

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"><li>Installation and Setup</li><li>How to Send a Project to SonarQube</li><li>Bugs</li><li>Code Smells</li><li>Coverage</li><li>Adding SonarQube Issues to GitHub</li><li>Static analysis for Node.js</li></ul>
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security

# Bugs

## Contents

- Overview
  - Issue Severity
  - Issue Life Cycle
  - Statuses
  - Resolutions
- Tutorial
  - How to See What Bugs are Affecting Your Code
  - How to Change the Status of a Bug
- Exercises

## Overview

In SonarQube Bugs are issues that are code breaking, meaning your code will not work as intended and need fixing immediately. Bugs will have a severity attached to them, indicating how code breaking the issue is as well as an issue life cycle to show whether it has been worked on or not.

## Issue Severity

Bugs can have one of five levels of severity, depending on how code breaking it is, they can be "BLOCKER", "CRITICAL", "MAJOR", "MINOR", and "INFO".

- BLOCKER** issues are bugs with a high probability to affect the behaviour of the live application. Examples of BLOCKER issues are possible memory leaks or connections not being closed. Offending code being flagged with a BLOCKER issue must be fixed immediately.
- CRITICAL** issues are bugs which have a low chance of affecting the behaviour of the live application, or present a security flaw within the code. Examples of CRITICAL issues are empty catch blocks, or code that could allow an SQL injection attack through. Offending code being flagged with a CRITICAL issue must be reviewed immediately.
- MAJOR** issues are quality flaws which can greatly impact the developers productivity. Examples of MAJOR issues could be code that is not covered by tests, duplicated code blocks, or unused parameters.
- MINOR** issues are quality flaws which can slightly impact the developers productivity. Examples of MINOR issues could be code lines being too long, or switch statements having less than 3 cases.
- INFO** issues are not necessarily a bug or a quality flaw, but a finding instead. An example of an INFO issue could be a comment starting with "TODO" to show you that you may still have work to complete on the source code.

## Issue Life Cycle

## Statuses

Once an issue is flagged it will go through the issue life cycle taking on one of five statuses.

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

- **Open** : An issue will have the open status when it is first flagged by SonarQube.
- **Confirmed** : An issue can be given the confirmed status manually and indicates that the issue has been verified, and is a valid issue.
- **Resolved** : An issue can be given the resolved status manually and indicates that the issue should be fixed, and the next analysis should close the issue.
- **Reopened** : An issue that had the resolved status between last analysis and newest analysis, but hasn't actually been corrected, this is set automatically by SonarQube.
- **Closed** : An issue that has been closed, this is set automatically by SonarQube for automatically created issues.

## Resolutions

Closed issues will have one of two resolutions:

- **Fixed** : Set automatically by SonarQube when a new analysis shows that the issue is no longer there, or when the file is removed from the project.
- **Removed** : Set automatically by SonarQube when the rule that flagged the offending code is no longer available. The rule may not be available due to it being disabled, removed or removed from the quality profile that the project runs against.

Resolved issues will also have one of two resolutions:

