Building and Running Tests

In this lesson, we will learn how to build and run the tests using Gradle.

WE'LL COVER THE FOLLOWING

Building the project

Go to the project root to run the following command.

```
./gradlew clean build -x test
```

Commands to run the test files from command line

Go to the project root to run the command.

Tests for allure steps and attaching screenshots on failure

Web UI tests

```
./gradlew clean test --tests "com.educative.test.TestGoogleSearch" -Dbrowser=firefox
```

Mobile web simulation using Chrome tests

```
./gradlew clean test --tests "com.educative.test.TestMobileGoogleSearch" -Dbrowser=chrome
```

Selenium actions example tests

Alert test

```
./gradlew clean test --tests "com.educative.test.actions.TestAlert" -Dbrow
ser=firefox
```

Confirm pop-up test

```
./gradlew clean test --tests "com.educative.test.actions.TestConfirm" -Dbr
owser=chrome
```

Prompt test

```
./gradlew clean test --tests "com.educative.test.actions.TestPrompt" -Dbro
wser=firefox
```

Drop down test

```
./gradlew clean test --tests "com.educative.test.actions.TestDropDown" -Db
rowser=chrome
```

CheckBox test

```
./gradlew clean test --tests "com.educative.test.actions.TestCheckBox" -Db
rowser=firefox
```

Radio button test

```
./gradlew clean test --tests "com.educative.test.actions.TestRadioButton"
-Dbrowser=chrome
```

Wait test

```
./gradlew clean test --tests "com.educative.test.actions.TestWaits" -Dbrow
ser=firefox
```

Cookie test

```
./gradlew clean test --tests "com.educative.test.actions.TestCookie" -Dbro
wser=chrome
```

Drag & drop test

```
./gradlew clean test --tests "com.educative.test.actions.TestDragAndDrop"
-Dbrowser=chrome
```

Mouseover test

```
./gradlew clean  test --tests "com.educative.test.actions.TestMouseOver" -
Dbrowser=firefox
```

Double click test

```
./gradlew clean test --tests "com.educative.test.actions.TestDoubleClick"
-Dbrowser=chrome
```

Right click test

```
./gradlew clean test --tests "com.educative.test.actions.TestRightClick" -
Dbrowser=firefox
```

Capture screenshot test

```
./gradlew clean test --tests "com.educative.test.actions.TestCaptureScreen
shot" -Dbrowser=chrome
```

Overriding configuration from command line

eg - We can set the browser to firefox/chrome for a test as:

```
./gradlew test -Dbrowser=firefox
```

Generated report paths

- **TestNG** - `./test-output/index.html`
- **Allure** - `./build/reports/allure-report/index.html`
- **Gradle** - `./build/reports/tests/test/index.html`

To Open Allure report from the command line

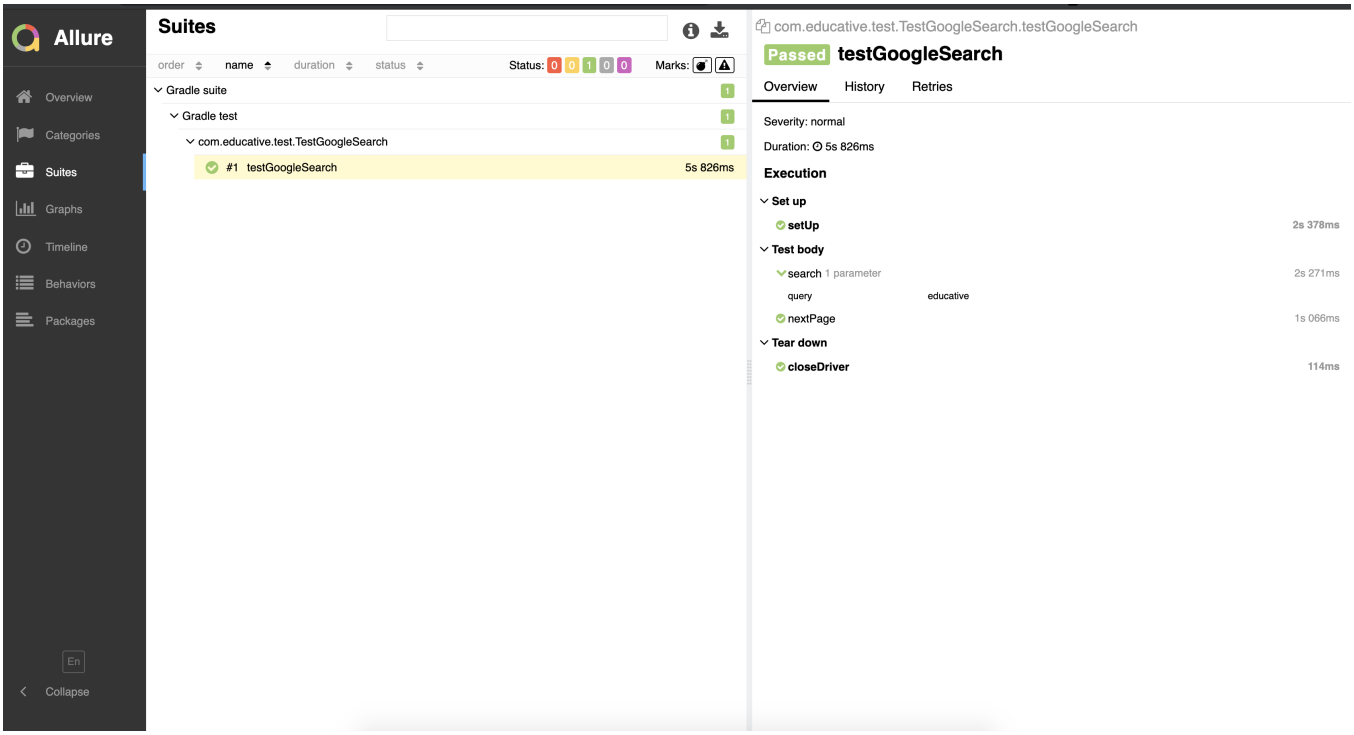
Once the test execution completes, please run the below command to open the Allure report

Allure report.

Go to the project root to run the command.

```
./gradlew allureServe
```

The above command will open a report similar to the following one:



Sample Allure report

Now, you can play around with the framework test code. In the next lesson, we will learn how to create a framework jar and upload it to an artifact.