API Automation (E2E Tests)

Approach

1. I have used a hybrid framework which includes Cucumber + JAVA + Rest Assured + JUnit.

2. Benefits of this is it is very easy to read Scenario flow and easy to maintain.

Library used to automate REST API Calls

1. Rest Assured - 4.5.1

2. Cucumber - 7.2.3

3. Log4j - 1.2.17

4. Maven Cucumber Reporting - 5.7.0

5. JUnit - 4.13.2

6. JAVA - Open JDK 11.0.8

Programming language

1. I have chosen JAVA as a programming language.

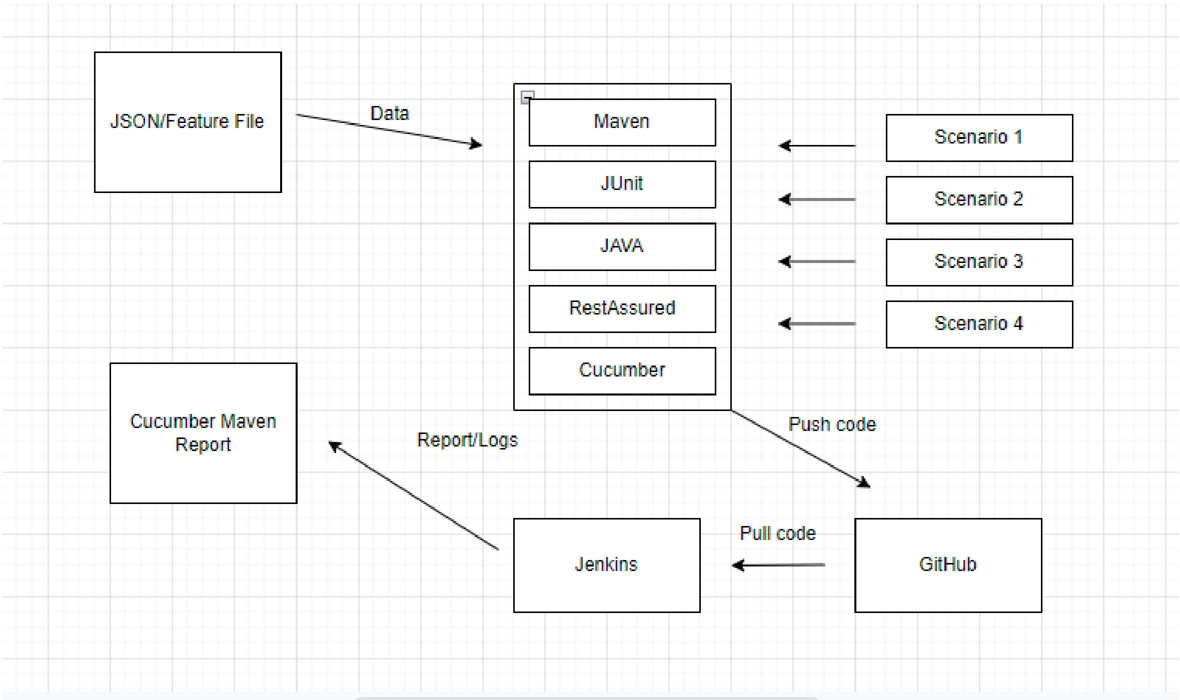
2. Java is the most widely used programming language and there are a lot of libraries available to read/write JSON, excel etc.

