**RestAssured**

**Q. What is RestAssured?**

* RestAssured is a Java library using which we can perform test automation on REST APIs.
* Only RESTful APIs are supported in RestAssured.
* We can validate both JSON and XML responses.
* Integrates well with Java Test Frameworks like Cucumber, TestNG, and Java dependency tools like Maven.
* By default, supports BDD-style Testing, so we don’t have to separately install Cucumber Plugin.

Pre-requisites:

1. Java + Eclipse
2. TestNG
3. Maven(Comes with Eclipse)

Dependencies:

1. rest-assured
2. json-path
3. json
4. gson
5. testng
6. scribejava-apis: For generating fake data
7. json-schema-validator
8. xml-schema-validator

TestNG and RestAssured Maven Dependencies to be added in the POM.xml file:

A screen shot of a computer program

Description automatically generated

**Testing approach**

**Q. What is the approach we are using to design the tests?**

A. We will be using BDD approach to design test cases using Gherkin syntax, i.e., using given, when, then templates.

i. given(): Content type, set cookies, add authentication tokens, add parameters, set header info. etc.

ii. when(): http requests including get,post,put,delete.

iii. then(): Validate status code, extract response, extract headers, cookies, response body etc.

A screen shot of a computer program

Description automatically generated

**Q. But instead of validating the responses, what if we want to capture the output of the HTTP response?**

A. In the case, we don’t use “then()” method for response validation but instead we use jsonPath().getInt/getString(“<propertyname>”) in the “when()” method.

A computer screen shot of text

Description automatically generated

**Q. What are the ways to create request body for POST, PUT, PATCH requests?**

A. 1. Hashmap: We can use this when we don’t have much data.

2. Using org.json library: We first have to add a maven dependency if we want to use ‘org.json’ jars. We create an object of the ‘JSONObject’ class and use the put(“<k>”,”<v>”) method like in Hashmaps and convert the object using <object>.toString()

3. Using POJO(Plain Old Java Object): We have to create a class with setters and getters for all the fields in the JSON request body file.

4. Using External JSON file: We have to create a new File, FileInputStream/FileReader, JSONTokener, JSONObject class objects and then use <object>.toString() on the final JSONObject.

**Path and Query Parameters**

**Q. How to give path and query parameters in the request?**

A. In the given() method, we can use the queryParams(“<key>”,”<value>”) method to give in the query params, and pathParams(“<endpoint path variable>”,”<value>”) method to give in the path parameter.

Then we have to use the path parameter variable in the then() method.

A screen shot of a computer code

Description automatically generated

**Cookies and Headers**

**Q. How to capture a specific cookie of a session?**

A.We first store the response of request in a variable and use the getCookie(“<cookie name>”) method on it.

A computer screen shot of code

Description automatically generated

**Q. How to capture all the session cookies?**

A. We first store the request response and then use the method getCookies() to store all the session cookies in a map.

A computer screen shot of a code

Description automatically generated

**Q. How to capture a specific header of a session?**

A.We first store the response of request in a variable and use the getHeader(“<header name>”) method on it.

A screen shot of a computer

Description automatically generated

**Q. How to capture all the session cookies?**

A. We first store the request response and then use the method getHeaders() to store all the response headers in a variable of “Headers” type.

A screen shot of a computer code

Description automatically generated

**Logging**

**Q. What if we want to log all, headers, cookies in the response?**

A. In the ‘then()’ method,

i. log().all(): To log everything in the response

ii. log().cookies(): To log only cookies in the response

iii. log().headers(): To log only headers in the response

**Parsing JSON Responses**

**Q. How to validate response codes, headers, json path values without using then()?**

A. First we store the JSON response in a variable,

1. For response codes, we use <response variable>.getStatusCode()

2. For response header, we use <response variable>.getHeader()

3. For response field value, we use <response variable>.jsonPath().get(“<Path to the the field>”).toString() or <response variable>.jsonPath().getString(“<Path to the field>”)

A screen shot of a computer program

Description automatically generated

**Q. How to get response field values using the “JSONObject” class?**

A. 1. We first create an object of the ‘JSONObject’ class using the response variable, then we can use the get(“<JSON Path>”).toString():

JSONObject <object> = new JSONObject(<response variable>.asString());