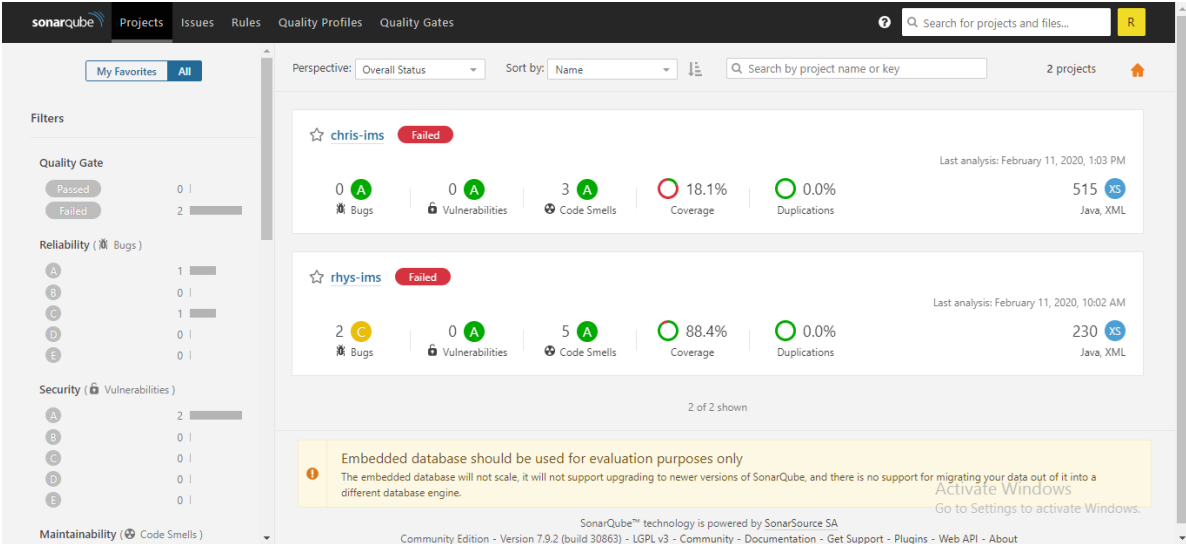
- **False Positive** : An issue can be manually marked as False Positive to resolve the issue, and generally is used to show that the rule has flagged code that shouldn't have been flagged. This would likely be an administrator privilege.
- **Won't Fix** : An issue can be manually marked as Won't Fix to resolve the issue, and is generally used to show that the flagged rule isn't a huge issue so it won't be fixed any time soon.

## Tutorial

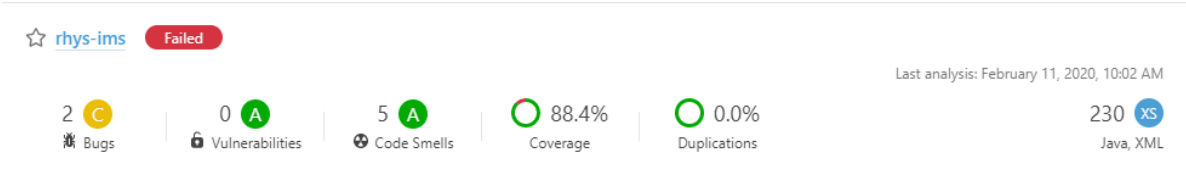
### How to See What Bugs are Affecting Your Code

In order to see what bugs are currently flagged on your project follow these steps:

1. Login to SonarQube.
2. Navigate to the projects window, a button for it can be found in the top navigation bar, it should look something like this:

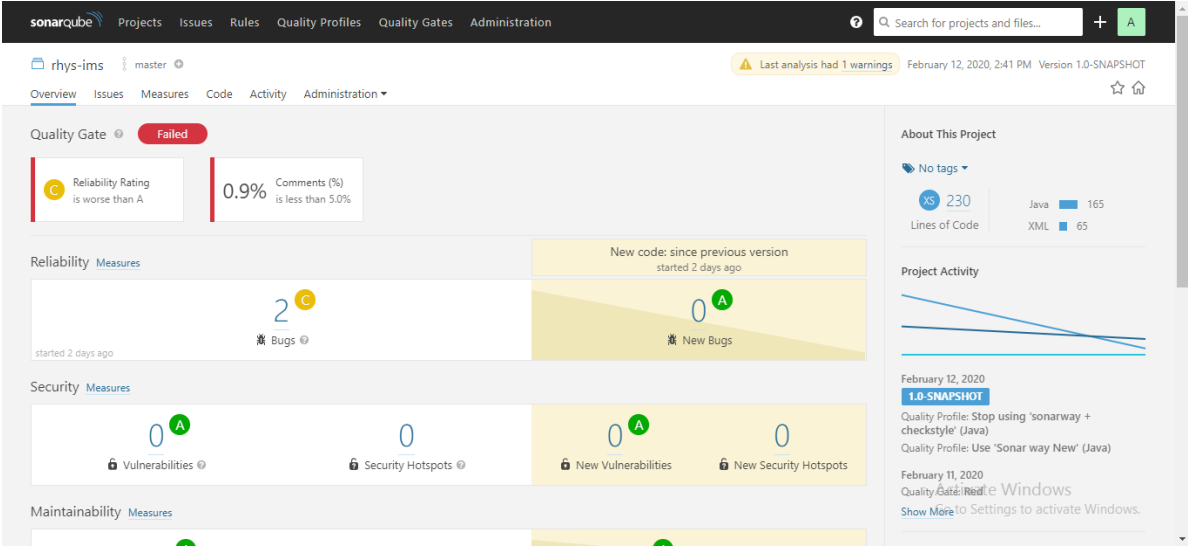


3. Find your project and click on the project name, in this example it is "rhys-ims".

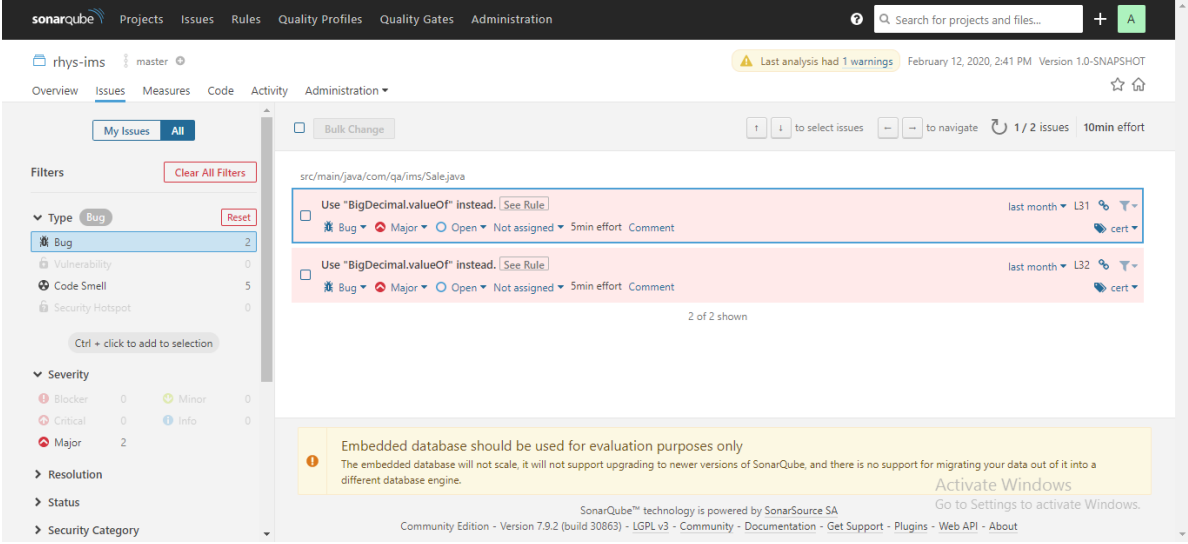


4. Once you land on the project's dashboard, scroll down to the "Reliability" section.  
This section will tell you how many "Bugs" you have, click on the number

above "Bugs" to see what they are.



5. Once you click on the number above the "Bugs" text you should land on the following screen, which shows you what bugs you have.



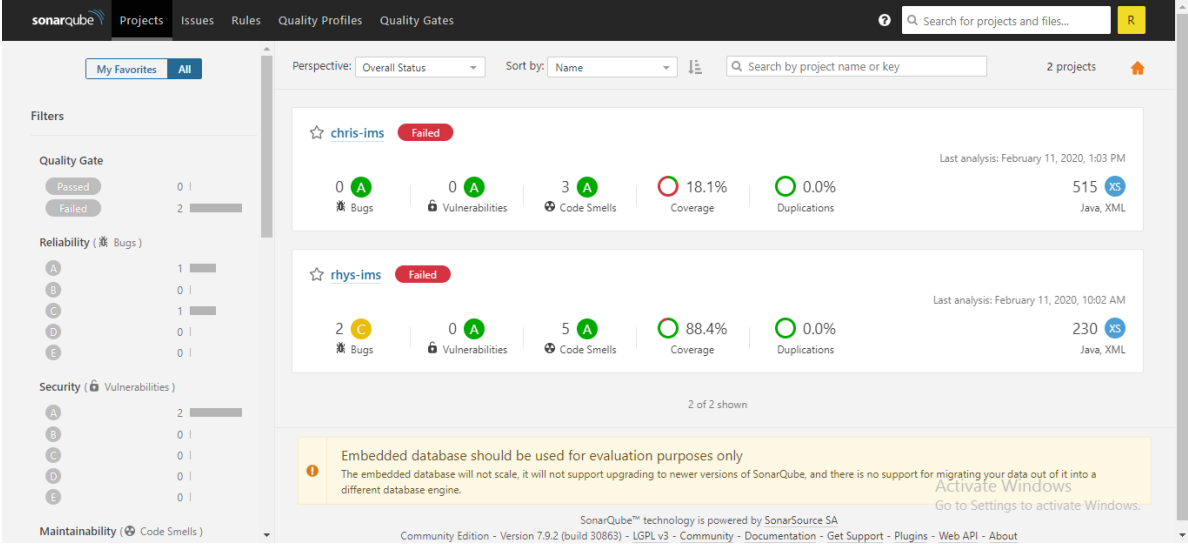
6. At the bottom of the red box for each bug, you can see what severity the bug is, whether it has been resolved or not, you can assign it to someone to fix (this could be yourself), and you can add a comment to the bug. You can also see how long ago the bug was flagged in the top right of the red box.

7. If you want to take a closer look at the offending code, you can click the red box and it will take you to the offending source code.

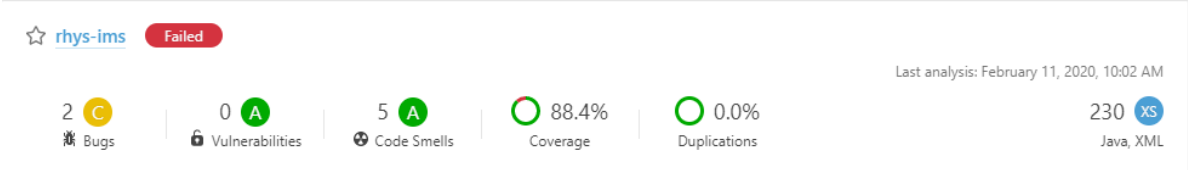
## How to Change the Status of a Bug

In order to change the status of a bug, follow these steps:

1. Login to SonarQube.
2. Navigate to the projects window, a button for it can be found in the top navigation bar, it should look something like this:



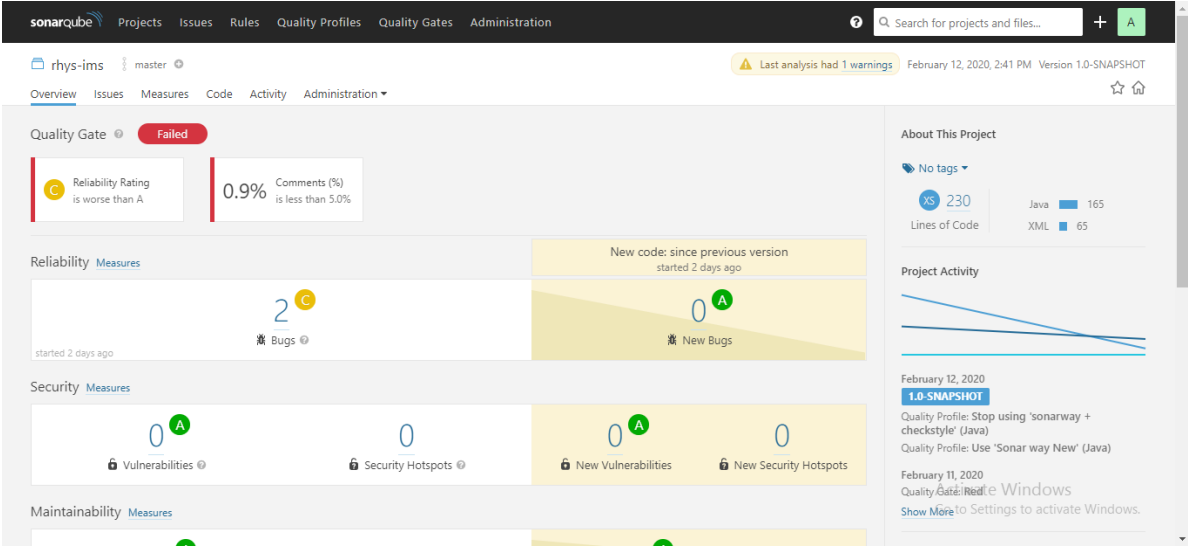
3. Find your project and click on the project name, in this example it is "rhys-ims".



