**Test Plan for REST-API**

* **Automated Test Guidelines**
  + Utilizing RestAssured Library using Java in IntelliJ IDE for scripting the test cases in RestAPI.
  + Maven as a build tool: All the necessary libraries are included along with the required dependencies and plugins
  + TestNG tool and its annotations for executing the steps for each test case
  + Matchers – Hemcrest chaining for assertions for validations of API calls: response status code, Json response body, parameters,
  + Framework should be reusable for other tests of the features within the project (Utility package including Fakers, POJO for serialization and deserialization)
  + Both positive and negative testing of API

**---------------------------------------------------------------------------------------------------------------------**

**Test Cases**

* Following verifications are made to validate the accurate output of API calls in reponses:

1. **Verify correct HTTP status code.** For example, creating a resource should return 201 CREATED and unpermitted requests should return 403 FORBIDDEN, etc.
2. **Verify response payload.** Check valid JSON body and correct field names, types, and values — including in error responses.
3. **Verify response headers.** HTTP server headers have implications on both security and performance.

**Case 1:** Creating a new User

POST: /createUser

Data:

Json body: {

"name": "brhmdmr",

"email": "brhmd@yahoo.com"

}

@Test

*Given accept type is Json*

*When user sends post request to Endpoint URI: "* [*http://localhost:8080/api/v1/users*](http://localhost:8080/api/v1/users)*"*

*And response body is in Json format*

*Then response status code should be 204*

*Then response body should match with the request body*

**Case 2:** Validating ID with GET Request

GET: /getUser/{some\_id}

@Test

*Given accept type is Json*

*When user sends get request to Endpoint URI: "* [*http://localhost:8080/api/v1/users/{some\_id*](http://localhost:8080/api/v1/users/%7bsome_id)*}"*

*And response body should be in Json format*

*Then response status code should be 200*

*Then validate the response body*

***Case 3:*** Validating All Users with Get Request

GET: /getUsers

@Test

*Given accept type is Json*

*When user sends get request to Endpoint URI: "*[*http://localhost:8080/api/v1/users*](http://localhost:8080/api/v1/users) *"*

*And response body should be in Json format*

*Then response status code should be 200*

*Then validate the response body*

***Case 4:*** *Updating a User with Put Request*

Put: /{some\_id}/update

@Test

*Given accept type is Json*

*When user sends put request to Endpoint URI: "*[*http://localhost:8080/api/v1/users*](http://localhost:8080/api/v1/users){some\_id}*"*

*And response body should be in Json format*

*Then response status code should be 204*

*Then response body should match with the request body*

**Case 5:** Delete a User

Delete: /{some\_id}/delete

*Given accept type is Json*

*When user sends delete request to Endpoint URI: "* [*http://localhost:8080/api/v1/users/{some\_id*](http://localhost:8080/api/v1/users/%7bsome_id)*}"*

*And response body should be in Json format*

*Then response status code should be 404*