## Git Checkout

git remote add origin <repository\_url>

git add .

git commit -m "Initial commit"

git push -u origin master

## api-automation-framework/

│

├── src

│ ├── main

│ │ ├── java

│ │ │ ├── utilities/

│ │ │ │ └── Config.java # Contains base URLs and endpoints

│ │ │ └── services/

│ │ │ └── UserService.java # Contains API methods for user-related APIs

│ │ └── resources/

│ │ └── config/

│ │ └── api-config.properties # Stores additional configuration like tokens, etc.

│ ├── test

│ │ ├── java

│ │ │ ├── base/

│ │ │ │ └── BaseTest.java # Setup/teardown methods for tests, reporting setup

│ │ │ ├── tests/

│ │ │ │ ├── UserTests.java # Test cases for User APIs

│ │ │ │ ├── LoginTests.java # Test cases for Authentication APIs

│ │ │ └── suites/

│ │ │ ├── SmokeTests.java # Collection of smoke tests

│ │ │ ├── SanityTests.java # Collection of sanity tests

│ │ └── resources/

│ │ └── testdata/

│ │ └── user-test-data.json # JSON file containing test data

│

├── pom.xml # Maven dependencies (RestAssured, TestNG, Extent Reports, etc.)

├── Jenkinsfile # Jenkins pipeline configuration

├── smoke-test-suite.xml # TestNG suite for smoke tests

├── sanity-test-suite.xml # TestNG suite for sanity tests

├── test-output/ # Reports generated after tests run (e.g., Extent reports)

│ └── report.html # Extent report file

├── README.md # Project documentation

└── .gitignore # Ignore unnecessary files in version control (e.g., target/, test-output/)

## Create Folder Structure

mkdir -p src/main/java/com/example

mkdir -p src/test/java/com/example

This will create the necessary folders for your main Java code (src/main/java) and your test code (src/test/java).

## Create Pom.xml

This pom.xml includes:

* **Group ID, Artifact ID, and Version:** Basic project info.
* **Dependencies:** Adds REST Assured and TestNG for your API automation framework.
* **Build Plugins:** Configures the Maven compiler plugin to use Java 8 (source and target set to 1.8).

## Create Java Class

Inside src/main/java/com/example, you can create your main code classes. For example, you can create a simple test for REST Assured:

1. Navigate to src/test/java/com/example.

 **Clone the Project**:

* git clone https://github.com/your-repo-url.git

 **Run the Tests Locally**:

* Run the command mvn clean test to run the test suite.

 **Run Specific Suites**:

* Use mvn test -DsuiteXmlFile=smoke-test-suite.xml for the smoke suite.
* Similarly, for sanity suite: mvn test -DsuiteXmlFile=sanity-test-suite.xml.

 **Jenkins Integration**:

* Push the project to a Git repository, and Jenkins will trigger the pipeline based on your Jenkinsfile configuration.

. given

* URL
  + baseUri
* Headers
  + Header(
* Query params
* auth

.when

* Method
* path

.then

* assertions
* Src
  + Main
    - Java
      * Services
        + (Common methods or reusable methods)
    - Utilities
      * Test data (properties/Java)
  + Test
    - Tests
      * Mrthod calling and assertions6666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666
  + Pom.xml
  + Dockerfile
  + Jenkinsfile
  + Output
    - Report.html (extent reports)