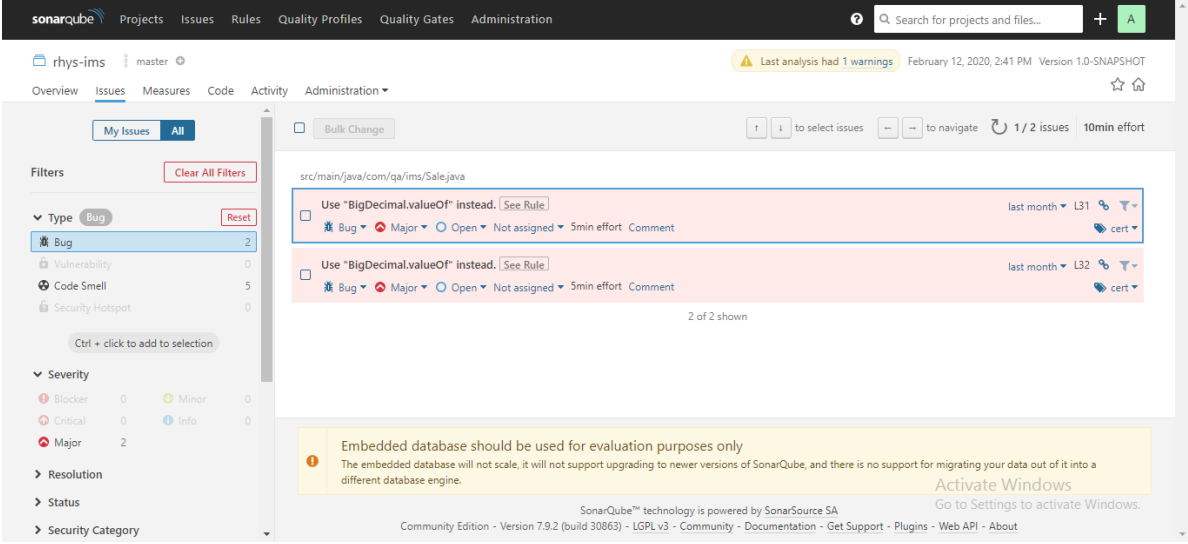
4. Once you land on the project's dashboard, scroll down to the "Reliability" section.

This section will tell you how many "Bugs" you have, click on the number

above "Bugs" to see what they are.

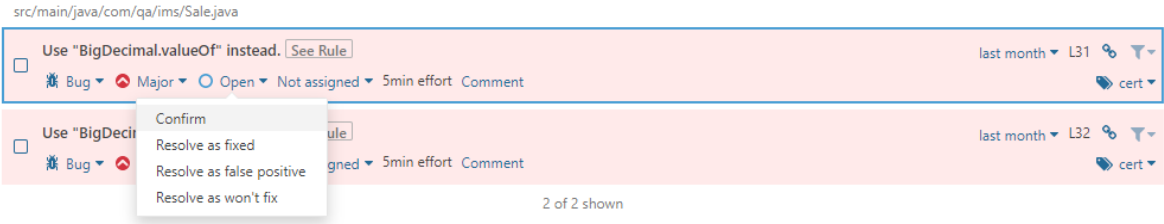


5. Once you click on the number above the "Bugs" text you should land on the following screen, which shows you what bugs you have.



6. Each bug will have a red box around it, at the bottom of the box is some text that by default will be "Open" when the issue is first flagged by SonarQube.

Click on the "Open" text and you will get the following drop down box.



7. Simply click on the new status you want the bug to have.

Changing a bug's status to anything other than "Confirm" may require administrator privileges.

## Exercises

There are no exercises for this module.