`

A black and white logo

Description automatically generated with low confidence

|  |
| --- |
| **Code Coverage of Angular application using**  **Karma-jasmine framework** |

**Code Coverage of Angular application using**

**Karma-jasmine framework**

# Table of Contents

[Table of Contents 2](#_Toc476761688)

[1. Application Details 3](#_Toc476761689)

[2. Generating code coverage 4](#_Toc476761690)

3. PERFORMING SCA AND DISPLAYING CODE COVERAGE IN SONARQUBE…………………………. 5

# Application Details

**Angular Application Details :**

* Angular CLI -8.3.29
* Karma-5.2.3
* Jasmine-3.6.0
* Karma-jasmine-coverage-istanbul-reporter- 3.0.2

**GitHub repository link of demo application:**

<https://github.com/deepikab97/Angular-test-project.git>

**Karma-Jasmine :**

* Karma is a direct product of the AngularJS team from struggling to test their own framework features with existing tools. As a result of this, they made Karma and have transitioned it to Angular as the default test runner for applications created with the Angular CLI.
* Karma also provides you options to replace Jasmine with other testing frameworks such as Mocha and QUnit or integrate with various continuous integration services like Jenkins, TravisCI, or CircleCI.
* Jasmine is a JavaScript testing framework  for testing JavaScript code that plays very well with Karma that supports a software development practice called [Behaviour-Driven Development](https://en.wikipedia.org/wiki/Behavior-driven_development), or BDD for short.
* Jasmine, and BDD in general, attempts to describe tests in a human readable format so that non-technical people can understand what is being tested. However even if you *are* technical reading tests in BDD format makes it a lot easier to understand what’s going on.
* To exlplore more about karma-jasmine…
  + Official Website

<https://www.digitalocean.com/community/tutorials/testing-angular-with-jasmine-and-karma-part-1>

<https://codecraft.tv/courses/angular/unit-testing/jasmine-and-karma/>

# Generating Code Coverage

**Configuring jasmine and karma runner :**

**Karma** is a console tool for running tests, which can track source code changes and display the percentage of code tests coverage. It is adjusted using the configuration file **karma.conf.js**, where the paths to tested files and to the files with tests should be specified.

