**REST-ASSURED\_ BDD**

Java –jar Prathap.jar –server.port=8086

**API:**

**Application Programming Interface.** It is in JASON, XML etc..

**REST:**

**Representaional State Transfer**

It make a call from a client to server to get data back over the HTTP Protocol.

We get API s from Url.

**REST ASSURED:**

* Rest Assured is Open source Java Domain Specific Language (DSL)
* It is built on top of HTTP Client, Internally it use classes, methods, interfaces.
* It supports both XML and JSON format.

**REST- Assured Supports Other Requests:**

* POST **creates resources in server**
* GET : It will give you data
* PUT
* DELETE
* OPTIONS
* PATCH
* HEAD
* **Get, Delete, Put will not create additional load on server while making multiple calls.**
* **POST: This can create load on server if made multiple calls.**

**Dependencies:**

* REST Assured
* Test-NG
* JSON Path
* JSON Schema Validator
* XML PATH
* Java-HamCrest
* HamCrest-Core
* HamCrest-Library
* Don’t Forgot to add “Jackson dataBined” dependency.

**In REST Assuresed we have different styles of writing test, Here we use BDD.**

**Intialization after “given”**

In BDD if you want set cookies, add authorization, parameters, or setting headers all we do after given sections.

In when, we get resource, or we consume resource.

**All actions after “When”**

Get,Post,put,delete,etc..

**Response after “Then”**

Validate status code, extract response, extract headers, cookies, extract response body.

**Methods (Requests): (In Eclipse StudentApp (RestAssured)**

**1) GET:**

**This Method Simply retrieves the data from server. No changes to server or resource.**

**EX:** Checking news, Searching on web ..

This action method we use for get data from file, its always after “When”.

Ex: .get(“/list”);

**Code:**

@Test

**public** **void** getstudentfrommed()

{

Response response3= *given*()

.param("programme", "Medicine")

.param("limit",1)

.when()

.get("/list");

System.***out***.println(response3.prettyPrint());

**2) POST:**

POST: This can create load on server if made multiple calls.

This Method perform the changes to server, In simple words post always creates resources in server.

Ex: uploading pic, submitting application online .. etc..

It is NOT idempotent, so if you retrey the reqst N times, you will end up having N resources with N different URLs created on server.

**Terminology:**

“When U hit URI”

“When you send this Payload”

“When You make post reqst”

We can do this by differentways:

* By JSON object
* By Hash Map
* By File
* BY adding data into Body

**In Code:**

**Create class for post**

**Write body for Jason/http**

**Create new package in src/main/java**

**Create new class(Pojio) in above package**

**Select Jason body**

**Right click source**

**Generate set setters getters**

**Select all fields, and select insertion point(after this seeters and getters will come)**

**3) PUT:**

**This Method update the existing resources. In simple words this method can update the existing file on server.**

**If we do same operation 2 times it will over ride. It won’t create additional load to server.**

**4) DELETE:**

**This method simply deletes data or resources from the server.**

**Ex: Deleting files from cloud, delete pics from fb etc..**

**REST API FRAMEWORK SETUP**

* We have to create project
* In src/main/java we have to create “Config.properties file”
  + - In this we give any URL
    - URI
    - Username and password etc.
* Then create a “Test base class”
  + - Here we have to create constructor of this class to call /read “config.properties file”
* Then Create “REST client” class
  + - In this we create C R U D methods