**REST API Automation User Guide**

1. **Automation Framework overview**
2. **Architecture diagram and explanation**
3. **Directory Structure**
4. **Configuration.properties**
5. **Project usage steps**
6. **Reporting**
7. **Enhancements**

Framework overview

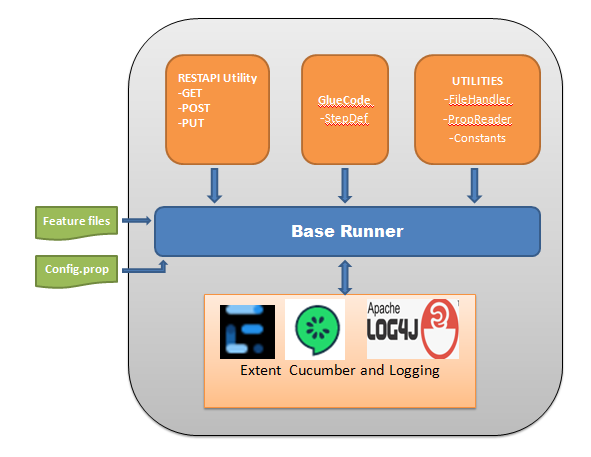
API Automation framework developed using **Java**+**RestAssured**. This provides built in BDD steps in order to create feature files as per user needs and provides extensive features in validating and handling of responses smoothly. It provides rich based reports in order to understand and analyze any issues for different stakeholders.

**Advantages of Cucumber**

* Cucumber supports different languages like Java,net and Ruby.
* It acts as a bridge between the business and technical language. We can accomplish this by creating a test case in plain English text.
* It allows the test script to be written without knowledge of code, it allows the involvement of non-programmers as well.
* It serves the purpose of end-to-end test framework unlike other tools.
* Due to simple test script architecture, Cucumber provides code reusability.

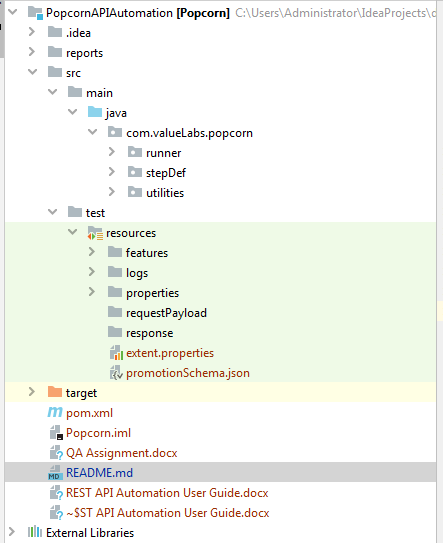
Architecture diagram and explanation

Below is the high level architecture diagram of this automation framework



Directory Structure

Below is the directory structure of the automation framework



**features** 🡪 directory has all the feature files

**logs** 🡪 will have all the log files for reference

**properties** 🡪 will have common.properties file which has all the configurations related to API endpoint and apiKey

**requestPayload** 🡪 in future if implemented will have POST payload json files

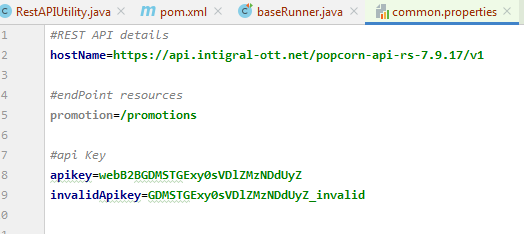
**response** 🡪 in future will have all responses in json saved under this directory

Configuration.properties

Configuration.properties is the file where all the configurations are handled such as

1. API hostname
2. Endpoint details
3. APIKey (Valid & Invalid)

Below is the screenshot for reference



Project Setup

Before going ahead with project setup, please follow the precondition step

**PreCondition:**

1. Make sure java is installed in the system
   1. Open command prompt and enter **java -version**, if prompts with valid version, proceed further with step 2, else go ahead with step 1.b
   2. Install java following below link

<https://www.oracle.com/in/java/technologies/javase-downloads.html>

1. Make sure maven is installed in the system
   1. Open command prompt and enter **mvn –version,** if prompts with valid version, proceed further with step 3, else go ahead with step 2.b
   2. Follow steps from below link