**Coverage Quick Start :**

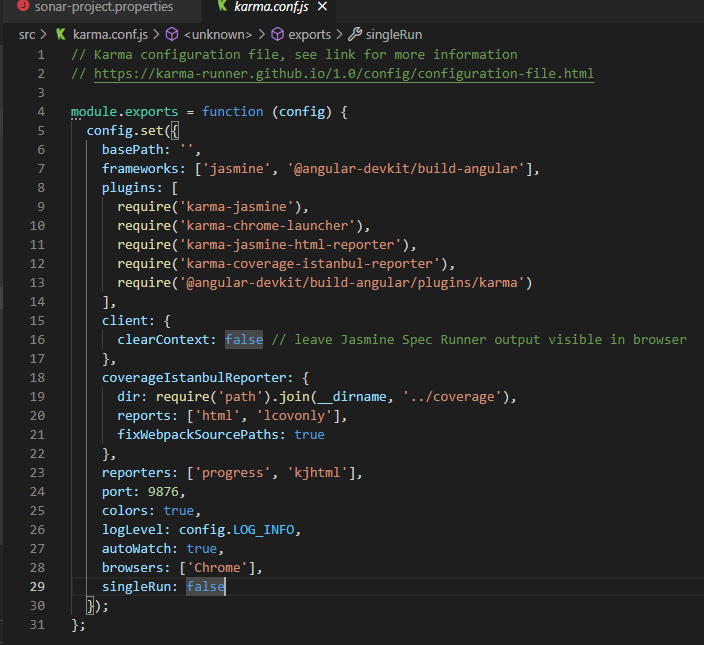
1. Install npm with all dependencies

$ npm install

1. Build the application

$ ng build

1. Running test cases first check there is Karma plugin installed or not (karma.conf.js)

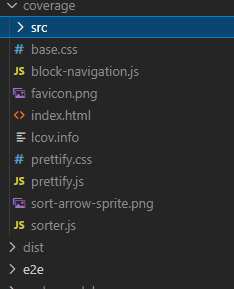


1. Install karma code coverage

$ npm install karma karma-coverage --save-dev

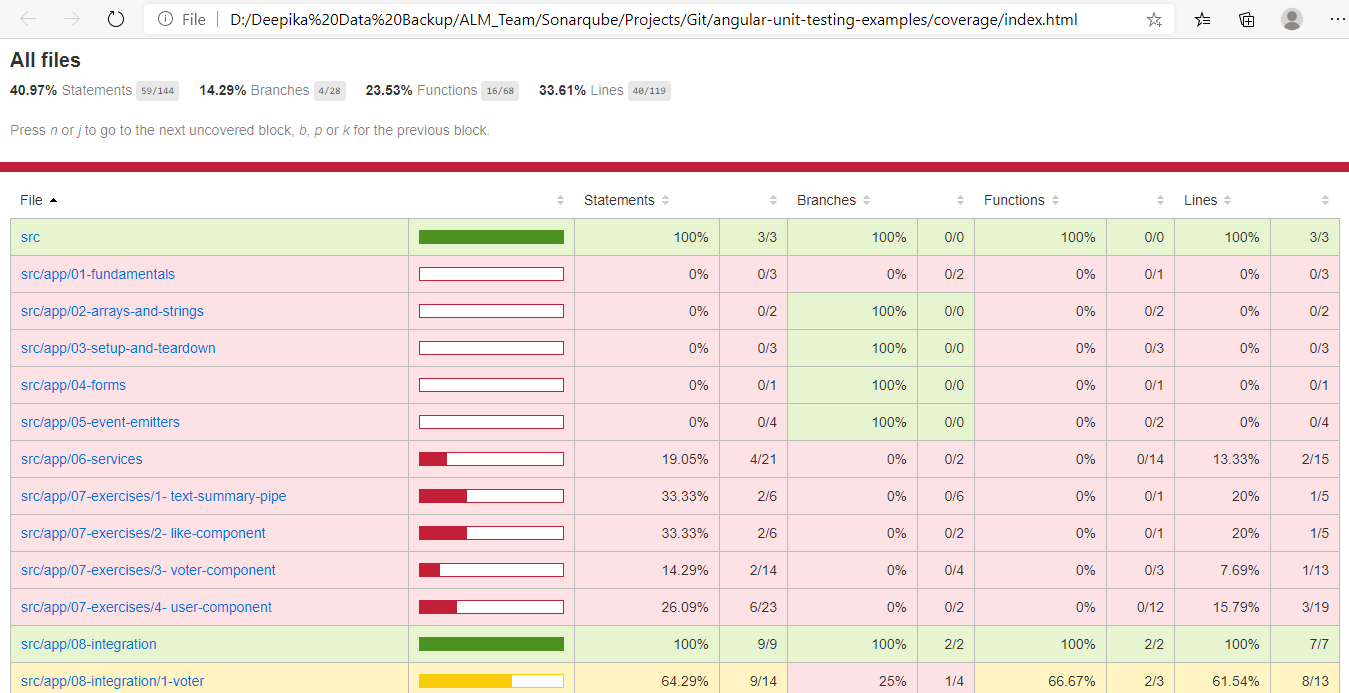
1. Generating coverage report

$ ng test --code-coverage



This step will generate the coverage report in project root directory as coverage. .html/.info

1. After running the code coverage open index.html in coverage directory



# Performing SCA And Displaying Code Coverage in SonarQube Dashboard

**sonar-project.properties file :**

sonar.host.url = http://sonar.cybage.com

sonar.projectKey = ALM\_Angular\_Sonar\_Demo:master

sonar.projectName = ALM\_Angular\_Sonar\_Demo

sonar.projectVersion = 1.0

sonar.login = sonarqube token

sonar.language = typescript

sonar.sources = src

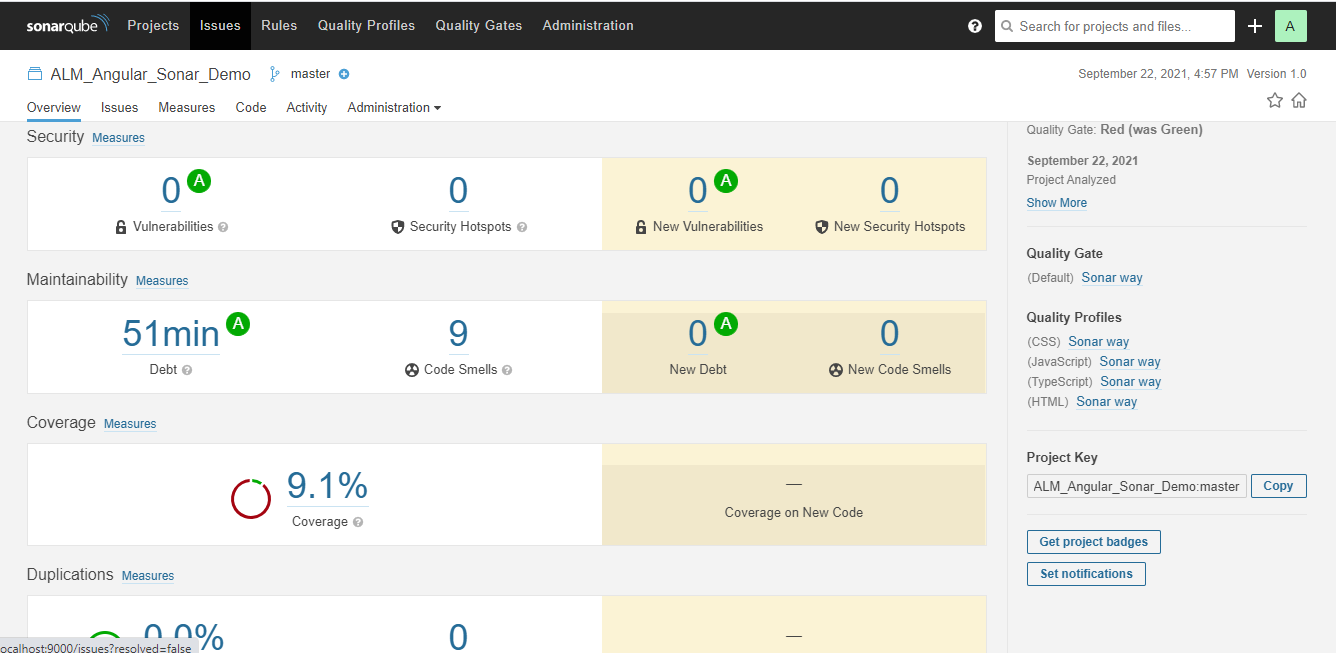
sonar.coverage.exclusions = \*.css

sonar.javascript.lcov.reportPaths = coverage/lcov.info

1. In order to display the code coverage in sonarqube serever dashboard we have to specify path of coverage report (locov.info) generated by karma

sonar.javascript.lcov.reportPaths = coverage/lcov.info

1. Execute the sonar scanner to analyse the source code.
2. SonarQube Dashboard



For More Details :

<https://docs.sonarqube.org/7.9/analysis/coverage/>