3. We can use the same programming language for **UI as well as API** Automation.

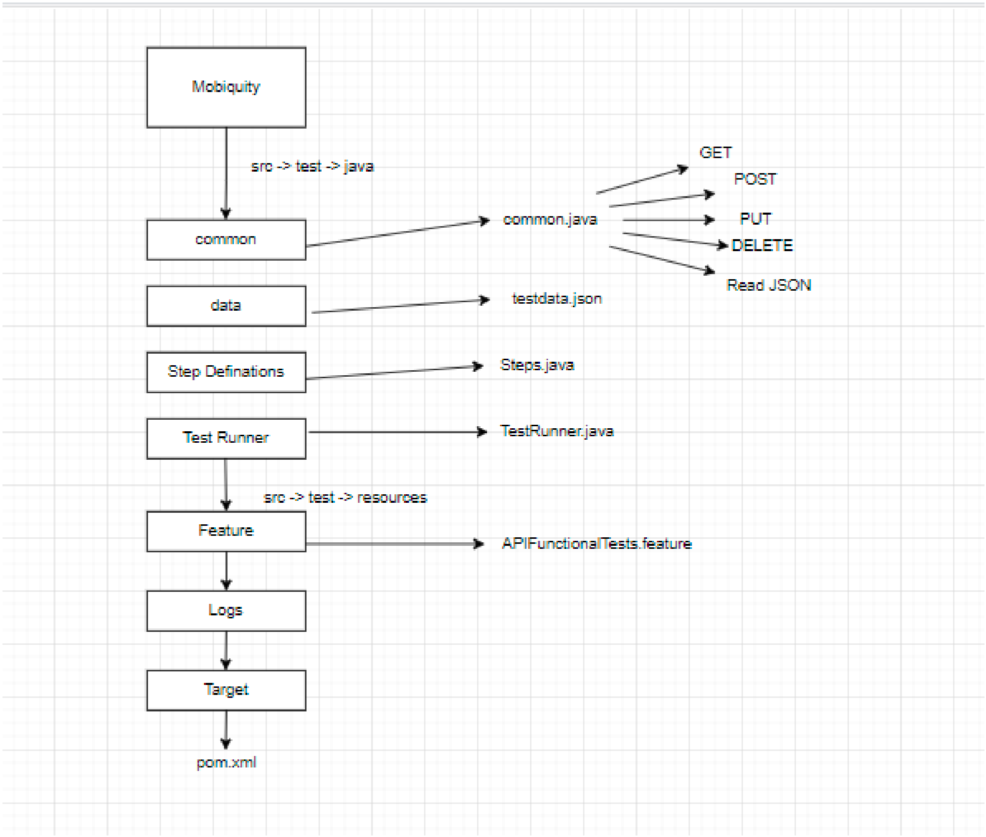
Unit testing framework

1. I have chosen **JUnit** as a unit-testing framework because it is most compatible with cucumber.

Architecture diagram -



Framework Diagram -



Build tool

I have chosen **Maven** as a build management tool and using this I can manage all dependencies required to run the project.

