3.3 POST Request and Response Automation



This section will guide you to understand:

* How to automate POST request and response using Rest Assured

**Development Environment:**

* Java 1.8
* Eclipse
* TestNG
* Maven

This guide has three subsections, namely:

3.3.1 Creating a Maven project in Eclipse

3.3.2 Executing the POST request and response program

3.3.3 Pushing the code to GitHub repositories

**Step 3.3.1:** Creating a Maven project in Eclipse

POST Request: POST requests are used to send data to the API server to create or update a resource. The data sent to the server is stored in the [request body](https://stackoverflow.com/questions/22034144/what-does-it-mean-http-request-body) of the HTTP request.

* 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 the pom.xml file.

**Step 3.3.2:** Executing the POST request and response program

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

**package** GetResponse;

import org.json.simple.JSONObject;

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** PostResponse {

@Test

void RegistrationSuccessful()

{

//specify base URI

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

//Request object

RequestSpecification httpRequest=RestAssured.given();

//contains the information in the json format

//Request payload sending along with post request

JSONObject requestParams=new JSONObject();

requestParams.put("empName","Lavanya");

requestParams.put("empAddress","Bomanahalli");

requestParams.put("mobileNumber","9900321102");

requestParams.put("department","testing");

requestParams.put("teamLead","Aruna");

requestParams.put("salary","10000");

requestParams.put("joiningDate","14-05-19");

httpRequest.header("Content-Type", "application/json");

httpRequest.body(requestParams.toJSONString());//attach above data to the request

//Response object

Response response=httpRequest.request(Method.POST,"/add");

//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:58

Status code is:200

PASSED: RegistrationSuccessful

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

Default test

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

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

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

Default suite

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

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

}

**Steps 3.3.3**: 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