

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube <ul style="list-style-type: none">Installation and SetupHow to Send a Project to SonarQubeBugsCode SmellsCoverageAdding SonarQube Issues to GitHubStatic analysis for Node.js
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security

Installation and Setup

Contents

- [Overview](#)
- [Tutorial](#)
 - [Prerequisites](#)
 - [Steps](#)
- [Exercises](#)

Overview

SonarQube is an open-source platform developed by SonarSource used for continuous inspection of static code quality. The platform analyses the code fed into it, providing detailed reports of bugs, vulnerabilities, code duplications and code smells.

With the vanilla build is supports 27 languages ranging from **Java** to **php** and **<XML/>**. By using plugins it can be used for more codebases and different metrics/reports can be generated.

The platform offers an automated way to get feedback on the code written where it can easily and rapidly point out simple duplications and vulnerabilities that may be missed by human inspection down the line.

When setting up a project to be reviewed by SonarQube there are a few steps to go through, and multiple ways of setting the platform up to work with your project.

Tutorial

Prerequisites

- Maven Java project with a pom.xml file being used
- Java 8+ installed locally
- Maven installed (with path variables) locally
- Running windows (ideally 10)

Steps

1) The pom.xml has to be configured to include the SonarQube dependency as well as a plugin to allow SonarQube to see the projects test coverage.

```
<dependency>
  <groupId>org.sonarsource.scanner.maven</groupId>
  <artifactId>sonar-maven-plugin</artifactId>
  <version>3.8.0.2131</version>
</dependency>
```

Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

```
<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-plugin</artifactId>
  <version>0.8.5</version>
  <executions>
    <execution>
      <goals>
        <goal>prepare-agent</goal>
      </goals>
    </execution>
    <!-- attached to Maven test phase -->
    <execution>
      <id>report</id>
      <phase>test</phase>
      <goals>
        <goal>report</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

You can also include a profile in your pom.xml to bypass the username and password later on, but without using environment variables/git ignore you will be pushing your username and password up to github

```
<profiles>
  <profile>
    <id>sonar</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <properties>
      <sonar.host.url>http://localhost:9000</sonar.host.url>
      <sonar.user>admin</sonar.user>
      <sonar.password>admin</sonar.password>
    </properties>
  </profile>
</profiles>
```

- 2. The following SonarQube [link](#) will take you to the 'get started' docs for SonarQube.
- 3. Navigate to the **From the zip File** section and download the linked Community Edition file (you may need to choose a version, choose the free version)
- 4. Unzip the contents of this file in a useful location (I.e, if using windows C:\sonarqube)
- 5. Using a CLI of your choice access the file StartSonar.bat, likely using a command similar to the following:
C:\sonarqube\bin\windows-x86-64\StartSonar.bat

Check the actual contents of the unzipped file to see if this is correct

If this has been configured correctly the terminal should hang for 5/10 seconds and will then output the logs of the server starting up. If the terminal doesn't allow you to type anything else at the bottom after outputting logs, the server is up and running correctly.

If the installation fails at this point one of the common issues is the ports 9000 and 9001 are already in use on your machine.

- a) Open a command prompt terminal (type in cmd in the windows search bar)
- b) Type in the following command into the command prompt**netstat -an**
- c) Look through the list of ports for 9000 and 9001 being in use
- d) If either of them are in use, download [TCP View](#) to get a more detailed view of what's in use

e) Install the application and find the process using ports 9000 and 9001

f) Depending on the application using the port try to **stop** or **kill** it using common sense on what application is already using that port

6. If the SonarQube server has been configured correctly access **localhost:9000** on your browser. A login page will open, enter the default user and password (admin, admin)

7. Once your password has been entered you will likely have to rewrite your password to a more secure password

8. You will be taken to the SonarQube splash page where there shouldn't be any projects up and running at the minute.

9. Open up a new CLI (Powershell, GitBash etc.) and cd to the java project directory. When in the right directory use the following maven command:

```
mvn clean
```

10. If you haven't entered your password username into the pom.xml use the following command in your cli (Changing password/username as appropriate)

```
mvn sonar:sonar -Dsonar.host.url=http://localhost:9000 -Dsonar.login=admin -Dsonar.password=admin
```

if you have included the profile in your pom.xml use the following command `

```
mvn sonar:sonar
```

11. Once the project has been built, access localhost:9000 and you should be able to see your project on the page. Click the name of the project to access it in more detail

Exercises

There are no exercises for this module.