COURSEWARE

Coverage

Contents

Overview

Code coverage in SonarQube is used to check how much of the source code has been covered by unit tests.
Coverage is worked out by checking the overall line coverage, and the conditional coverage.
Line coverage checks how many of the total number of executable lines have been covered by unit tests, and conditional coverage checks to make sure that all possible outcomes of a conditional statement have been covered by unit tests.

Tutorial

Setting up a Report Plugin

In order to get SonarQube to evaluate our code coverage, we will need to setup a report plugin so that SonarQube can read from those reports and output our overall coverage.
We will be going over how to set this up for Java source code.
In order to bring dependencies and plugins into our code, we need our project to be a Maven project.
Once your project is a Maven project, with Java source code, follow these steps in order to setup code coverage reports for SonarQube to use.

1. Add the following dependency to your pom.xml file, within your "<dependencies></dependencies>" tags.

```
<dependency>
    <groupId>org.jacoco</groupId>
    <artifactId>jacoco-maven-plugin</artifactId>
    <version>0.8.5</version>
</dependency>
```

2. Next we need to add the following plugin within our "<build><plugins></plugins></build>" tags.

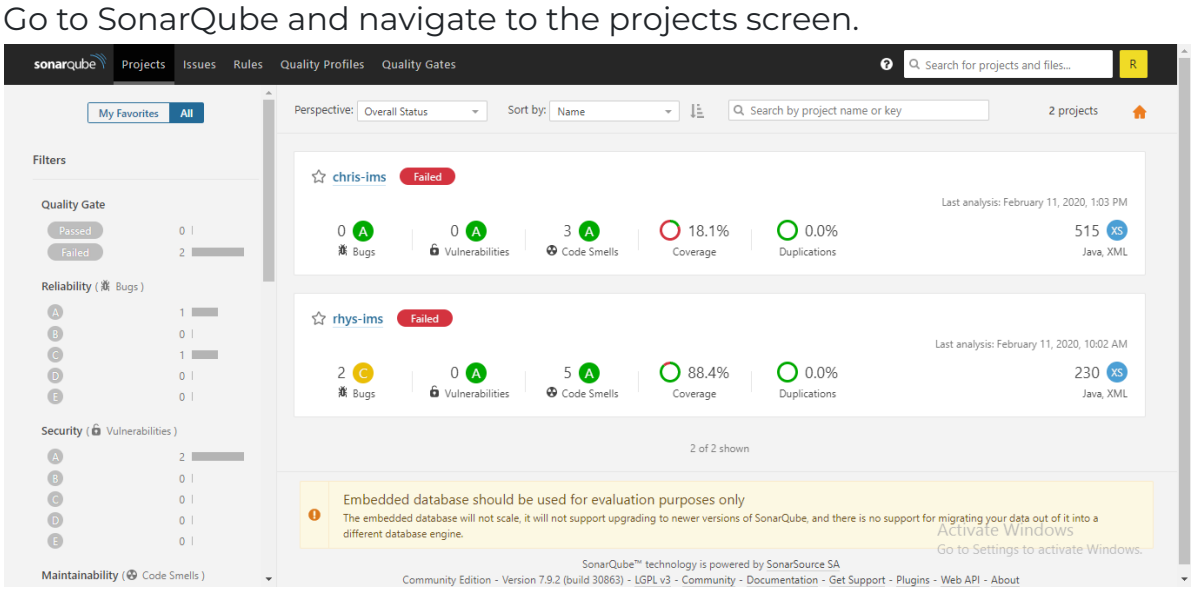
```
<plugin>
    <groupId>org.jacoco</groupId>
    <artifactId>jacoco-maven-plugin</artifactId>
    <version>${jacoco.version}</version>
    <executions>
        <execution>
            <goals>
                <goal>prepare-agent</goal>
            </goals>
        </execution>
        <execution>
            <id>jacoco-report</id>
            <phase>test</phase>
            <goals>
                <goal>report</goal>
            </goals>
            <!-- default target/jscoco/site/* -->
            <configuration>
                <outputDirectory>
                    target/jacoco-report
                <outputDirectory>
            </configuration>
        </execution>
    </executions>
</plugin>
```

3. Once we save our pom.xml file with this new dependency and new plugin JaCoCo will begin to generate code coverage reports for us whenever we build our project.
   In order to see the code coverage report, we will need to send our project to SonarQube using the "mvn sonar:sonar" command.

