1. Install SonarQube server: <https://www.voyalab.com/2016/10/06/install-sonarqube-ubuntu/>
2. Install Sonar Scanner: <https://www.voyalab.com/2016/10/08/installing-sonarqube-scanner/>
3. Sonar report analyze: <https://www.voyalab.com/2016/10/12/analyze-sample-project-with-sonarqube/>
4. It required MySql server: <https://www.digitalocean.com/community/tutorials/how-to-install-the-latest-mysql-on-ubuntu-16-04>
5. It also need Java 8: <https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-get-on-ubuntu-16-04>

How to execute:

1. mvn clean install -e -DskipTests=false (execute test cases while building)
2. mvn sonar:sonar -Dsonar.host.url=<sonarQube host url with port>

How get coverage report:

1. During maven build, reports are generated in /target folder which are then used by sonar.
2. Sonar, by-default takes only unit test reports. Not integration test results.
3. So get overall coverage (unit + integration), use Surefire (for unit test) and Failsafe (for integration test) plugins in pom.xml

|  |
| --- |
| <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-surefire-plugin</artifactId> <version>2.12.1</version> <configuration> <excludes> <exclude>\*\*/\*IntegrationTest\*</exclude> </excludes> </configuration> </plugin>  <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-failsafe-plugin</artifactId> <version>2.12.4</version> <configuration> <includes> <include>\*\*/\*IntegrationTest\*</include> </includes> </configuration> <executions> <execution> <goals> <goal>integration-test</goal> <goal>verify</goal> </goals> </execution> </executions> </plugin> |

1. JaCoCo agent is required to analyze surefire and failsafe reports.

|  |
| --- |
| <plugin> <groupId>org.jacoco</groupId> <artifactId>jacoco-maven-plugin</artifactId> <version>${jacoco.version}</version> <executions> <!– Prepares a variable, jacoco.agent.ut.arg, that contains the info to be passed to the JVM hosting the code being tested. –> <execution> <id>prepare-ut-agent</id> <phase>process-test-classes</phase> <goals> <goal>prepare-agent</goal> </goals> <configuration> <destFile>${sonar.jacoco.reportPath}</destFile> <propertyName>jacoco.agent.ut.arg</propertyName> <append>true</append> </configuration> </execution> <!– Prepares a variable, jacoco.agent.it.arg, that contains the info to be passed to the JVM hosting the code being tested. –> <execution> <id>prepare-it-agent</id> <!– moved to package phase to be sure all pre-integration test have it already set to bring up environments with jacoco agent as JVM params –> <phase>package</phase> <goals> <goal>prepare-agent-integration</goal> </goals> <configuration> <destFile>${sonar.jacoco.itReportPath}</destFile> <propertyName>jacoco.agent.it.arg</propertyName> <append>true</append> </configuration> </execution> </executions> </plugin> |

1. Use below properties in pom.xml for sonar-jacoco integration:

|  |
| --- |
| <properties>  <sonar.java.coveragePlugin>jacoco</sonar.java.coveragePlugin>  <sonar.dynamicAnalysis>reuseReports</sonar.dynamicAnalysis>  <jacoco.version>0.7.6.201602180812</jacoco.version>  <jacoco.outputDir>${project.build.directory}/jacoco</jacoco.outputDir>  <jacoco.out.ut.file>jacoco-ut.exec</jacoco.out.ut.file>  <jacoco.out.it.file>jacoco-it.exec</jacoco.out.it.file>  <sonar.jacoco.reportPath>${jacoco.outputDir}/${jacoco.out.ut.file}</sonar.jacoco.reportPath>  <sonar.jacoco.itReportPath>${jacoco.outputDir}/${jacoco.out.it.file}</sonar.jacoco.itReportPath>  </properties> |

1. Login to sonarqube server (default username/password: admin/admin)
2. Go to Administration > General Settings > Analysis Scope > Code Coverage and mention the packages/classes which needs to be excluded for coverage.
3. Else, add the properties to exclude packages/classes from pom:

|  |
| --- |
| <properties>  <sonar.exclusions> [...] </sonar.exclusions>  </properties>  Example:  <properties>  <sonar.coverage.exclusions>  \*\*/domain/\*\*/\*,  \*\*/pojos/\*  </sonar.coverage.exclusions>  </properties>  <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <sonar.host.url>http://www.example.com/</sonar.host.url>  <sonar.jdbc.url>jdbc:postgresql://www.example.com/sonar</sonar.jdbc.url>  <sonar.jdbc.driver>org.postgresql.Driver</sonar.jdbc.driver>  <sonar.jdbc.username>sonar</sonar.jdbc.username>  <sonar.jdbc.password>sonar</sonar.jdbc.password>  <sonar.exclusions>org/binarytherapy/generated/\*\*/\*, \*\*/GuiceBindComposer.java</sonar.exclusions>  <sonar.dynamic>reuseReports</sonar.dynamic>  </properties> |

1. Run mvn cmd for build and sonar analysis.

Notes:

1. Best practice is to end the unit test classes with \*UnitTest.java and integration test classes with \*IntegrationTest.java

Links:

1. <https://aroundthecode.org/2014/07/07/unit-and-integration-tests-coverage-with-sonarqube-and-jacoco/>
2. <http://ashismo.github.io/java-code%20quality%20analyzer/2016/03/14/SONAR-JaCoCo-And-Maven-Integration>
3. <https://docs.sonarqube.org/display/SONAR/Narrowing+the+Focus#NarrowingtheFocus-IgnoreCodeCoverage>
4. <https://docs.sonarqube.org/display/PLUG/SonarJava>
5. <https://docs.sonarqube.org/display/PLUG/Code+Coverage+by+Unit+Tests+for+Java+Project>

Final Target folder structure:

