

Building and Running Tests: REST API

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

We'll cover the following

- Building the project
- Commands to run the test files from the command line
 - Sample tests for reference
- Generated report paths
- How to open an Allure report from the command line

Building the project

Go to the project root to run the following commands:

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

Commands to run the test files from the command line

Go to the project root to run the command.

Sample tests for reference

GET request test

```
./gradlew clean build test --tests "io.educative.api.tests.TestGET"
```

POST request test

```
./gradlew clean build test --tests "io.educative.api.tests.TestPOST"
```

DELETE request test

```
./gradlew clean build test --tests "io.educative.api.tests.TestDelete"
```

Basic authentication test

```
./gradlew clean build test --tests "io.educative.api.tests.TestAuthentication"
```

Hamcrest matchers test

```
./gradlew clean build test --tests "io.educative.api.tests.TestHamcrestMatchers"
```

Async API test

```
./gradlew clean build test --tests "io.educative.api.tests.TestAsyncAPI"
```

Generated report paths

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

How to open an Allure report from the command line

Once the test execution is complete, please go to the project root and run the command below to open the Allure report:

```
./gradlew allureServe
```

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

The screenshot displays the Allure web interface. On the left is a sidebar with navigation links: Overview, Categories, Suites, Graphs, Timeline, Behaviors, and Packages. The main area is titled 'Suites' and contains a table of test results. The table has columns for 'name', 'duration', and 'status'. The status column shows a summary: 0 failures, 0 errors, 2 passed, 0 skipped, and 0 not executed. The table lists the following tests:

name	duration	status
REST API Test Suite		2
GET Test		2
io.educative.api.tests.TestGET		2
#1 testGetAllStudents	3s 220ms	Passed
#2 testGetStudentById	663ms	Passed

On the right side, there is a detailed view for the test 'io.educative.api.tests.TestGET.testGetAllStudents'. It shows a 'Passed' status, 'Severity: normal', and 'Duration: 3s 220ms'. Under the 'Execution' section, it lists the steps: 'Set up' (init, 1s 164ms) and 'Test body' (getStudents, 3s 182ms).

Now you can play around with the framework test code. In the next lesson we will

Now, you can play around with the framework test code. In the next lesson, we will learn about the SOAP API test framework.