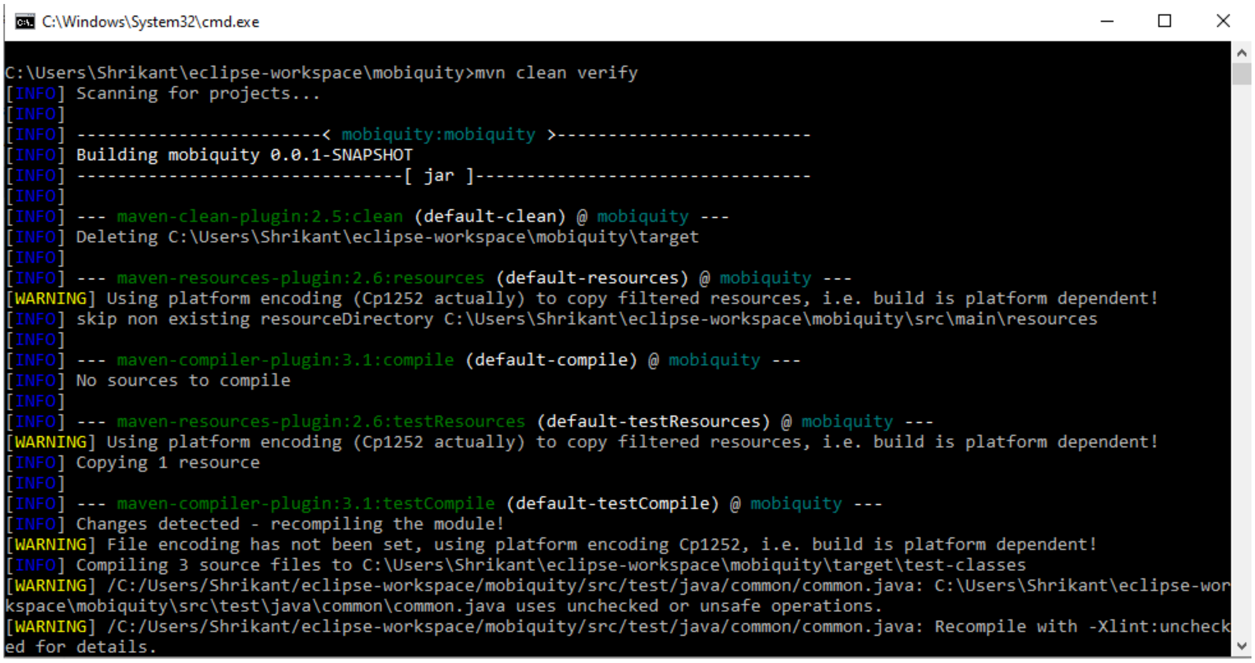
Logging Mechanism -

1. Use **Log4J** Java based library to generate automation logs.

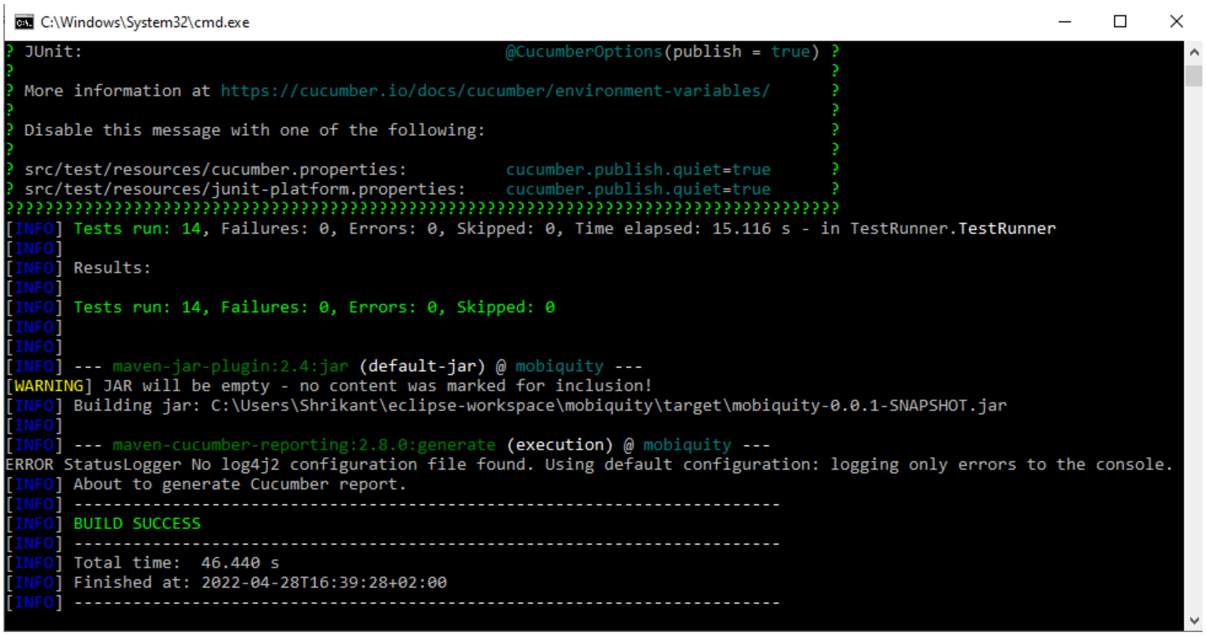
How to run

1. Clone the entire project on a local machine from GitHub.

2. Execute maven command from command-line as per below screen-shot.



```
C:\Users\Shrikant\eclipse-workspace\mobiquity>mvn clean verify
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< mobiquity:mobiquity >------------------------
[INFO] Building mobiquity 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ mobiquity ---
[INFO] Deleting C:\Users\Shrikant\eclipse-workspace\mobiquity\target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ mobiquity ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Shrikant\eclipse-workspace\mobiquity\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mobiquity ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ mobiquity ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ mobiquity ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 3 source files to C:\Users\Shrikant\eclipse-workspace\mobiquity\target\test-classes
[WARNING] /C:/Users/Shrikant/eclipse-workspace/mobiquity/src/test/java/common/common.java: C:\Users\Shrikant\eclipse-wor
kspace\mobiquity\src\test\java\common\common.java uses unchecked or unsafe operations.
[WARNING] /C:/Users/Shrikant/eclipse-workspace/mobiquity/src/test/java/common/common.java: Recompile with -Xlint:uncheck
ed for details.
```