<https://maven.apache.org/install.html>

1. Make sure to have IDE installed in the system (either Eclipse or IntelliJ)

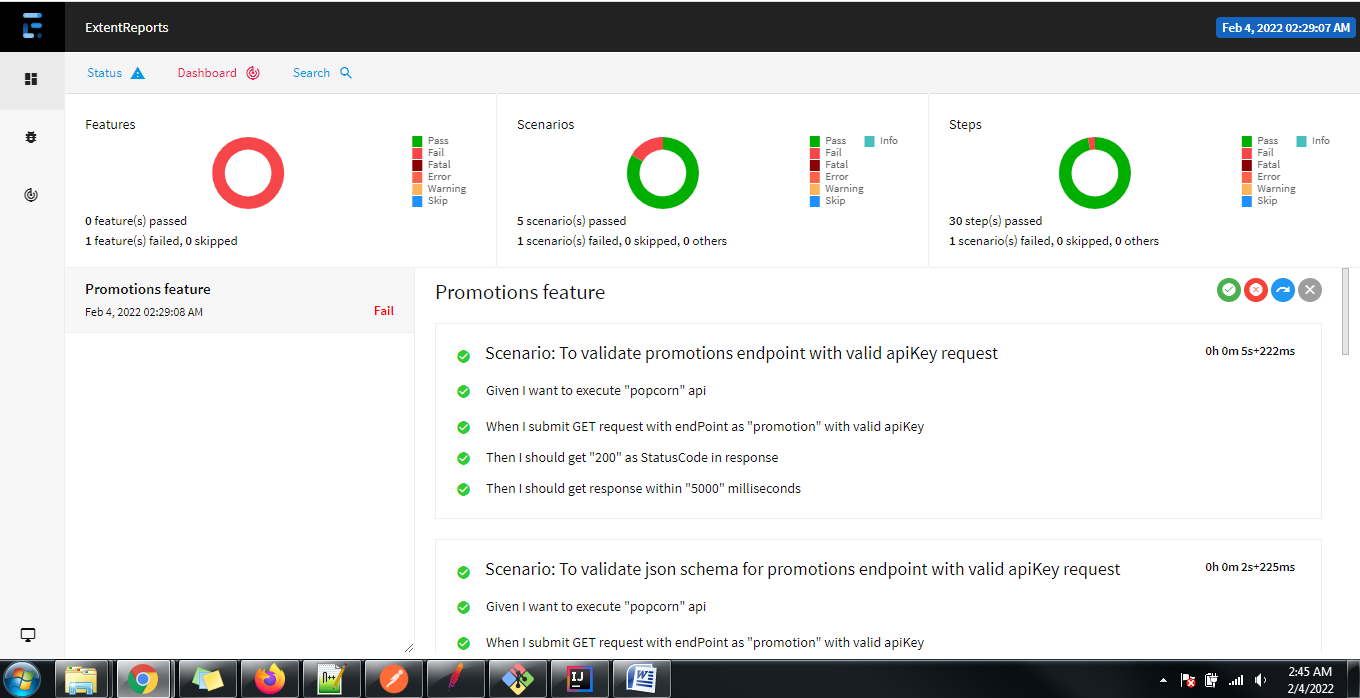
Below are the steps

1. Checkout the code from Git Repository
2. Launch IDE (Eclipse or IntelliJ) and click open existing project and select
3. From IDE terminal execute **mvn clean,** once build is successful navigate to step4
4. Open **baseRunner.java** file and rightClick run as 🡪 Run baseRunner
5. This will execute runner class invoking the feature files configured under them
6. Once after completion of the execution, reports will be generated in multiple forms along with logs
   1. Extent reports 🡪 reports/extent-cucumber-report.html
   2. Cucumber reports 🡪 target/cucumber/index.html
   3. Logs 🡪 src/test/resources/logs/popCorn.log

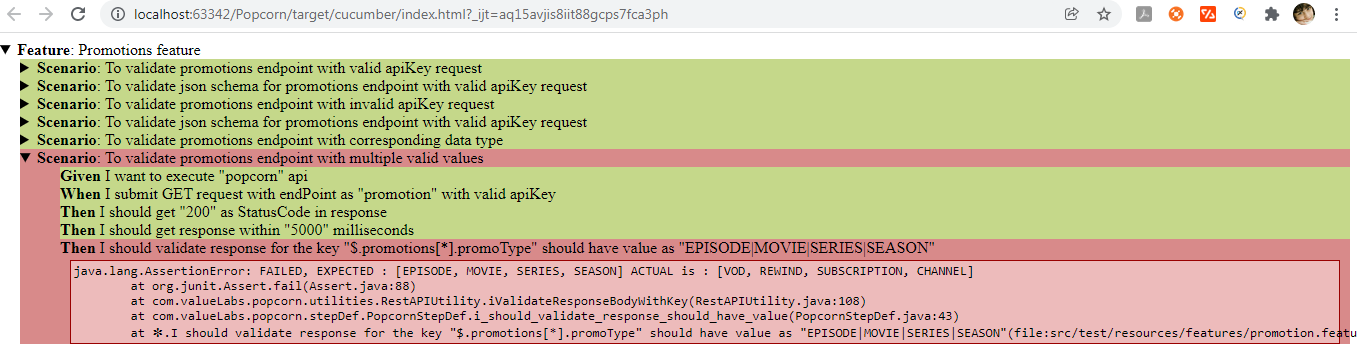
Reporting

Below are multiple forms of reports generated once the execution completes

1. Extent Report



1. Cucumber Report



Enhancements

Possible enhancements

1. Can be integrated to Jenkins
2. Enhance framework to include other CRUD operations