3.6 SSL Authentication



This section will guide you to understand:

* SSL Authentication
* How to handle SSLPeerUnverifiedException in REST Assured

**Development Environment:**

* Eclipse IDE
* Java 1.8

This guide has four subsections, namely:

3.6.1 Creating a Maven project

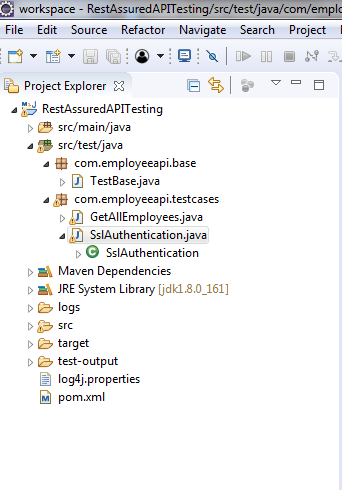
3.6.2 Updating the pom.xml file with required dependencies

3.6.3 Handling SSLPeerUnverifiedException using REST Assured

3.6.4 Pushing the code to GitHub repositories

**Step 3.6.1:** Creating aMaven Project

* The project structure looks like the screenshot below:



* 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.

**Step 3.6.2:** Updating the pom.xml file with required dependencies

* Open the pom.xml file.
* Add the given dependencies 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>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.14.3</version>

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

**Step 3.6.3:** Handling SSLPeerUnverifiedException using REST Assured

* Create a package “com.employeeapi.testcases” inside the src/test/java directory.
* Create a class“SslAuthentication.java” inside the package “com.employeeapi.testcases”.
* Write the below code:

**package** com.employeeapi.testcases;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.Test;

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

**import** **static** org.hamcrest.Matchers.**\***;

**import** io.restassured.RestAssured;

**public** **class** SslAuthentication {

/\*Suppose has invalid certificate and throwing an

SSLPeerUnverifiedException

\* so to handle this case we can relax certificate condition and now SSL

Exception will not come

\*

\* Do not have any proper url to test this feature\*/

**@Test**

**public** void testSsl() {

given().relaxedHTTPSValidation()

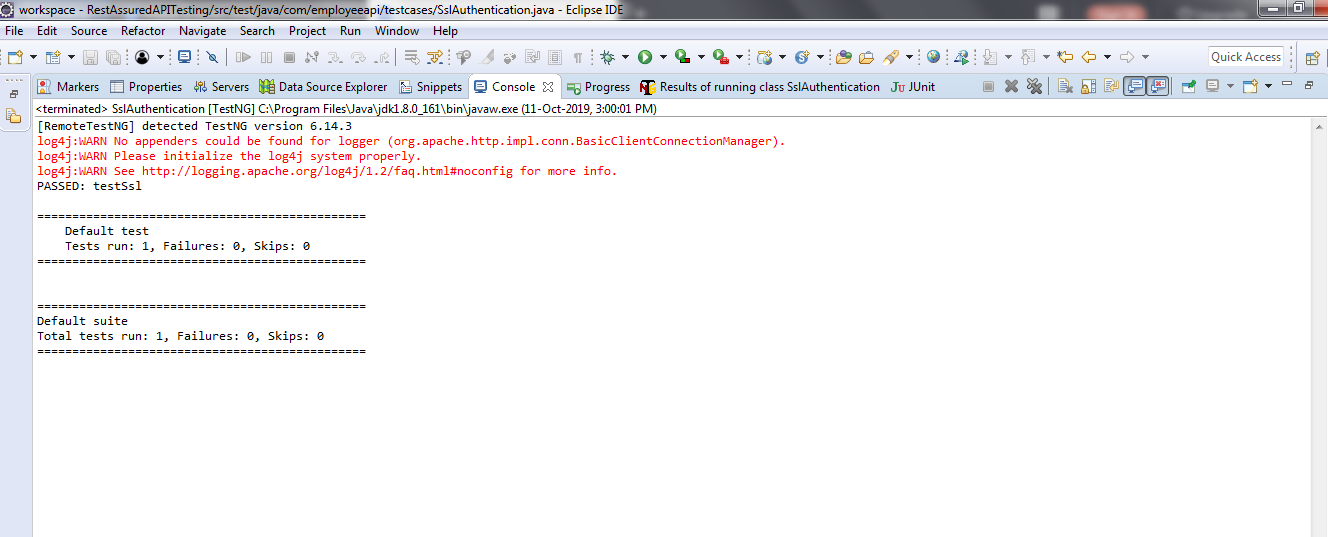
.when().get("http://www.bupa.com.au/")

.then().statusCode(200);

}

}

* Suppose the url has an invalid certificate then it will throw an SSLPeerUnverifiedException.
* To handle this case we can relax the certificate condition using relaxedHTTPSValidation() and then SSL Exception will not occur.
* Note: Do not have any proper url to test this feature.
* Right click on SSAuthentication class --> Run As --> TestNg Test.
* Verify the output from the Console:



* Note: testSsl is Passed.

**Step 3.6.4:** 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