSet**>

<**reports**>

<**report**>checkstyle</**report**>

</**reports**>

</**reportSet**>

</**reportSets**>

</**plugin**>

Executing using Maven:

1. In IntelliJ,
2. navigate to Maven Projects panel
3. Under Plugins -> Checkstyle, double click any goal which you want to be executed.
4. This shows errors/output in the below Run panel

1. Using site plugin,
   1. navigate to Maven Projects panel
   2. Under Plugins -> site, double click any goal which you want to be executed. For now, I prefer site:site to generate report
   3. Under project’s target folder, open site folder
   4. Open checkstyle.html to see the report

PMD plugin

Used to find flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth and COPY/PASTE detector(code duplicates)

Goals:

* pmd:check Fail the build if there were any PMD violations in the source code.
* pmd:cpd Creates a report for PMD's CPD tool. See Finding duplicated code for more details.
* pmd:cpd-check Fail the build if there were any CPD violations in the source code.
* pmd:pmd Creates a PMD report.

To configure in maven,

1. In pom.xml, Under build -> plugins, enter the below plugin,

<**plugin**>

<**groupId**>org.apache.maven.plugins</**groupId**>

<**artifactId**>maven-pmd-plugin</**artifactId**>

<**version**>3.11.0</**version**>

<**dependencies**>

<**dependency**>

<**groupId**>net.sourceforge.pmd</**groupId**>

<**artifactId**>pmd-core</**artifactId**>

<**version**>6.10.0</**version**>

</**dependency**>

<**dependency**>

<**groupId**>net.sourceforge.pmd</**groupId**>

<**artifactId**>pmd-java</**artifactId**>

<**version**>${pmdVersion}</**version**>

</**dependency**>

<**dependency**>

<**groupId**>net.sourceforge.pmd</**groupId**>

<**artifactId**>pmd-javascript</**artifactId**>

<**version**>${pmdVersion}</**version**>

</**dependency**>

<**dependency**>

<**groupId**>net.sourceforge.pmd</**groupId**>

<**artifactId**>pmd-jsp</**artifactId**>

<**version**>${pmdVersion}</**version**>

</**dependency**>

</**dependencies**>

<**executions**>

<**execution**>

<**goals**>

<**goal**>check</**goal**>

<**goal**>cpd-check</**goal**>

</**goals**>

</**execution**>

</**executions**>

</**plugin**>

1. Under build -> reporting -> plugins, enter the below plugin,

<**plugin**>

<**groupId**>org.apache.maven.plugins</**groupId**>

<**artifactId**>maven-pmd-plugin</**artifactId**>

<**version**>3.11.0</**version**>

<**reportSets**>

<**reportSet**>

<**reports**>

<**report**>cpd-check</**report**>

</**reports**>

</**reportSet**>

</**reportSets**>

</**plugin**>

Executing using Maven:

1. In IntelliJ,
   1. navigate to Maven Projects panel
   2. Under Plugins -> pmd, double click any goal which you want to be executed.
   3. This shows errors/output in the below Run panel
2. Using site plugin,
   1. navigate to Maven Projects panel
   2. Under Plugins -> pmd, double click any goal which you want to be executed. For now, I prefer site:site to generate report
   3. Under project’s target folder, open site folder
   4. Open cpd.html to see the duplicate error report

SpotBugs plugin

SpotBugs looks for bugs in Java programs. It is based on the concept of bug patterns. A bug pattern is a code idiom that is often an error. Bug patterns arise for a variety of reasons:

* Difficult language features
* Misunderstood API methods
* Misunderstood invariants when code is modified during maintenance
* Garden variety mistakes: typos, use of the wrong boolean operator

Goals:

* spotbugs:check
* spotbugs:gui
* spotbugs:spotbugs

To configure in maven,

1. In pom.xml, Under build -> plugins, enter the below plugin,

<**plugin**>

<**groupId**>com.github.spotbugs</**groupId**>

<**artifactId**>spotbugs-maven-plugin</**artifactId**>

<**version**>3.1.10</**version**>

</**plugin**>

1. Under build -> reporting -> plugins, enter the below plugin,

<**plugin**>

<**groupId**>com.github.spotbugs</**groupId**>

<**artifactId**>spotbugs-maven-plugin</**artifactId**>

<**version**>3.1.10</**version**>

<**configuration**>

<**xmlOutput**>true</**xmlOutput**>

*<!-- Optional directory to put spotbugs xdoc xml report -->*

<**xmlOutputDirectory**>target/site</**xmlOutputDirectory**>

</**configuration**>

<**reportSets**>

<**reportSet**>

<**reports**>

<**report**>spotbugs</**report**>

</**reports**>

</**reportSet**>

</**reportSets**>

</**plugin**>

Executing using Maven:

1. In IntelliJ,
   1. navigate to Maven Projects panel
   2. Under Plugins -> spotbugs, double click any goal which you want to be executed.
   3. This shows errors/output in the below Run panel
2. Using site plugin,
   1. navigate to Maven Projects panel
   2. Under Plugins -> spotbugs, double click any goal which you want to be executed. For now, I prefer site:site to generate report
   3. Under project’s target folder, open site folder
   4. Open spotbugs.html to see the bug report

QAPlug

An Intellij IDEA Plugin To Manage Code Quality Which Integrates Tools Such As PMD, Checkstyle And Findbugs.

To install in Intellji:

<https://qaplug.com/download/> and restart Intellji

To run:

Right click project -> Analyze -> Analyze Code

Results will be displayed in Analysis panel below, which includes for Findbugs, Checkstyle,PMD

IntelliJ plugins installation

FireBug Intellij plugin:

File-> Settings -> Plugins -> Browse Repositories -> Search “Firebug” and install

PMD Intellij plugin:

File-> Settings -> Plugins -> Browse Repositories -> Search “Firebug” and install

CheckStyle Intellij plugin:

File-> Settings -> Plugins -> Browse Repositories -> Search “Firebug” and install

After installing the above restart IntelliJ

To run any of the plugin, Look for plugin in IntelliJ toolbar(bottom of IntelliJ) and run.