3.2 GET Request and Response Automation



This section will guide you to understand:

* How to automate GET request and response using REST Assured

**Development Environment:**

* Java 1.8
* Eclipse
* TestNG
* Maven

This guide has three subsections, namely:

3.2.1 Creating a Maven project in Eclipse

3.2.2 Executing the GET request and response program

3.2.3 Pushing the code to GitHub repositories

**Step 3.2.1:** Creating a Maven project in Eclipse

Get Method: Get method is used to retrieve data from the server at the specified resource.

* Open Eclipse.
* Click on file ---> click on New ---> Project.
* Select the Maven project and click on Next.
* Enter the Group id, Artifact id, and click on Finish.
* Right click on Project---> src/test/java---> Package.
* Enter the package name and click Finish.
* Right click on Package---> New---> Class.
* Enter the class name and click Finish.
* Add dependencies to pom.xml file.

**Step 3.2.2:** Executing the GET request and response program

* Write the program GET request using REST Assured and click on Save.

**package** GetResponse;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

import io.restassured.http.Method;

import io.restassured.response.Response;

import io.restassured.specification.RequestSpecification;

import junit.framework.Assert;

public **class** GetResponse1 {

@Test

void getempDetails()

{

//specify baseUrI

RestAssured.baseURI="http://192.168.1.207:8080/api/employee/search";

//Request object

RequestSpecification httpRequest=RestAssured.given();

//Response object

Response response=httpRequest.request(Method.GET,"/8095393564");

//print response in console window

String responseBody=response.getBody().asString();

         System.out.println("Response Body is:" +responseBody);

//status code validation

int statusCode=response.getStatusCode();

System.out.println("status code is :"+statusCode);

Assert.assertEquals(200,statusCode);

}

}

* Click on run and check the output in TestNG.

[RemoteTestNG] detected TestNG version 6.14.3

Response Body is:{"empId":3,"empName":"Sumit Kumar","empAddress":"Bellandur","mobileNumber":8095393564,"department":"development","project":"Cervical cancer application","teamLead":"RamaKrishnan","salary":10000.0,"joiningDate":"24-06-19"}

status code is :200

PASSED: getempDetails

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Failures: 0, Skips: 0

===============================================

**Step 3.2.3**: 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