**Complex response Validation**

In order to verify the complex response, we should always go for JssonPath() .get(jsonXpath), used to navigate specific location of the response then capture the data for validation.

There are two types of Response

1. Static Response
2. Dynamic Response

* **Static Response:** response body content is always fixed**,** so that we can write JsonPath using index, it always returns single Value

@Test

**public** **void** verifyPRojectNameInStaticResponse() {

String expectedData = "ICICI";

Response resp = *when*()

.get("http://localhost:8084/projects");

String actProjectNAme = resp.jsonPath().get("[0].projectName");

resp.then()

.log().all();

Assert.*assertEquals*(actProjectNAme, expectedData);

}

* **Dynamic Response:** response body content is dynamic in nature**,** so that we can write JsonPath without index, & it always return List of values (Collection)

@Test

**public** **void** verifyPRojectNameinDynamicResponse() {

String expectedData = "zoho";

Response resp = *when*() .get("http://localhost:8084/projects");

List<String> list = resp.jsonPath().get("projectName");

**for**(String str : list) {

**if**(str.equals(expectedData)) {

System.***out***.println(expectedData + "==> is available in Dynamic Response");

**break**;

}

}

resp.then().log().all();

}

**Parameters:**

**Path Parameter:**

**package** com.rmgyantra.Parmeters;

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

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

**public** **class** PathParameterTest {

@Test

**public** **void** pathParameterTest()

{

*given*()

.pathParam("proId", "TY\_PROJ\_6802")

.when()

.get("http://localhost:8084/projects/{proId}")

.then()

.log()

.all();

}

}

**Query Parameter:**

**package** com.rmgyantra.Parmeters;

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

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

**public** **class** QueryParameterTest {

@Test

**public** **void** queryParameter()

{

*given*()

.queryParam("page", 2)

.when()

.get("https://reqres.in/api/users")

.then()

.log()

.all();

}

}

**package** com.rmgyantra.Parmeters;

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

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

**public** **class** GitHubParametersTest {

@Test

**public** **void** gitHubParametersTest()

{

*given*()

.pathParam("username", "Soumyasantasahoo2013")

.queryParam("sort", "created")

.when()

.get("https://api.github.com/users/{username}/repos")

.then()

.log()

.all();

}

}

**Request Chaining:**

Fetch the value from one response API & pass same data in to another Request is called request chaining

@Test

**public** **void** deleteProject() {

Response resp = *when*()

.get("http://localhost:8084/projects");

String firstProjectID = resp.jsonPath().get("[0].projectId");

*given*()

.pathParam("projectID", firstProjectID)

.when()

.delete("http://localhost:8084/projects/{projectID}")

.then()

.log().all();

}

}