3.10 BDD with REST Assured Using Cucumber



This section will guide you to:

* Test an API by BDD with REST Assured using Cucumber

**Development Environment:**

* Eclipse IDE
* Java 1.8

This guide has four subsections, namely:

3.10.1 Creating a Maven project

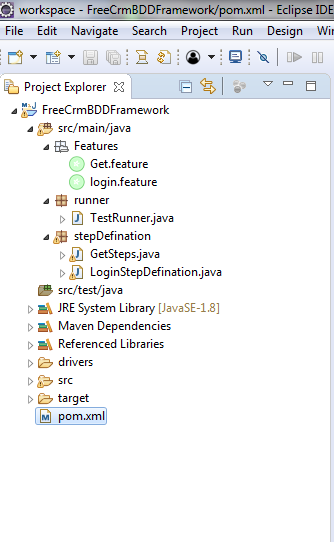
3.10.2 Updating the pom.xml file with the required dependencies

3.10.3 Testing an API BDD with REST Assured using Cucumber

3.10.4 Pushing the code to GitHub repositories

**Step 3.10.1:** Creating a Maven project

* Given below is the layout or example of a typical project structure:



* Open Eclipse.
* Open Eclipse Marketplace and install Natural.
* Restart Eclipse.
* Click on File---> New--->Project.
* Select the Maven project and click on Next.
* Enter the groupId and artifactId and click on Finish.

**Step 3.10.2:** Updating the pom.xml file with the required dependencies

* Open the pom.xml file.
* Add the dependencies given below to the pom.xml file:

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>rest-assured</artifactId>

<version>3.3.0</version>

<scope>test</scope></dependency>

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-java</artifactId>

<version>2.0.0</version></dependency>

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-junit</artifactId>

<version>3.0.0</version>

<scope>test</scope></dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>gherkin</artifactId>

<version>2.12.2</version>

<scope>provided</scope></dependency>

<dependency>

<groupId>info.cukes</groupId>

<artifactId>cucumber-jvm-deps</artifactId>

<version>1.0.5</version>

<scope>provided</scope></dependency>

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-jvm</artifactId>

<version>2.0.0</version>

<type>pom</type></dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope></dependency>

**Step 3.10.3:** Testing an API BDD with REST Assured using Cucumber

* Create a package **Features** inside the src/main/java directory.
* Create a file **Get.feature** inside the package **Features.**
* Write the code given below:

Feature: Verify GET operation by BDD with Rest Assured **using**

Cucumber

Scenario: Verify Employee Details by Get Request

Given Perform get operation **for** employee

And Perform get **for** mobile number

Then Veriyfy the status code

* Create a package **stepDefination** inside the src/main/java directory.
* Create a class **GetSteps.java** inside the package **stepDefination.**
* Write the code given below:

**package** stepDefination;

**import** cucumber.api.java.en.And;

**import** cucumber.api.java.en.Given;

**import** cucumber.api.java.en.Then;

**import** io.restassured.http.ContentType;

**import** **static** io.restassured.RestAssured.**\***;

**public** **class** GetSteps {

**@Given("^Perform get operation for employee$")**

**public** void getMethod() {

given().contentType(ContentType.JSON);

}

**@And("^Perform get for mobile number$")**

**public** void getdata() {

when()

.get("http://192.168.1.207:8080/api/employee/search/8095393564")

.then().statusCode(200);

}

**@Then("^Veriyfy the status code$")**

**public** void checkstatus() {

}

}

* Create a package **runner** inside the src/main/java directory.
* Create a class **TestRunner.java** inside the package **runner.**
* Write the code given below:

1. features= “created feature file location”
2. glue={“stepDefination”} package name of “GetSteps.java” class

**package** runner;

**import** org.junit.runner.RunWith;

**import** cucumber.api.junit.Cucumber;

**import** cucumber.api.CucumberOptions;

**@RunWith(Cucumber.class)@CucumberOptions**

**(features="C:\\Users\\Prakat-Intern\\Desktop\\Cucumber\\**

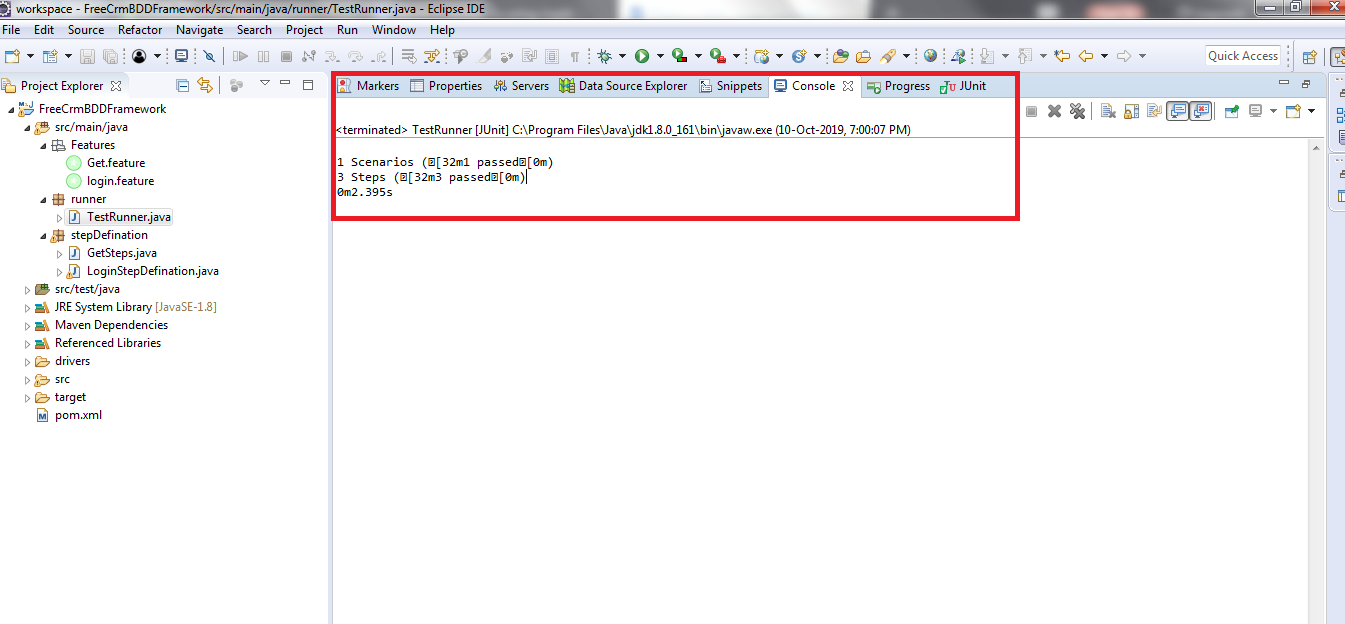
**workspace\\FreeCrmBDDFramework\\src\\main\\java\\Features**

**\\Get.feature",**

**glue= {"stepDefination"})public** **class** TestRunner {

}

* Right click on TestRunner class --> Run As --> JUnit Test.
* Verify the output in the console.



* Note: 1 Scenario and 3 steps are executed and all have passed.

**Step 3.10.4:** Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master