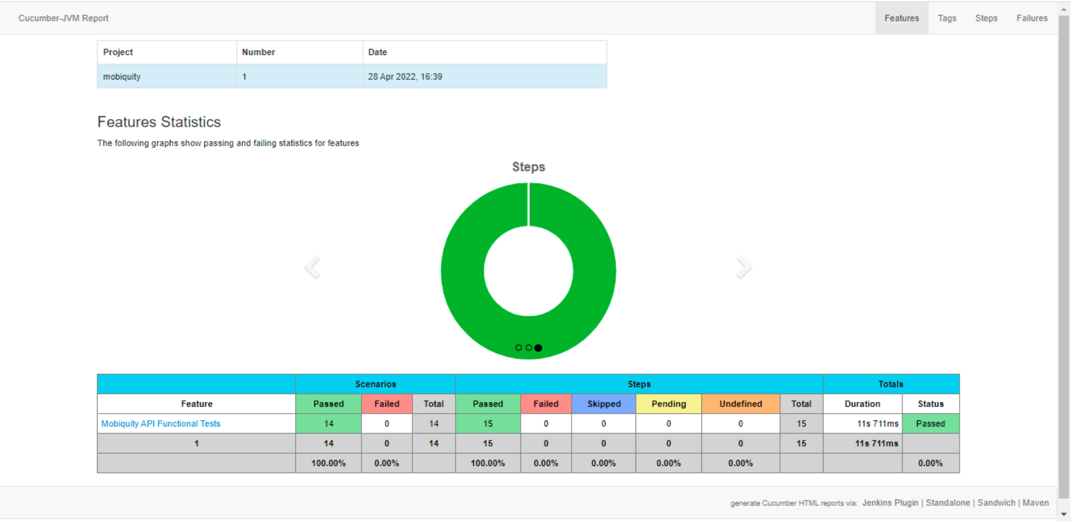


C:\Windows\System32\cmd.exe

```
? JUnit:                                           @CucumberOptions(publish = true) ?
?                                                                                   ?
? More information at https://cucumber.io/docs/cucumber/environment-variables/      ?
?                                                                                   ?
? Disable this message with one of the following:                                   ?
?                                                                                   ?
? src/test/resources/cucumber.properties:          cucumber.publish.quiet=true      ?
? src/test/resources/junit-platform.properties:    cucumber.publish.quiet=true      ?
?????????????????????????????????????????????????????????????????????????????????????
[INFO] Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 15.116 s - in TestRunner.TestRunner
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 14, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ mobiquity ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\Users\Shrikant\eclipse-workspace\mobiquity\target\mobiquity-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- maven-cucumber-reporting:2.8.0:generate (execution) @ mobiquity ---
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
[INFO] About to generate Cucumber report.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  46.440 s
[INFO] Finished at: 2022-04-28T16:39:28+02:00
[INFO] ------------------------------------------------------------------------
```