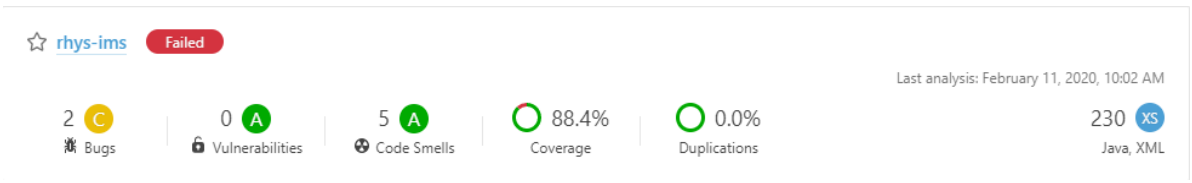
Viewing Our Overall Coverage

Now that we have JaCoCo setup and have sent our code to SonarQube for analysis, we can go and see how much of our code is covered by unit tests.
In order to find out your overall code coverage follow these steps:

1. Go to SonarQube and navigate to the projects screen.

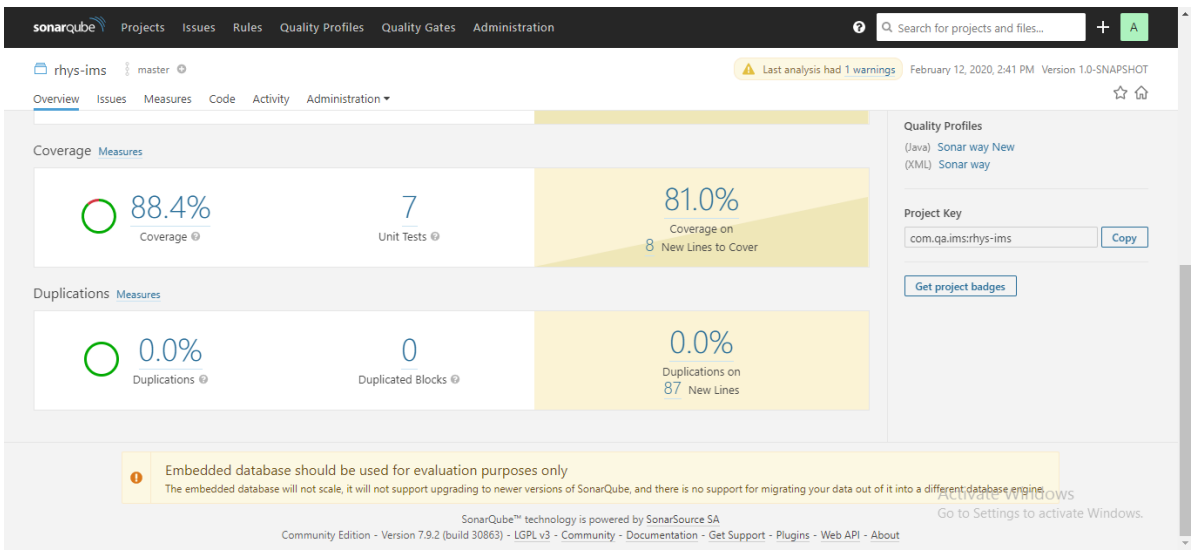


2. Find your project in the project window.

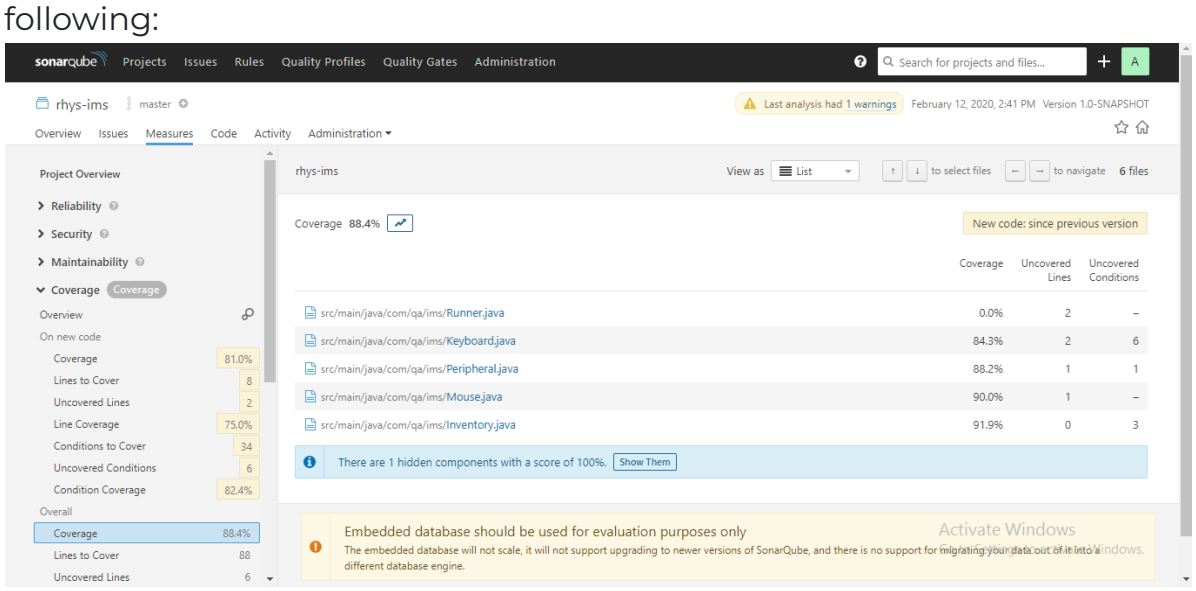


   On the project card we can see that it has a "Coverage" value with a percentage above it, in this case 88.4%.
   This value means that 88.4% of our code is covered, meaning that we have 11.6% uncovered, which could be uncovered lines of executable code or uncovered conditional outcomes.

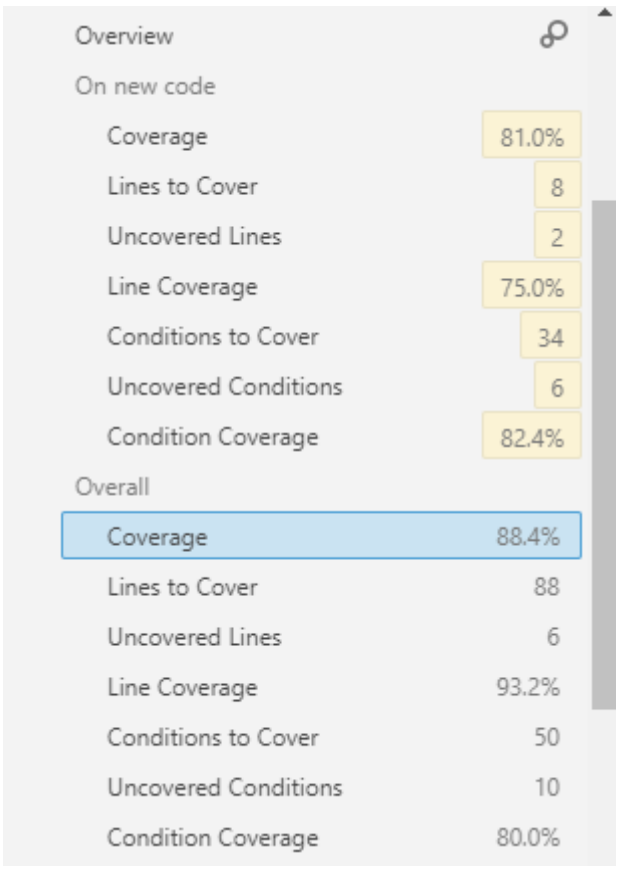
3. In order to see which parts of our code are uncovered, click on the project's name to go to the project dashboard, and scroll down to the "Coverage" section.



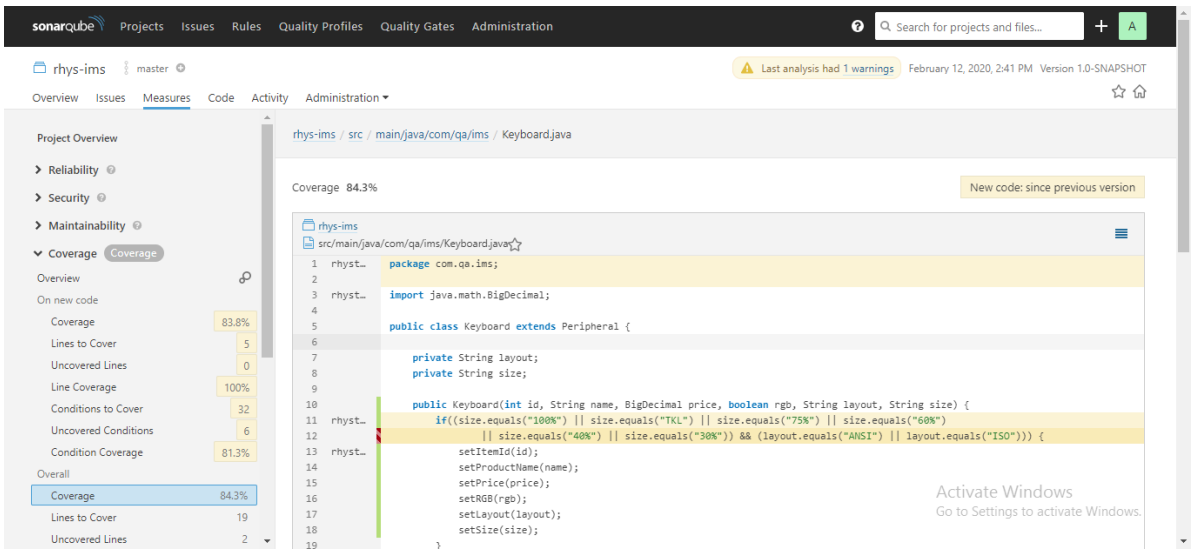
4. Click on the value above "Coverage" on the Coverage card, in this case it is the "88.4%" value, and you should land on a page that looks similar to the following:



In this window we can see the overall coverage value at the top, and we have a list of all of our classes in our project that don't have 100% coverage. To the right of the class name, you can see the overall coverage for that specific class, the number of uncovered executable lines, and the number of uncovered conditional outcomes.
If we scroll down on the left hand section of this window we can also see both overall coverage metrics and coverage metrics on new code (see below); including coverage percentage, total lines to cover, uncovered lines, total conditions to cover, and uncovered conditions.



5. If you want to see which lines or conditions are uncovered, you can click on the class name on the right hand side of the coverage window and you will get a screen that looks similar to the following:



On this window we can see the classes code, and in the left hand side bar each line will have a solid green line, a dashed red line or a solid red line.

A *solid green line* indicates that the line or conditional is fully covered by tests.
A *dashed red line* indicates that the conditional is only partially covered by tests.
A *solid red line* indicates that the line or conditional is not covered by tests at all.

If a conditional has a dashed red line, you can see how many of the possible outcomes have been covered by tests by hovering over the line, and you will see something like this:

```
 ![PartiallyCoveredConditionalPopUp](https://qa-courseware-images.s3.eu-west-2.amazonaws.com/sonarqube/coverage/006.png)
```

Exercises

There are no exercises for this module.