**SonarQube**

SonarQube is a powerful tool for improving code quality, identifying and fixing issues early in the development process, and ensuring that software projects meet quality and security standards.

**SonarQube with Spring boot**

Demo project was done to illustrate the SonarQube Integration with Springboot

**Jacoco Plugin**:

JaCoCo (Java Code Coverage) is a widely used code coverage tool for Java projects. It helps developers understand how much of their codebase is being tested by providing insights into which parts of the code are exercised by tests and which are not. SonarQube integrates with JaCoCo through a plugin, allowing users to analyze code coverage metrics alongside other code quality metrics.

**Dependencies used for the project:**

<dependency>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-starter-data-mongodb</artifactId>

</dependency>

<dependency>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springdoc</groupId>

<artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>

<version>2.2.0</version>

</dependency>

<dependency>

<groupId>org.sonarsource. scanner.maven</groupId>

<artifactId>sonar-maven-plugin</artifactId>

<version>3.10.0.2594</version>

</dependency>

<dependency>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springdoc</groupId>

<artifactId>springdoc-openapi-ui</artifactId>

<version>1.6.9</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-boot-starter</artifactId>

<version>3.0.0</version>

</dependency>

<dependency>

<groupId>com.fasterxml</groupId>

<artifactId>classmate</artifactId>

<version>1.5.1</version> <! -- Use the latest version -->

</dependency>

<dependency>

<groupId>com.github. ulisesbocchio</groupId>

<artifactId>jasypt-spring-boot-starter</artifactId>

<version>3.0.5</version>

</dependency>

<dependency>

<groupId>org.sonarsource. scanner.maven</groupId>

<artifactId>sonar-maven-plugin</artifactId>

<version>3.10.0.2594</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.9.2</version>

</dependency>

**Pluggin used:**

<plugin>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

<plugin>

<groupId>com.github. ulisesbocchio</groupId>

<artifactId>jasypt-maven-plugin</artifactId>

<version>3.0.5</version>

</plugin>

<plugin>

<groupId>org.jacoco</groupId>

<artifactId>jacoco-maven-plugin</artifactId>

<version>0.8.8</version>

<executions>

<execution>

<id>prepare-agent</id>

<goals>

<goal>prepare-agent</goal>

</goals>

</execution>

<execution>

<id>report</id>

<goals>

<goal>report</goal>

</goals>

</execution>

</executions></plugin>

**Steps for Checking code coverage using SonarQube.**

* Start Sonarqube.bat file.
* Hit the localhost:9000 API.
* It will redirect to SonarQube UI
* Go to my account.
* Click on Security
* Generate the user token by giving it a name and expiration date.
* Copy the Token
* In the Spring boot application, Run the application as Maven Build using the command.

“clean org.jacoco:jacoco-maven-plugin:prepare-agent install”

* The jar file of the application will be generated.
* Again, Run the application as Maven build using the command.

“sonar:sonar -Dsonar.login=squ\_fe83e03cec123869aab8eef3b01f7ea5b0bfd52a”

* If the above command is not working,use these commands

“verify sonar:sonar -Dsonar.host.url=http://localhost:9000

-Dsonar.login=sqp\_4f9d9eb6d45bfaa5bb273c78c39fb6a2a1c7bfaa -Dsonar.login=admin -Dsonar.password=root”

“sonar:sonar -Dsonar.host.url=http://localhost:9000 -Dsonar.login=sqp\_4f9d9eb6d45bfaa5bb273c78c39fb6a2a1c7bfaa -Dsonar.login=admin -Dsonar.password=root”

* If the build is successful, the application will be available in the SonarQube UI.
* Analyse the code by viewing the Bugs, Vulnerabilities, Code Smells, Duplications
* Fix those in the code and rerun the application by using the same steps

**Sonar Cloud with GitHub Integration**

Analyse the Project directly in SonarCloud by pulling the code from GitHub.

Steps for Checking code coverage using SonarCloud.

* Click on Analyse new Project.
* Import another Organization.
* Import an Organization from GitHub.
* Login using GitHub.
* Click on Only select repositories.

If it shows an error that already an organization is there, then create the organisation manually by entering the name and key.

* The repositories of the GitHub will be displayed, select the Repository you want to setup and Click on Setup and select on Previous Version
* Click on create Project.
* Analyse the code by viewing the Bugs, Vulnerabilities, Code Smells, Duplications
* Change the code and follow the above steps.