Reporting

2. HTML Reporting



Project	Number	Date
mobiquity	1	28 Apr 2022, 16:39

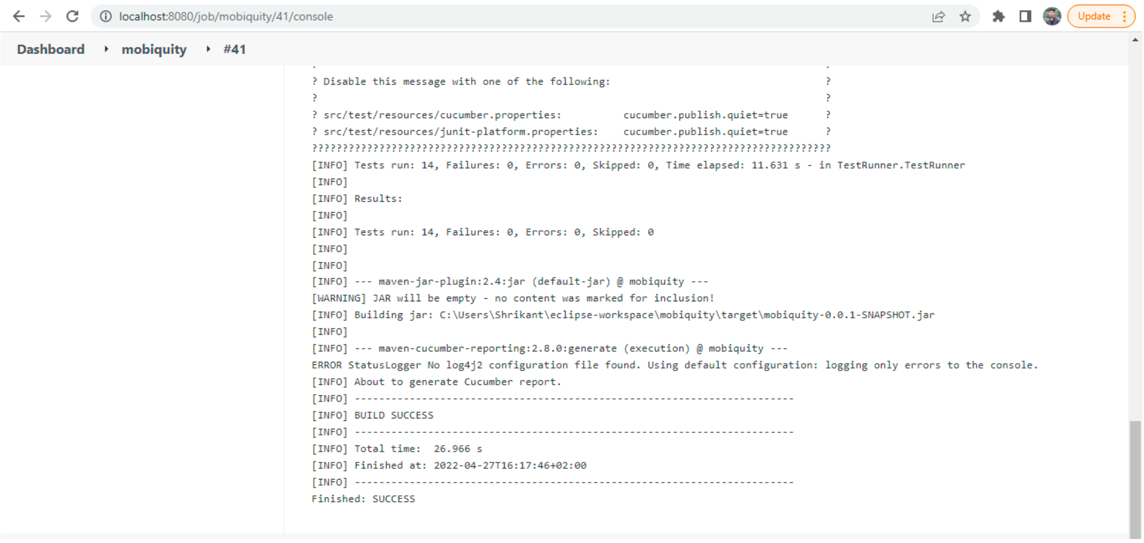
Features Statistics

The following graphs show passing and failing statistics for features

	Scenarios			Steps						Totals	
Feature	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Undefined	Total	Duration	Status
Mobiquity API Functional Tests	14	0	14	15	0	0	0	0	15	11s 711ms	Passed
1	14	0	14	15	0	0	0	0	15	11s 711ms	
	100.00%	0.00%		100.00%	0.00%	0.00%	0.00%	0.00%			0.00%

generate Cucumber HTML reports via: Jenkins Plugin | Standalone | Sandwich | Maven

CI/CD Execution Using Jenkins -



```
? Disable this message with one of the following:                             ?
?                                                                             ?
? src/test/resources/cucumber.properties:          cucumber.publish.quiet=true     ?
? src/test/resources/junit-platform.properties:    cucumber.publish.quiet=true     ?
?????????????????????????????????????????????????????????????????????????????????
[INFO] Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 11.631 s - in TestRunner.TestRunner
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 14, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ mobiquity ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\Users\Shrikant\eclipse-workspace\mobiquity\target\mobiquity-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- maven-cucumber-reporting:2.8.0:generate (execution) @ mobiquity ---
ERROR StatusLogger No log4j2 configuration file found. Using default configuration: logging only errors to the console.
[INFO] About to generate Cucumber report.
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  26.966 s
[INFO] Finished at: 2022-04-27T16:17:46+02:00
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

Current Automation Features -

1. Following the **BDD approach** using cucumber and gherkin language any non-technical user can maintain/execute tests.

2. **Data and test-cases** are separate; we use a Data driven approach easy to maintain.

3. Most of the things are **configurable** ex - URL etc.

4. **HTML reporting** anyone can understand automation reports.

5. Logging mechanism using **Log4J** is easy to debug.

6. We can run tests from the **IDE + Command line** as well.

7. Execute the same tests for **multiple API paths** with the same code base.

8. All dependencies imported at **runtime** using maven as a build tool.

9. Passes expected **status codes** from feature file only and it’s easily configurable.

10. Created **Reusable** methods ex - GET/POST/DELETE.

What features we can add to framework in next phase

1. Implement **parallel execution** to reduce overall execution time.

2. Integrate Automation with **Jenkins** for continuous integration.

3. Pull code at runtime from **GitHub** by Jenkins, execute, and send emailable reports to relevant audiences.

4. Create Automation dashboard to **monitor 24*7 Jenkins jobs** status.

5. Store automation reports on cloud ex – S3 bucket in AWS.

6. Manage most of Automation configuration as command-line arguments.

7. Separate Object Repository/Test Data on Cloud or any other Third Party tool for less maintenance.

8. Execute same scripts on different environments ex – Integration/Staging/Preview/Production etc.

9. Integrate Automation with different Third party tools ex – Slack for notification purpose / Integrate with JIRA to update automation results.