2. Using ‘for-loop’ and ‘getJSONArray’, ‘getJSONObject’ methods, we can iterate over the JSON file and get the values.

A computer screen shot of text

Description automatically generated

**Parsing XML Responses**

**Q. How to validate response codes, headers, json path values without using then()?**

**A.** First we store the XML response in a variable,

1. For response codes, we use <response variable>.getStatusCode()

2. For response header, we use <response variable>.getHeader()

3. For response field value, we use <response variable>.jsonPath().get(“<Path to the the field>”).toString() or <response variable>.xmlPath().getString(“<Path to the field>”)

A screen shot of a computer code

Description automatically generated

**Q. How to get response field values using the “XmlPath” class?**

**A.** 1. We create an object of the ‘XmlPath’ class using the response variable:

XmlPath <object> = new XmlPath (<response variable>.asString());

2. Using the getList(“<Multiple Tag Path>”) method we can get the list of objects in the multiple tag.

A screen shot of a computer code

Description automatically generated

**Authorization**

**Q. What types of authentication are supported in RestAssured?**

A. 1. Basic

2. Digest

3. API Key

4. Preemptive

5. Bearer Token

6. oauth 1.0,2.0

**Q. How to use authentications in RestAssured?**

A. In the “given()” method we can use “auth().basic(“<username>”,”<password>”)” or auth().digest(“<username>”,”<password>”) to pass in the auth. credentials.

A screen shot of a computer code

Description automatically generated

**Q. How to pass in API key authentication as k-v parameters along with other params?**

A. We use “queryParam(“<auth. key>”,”<auth. password>”)” along with other params in the “given()” method.

A screenshot of a computer code

Description automatically generated

**API Chaining**

GoRest

Create user -> Get user -> Update user -> Delete user

**Q. How to do API Chaining in RestAssured?**

A. To chain APIs, we capture the response output of one API to use it in another using:

<ITestContext Variable>.setAttribute(“<var\_name>”,<value>)

A screen shot of a computer screen

Description automatically generated

Then use <ITestContext Variable>.getAttribute(“<var\_name>”) in the other test where we are using the variable and type cast to whatever type the value that variable is returning:

A screen shot of a computer program

Description automatically generated

**RestAssured Test Framework Development**

**Framework:** Maintain all project-related files.

**Objectives:** Re-usabilility, maintainability, readability.

We will be designing a hybrid framework.

**Phases:**

1. Understanding requirements:
   1. Functional Specifications(static)
   2. Swagger
2. Choose Automation Tool: RestAssured Library
3. Design
4. Development
5. Execution + CI

A diagram of a software development process

Description automatically generated

**Pre-requisites:**

**Step-1:** Create a Maven Project.

**Step-2:** Update pom.xml file with required dependencies.

**Step-3:** Create folder structure.

**Step-4:** Create Routes.java class for containing all URLs.

**Step-5:** Create UserEndPoint.java class for CRUD methods implementation.

**Step-6:** Create Tests

**Step-7:** Create Data-driven test:

Excel Sheet -> For data

ExcelUtility -> Contains methods for getting Excel Data.

DataProviders -> Get data from excel sheet using TestNG annotation.

**Data-Driven Testing approach:**

1. We will pull-in the data from the excel sheet and use that data as the request body in a POST request to create a user.
2. Then only using the usernames from the excel sheet, using the DELETE request, we are going to delete the created users.

**Reports:**

**Step-8:** Generate extent reports.

Using ITestListener, we override the onStart, onTestSuccess, onTestFailure, onTestSkipped, onFinish and add the necessary code.

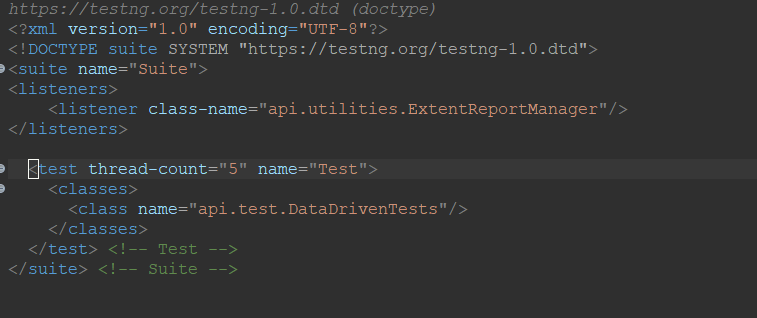
A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Also add the <listeners><listener class-name=”….”/></listeners> tag in the xml test config file.



**Logs:**

**Step-9:** Add Logs using log4j2 library.

i. Add dependency in pom.xml

ii. Add log4j2’s configuration log4j2.xml file in src/test/resources.