

# Download the REST API Test Framework

In this lesson, we will download and understand the framework structure.

## We'll cover the following



- Downloading the framework code
- Project structure
  - `src/main/java`
  - `src/main/resources`
- Sample test written using the framework
  - `src/test/java`
  - `src/test/resources`

## Downloading the framework code #

This is a sample framework that can be extended and reused for any REST API testing requirement. It can be downloaded [here](#).

## Project structure #

```

.
├── build.gradle
├── gradle
│   └── wrapper
│       ├── gradle-wrapper.jar
│       └── gradle-wrapper.properties
├── gradlew
├── gradlew.bat
├── settings.gradle
└── src
    ├── main
    │   ├── java
    │   │   ├── io
    │   │   │   ├── educative
    │   │   │   │   ├── api
    │   │   │   │   │   ├── constants
    │   │   │   │   │   │   ├── Endpoints.java
    │   │   │   │   │   ├── helpers
    │   │   │   │   │   │   ├── StudentServiceHelper.java
    │   │   │   │   │   ├── model
    │   │   │   │   │   │   ├── Student.java
    │   │   │   │   │   └── utils
    │   │   │   │   │       ├── ConfigManager.java
    │   │   └── resources
    │   │       └── logback.xml
    └── test
        ├── java
        │   ├── io
        │   │   ├── educative
        │   │   │   ├── api
        │   │   │   │   ├── tests
        │   │   │   │   │   ├── TestAsyncAPI.java
        │   │   │   │   │   ├── TestAuthentication.java
        │   │   │   │   │   ├── TestDelete.java
        │   │   │   │   │   ├── TestGET.java
        │   │   │   │   │   ├── TestHamcrestMatchers.java
        │   │   │   │   │   └── TestPOST.java
        └── resources
            ├── config.properties
            └── testng.xml

```

20 directories, 19 files

src/main/java #

- `io.educative.api.constants.Endpoints` – a constant class for resource paths
- `io.educative.api.helpers.StudentServiceHelper` – a class for writing helper

methods used in making a request

- `io.educative.api.model.Student` – a POJO class used for serialization and deserialization of data
- `io.educative.api.utils.ConfigManager` – a class for configuration management that takes configuration parameters from project-specific `config.properties` and can be overridden by a JVM argument

`src/main/resources` #

- `logback.xml` – the configuration for logback

## Sample test written using the framework #

`src/test/java` #

- `io.educative.api.tests.TestGET` – a test for demonstrating a GET request
- `io.educative.api.tests.TestPOST` – a test for demonstrating a POST request
- `io.educative.api.tests.TestDelete` – a test for demonstrating a DELETE request
- `io.educative.api.tests.TestAuthentication` – a test for demonstrating a basic authentication for a GET request
- `io.educative.api.tests.TestHamcrestMatchers` – a test for demonstrating Hamcrest matchers usage.
- `io.educative.api.tests.TestAsyncAPI` – a test for handling an async request

`src/test/resources` #

- `config.properties` - a property file that holds the project-specific configuration parameters and can be read using `ConfigManager`
- `testng.xml` - a suite of TestNG tests

---

Now, we are familiar with the framework code structure. In the next lesson, we will learn to build, run, and experiment with the downloaded tests.