**Testing RESTful Web Services with Rest Assured**

Rest Assured is one of the most popular frameworks to test your RESTful Web services. When testing your RESTful web service with RA your test methods sends real Http requests. It can be Http GET, POST, PUT, or DELETE.

But it is going to be real HTTP request which contains actual behaviors and request parameters and when your get response back from a web service end point you can then validate with the JAVA code if your web service end point did return all the required information. You can make sure Http response contains valid Http headers and valid Http response body. For example you can validate that Http response contains authorization header or the body of Http response contains a valid JSON representation of user details and public user id value which is returned with the response is not empty and it is exactly 30 characters long for example.

Writing integration tests with Junit and RA will help you to automate your tests in work. You can run tests one by one or you can prepare series of tests and then run them all when needed to make sure that your API still works as expected.

You can make your test code send the Http request and then retrieve some details like for example access token or user Id from http response and use those details in the form of http request and when your API grows and becomes large testing your code with RA then Junit and automate test cases so much more efficient then every time testing your application using UI which is much more time consuming and you can always forget to test one of the cases.

We have created a RESTful web service application and during the development time we use post manage client to send the http request tour web service end points and see if they worked and if the response returned is correct and RESTful controller is inside of UI controller, is called UserController, it has different HTTP request mapping defined, we have get and post mapping to get user details, to create user details, to update user and so on.

So now we are ready to test the web service end points and we would like to automate our test a little bit so we don’t have to use curl or http postman or any other http clients to compose this http request manually and to send them manually.

We will try to use a Rest Assured framework which will allow us to test our RESTful web service endpoint programmatically and we will event try to automate them so that our tests run in sequence.

Currently our web service application is running on port 8080, and it is connecting to spring H2 consoleto our project so that we can use in-memory database rather than actual MySql database. Working with in-memory database is helping us not to use our MySQL database with test data. If we want to start fresh and clean we will simply restart our application and all the data we be cleared out and we can start with the correct one again.

So to test our RESTful we service API with Rest Assured framework, I will create a new project which will run as a standalone and I can use that standalone project to send the http request to my RESTful web service application running on localhost and if I want to send http request to a different URL like when I deploy my RESTful web API to an AWS server, I then can simply change the URL and if need port number and my test scripts will be testing a project already allocated on a different server.

so let’s create a new Maven project:

file -> new -> Sprint Starter Project

On the form: leave the Service UR no change,

Name : mobile-app-ws-rest-assured-test

Group: com.mycompanyname.app.ws

Package: com.mycompanyname.app.ws.restassuredtest

Click next

Now we will provide the dependencies for our project and for now no dependencies.

Click next

Click finish

Now the project is created.in this project we will have ‘MobileAppWsRestAssuredTestApplication’ will run our Mavin project and ‘MobileAppWsRestAssuredTestApplicationTests’ we are going to run our unit tests.

We need to add support for Rest Assured and Junit5 to our project:

Go to ‘<https://github.com/rest-assured/rest-assured/wiki/GettingStarted>’ and get the dependencies for JsonPath and XmlPath:

<dependency>

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

<artifactId>rest-assured</artifactId>

<version>4.3.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>json-path</artifactId>

<version>4.3.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>xml-path</artifactId>

<version>4.3.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>json-schema-validator</artifactId>

<version>4.3.0</version>

<scope>test</scope>

</dependency>

Now for Junit5, unfortunately Junit and Rest Assured are not compatible, so we have to do some advance setup:

First add ‘Junit Jupiter engine’ from Maven repo web site

Then add ‘Junit Platform Launcher’ from maven repo web site

Also look at the pom.xml file that added some more dependencies as well …

To create a test cases, we are going to ***mobile-app-ws-rest-assured-test -> src/test/java -> com.mycompanyname.app.ws.restassuredtest***, Then create a new Junit Test Case:

Select New Junit Jupiter test

Name: TestCreateUser

Leave the rest as default

Click finish.