**POST for create a register**

* **Create a util class for generate the random data. Doesn't want to do hardcore.**
* **import org.apache.commons.lang3.RandomStringUtils;**
* **create all static methods here and when I call this method every time its generate new things**

**public static String getFirstName() {**

**String generatedString = RandomStringUtils.randomAlphabetic(1);**

**return ("John"+generatedString);**

**}**

**Declare a Test Class and generate HashMap Object for store multiple values**

**public static HashMap map=new HashMap ();**

**@BeforeClass**

**public void postdata()**

**{** **map.put("FirstName",RestUtils.getFirstName());**

**map.put("LastName", RestUtils.getLastName());**

**map.put("UserName", RestUtils.getUserName());**

**map.put("Password", RestUtils.getPassword());**

**map.put("Email", RestUtils.getEmail());**

**RestAssured.baseURI="http://restapi.demoqa.com/customer";**

**RestAssured.basePath="/register";**

**}**

**@Test**

**public void testPost() {**

**given() .contentType("application/json").body(map)**

**.when().post()**

**.then().statusCode(201).and()**

**.body("SuccessCode",equalTo("OPERATION\_SUCCESS"))**

**.and()**

**.body("Message",equalTo("Operation completed successfully"));}**

**PUT FOR UPDATE EMPLOYEE RECORD**

**public class Demo3\_PUT\_Request {**

**public static HashMap map=new HashMap();**

**String empName=RestUtils.empName();**

**String empSalary=RestUtils.empSal();**

**String empAge=RestUtils.empAge();**

**int emp\_id=11501;**

**@BeforeClass public void putData() {**

**map.put("name",empName);**

**map.put("salary",empSalary);**

**map.put("age",empAge);**

**RestAssured.baseURI="http://dummy.restapiexample.com/api/v1";**

**RestAssured.basePath="/update/"+emp\_id;}**

**@Test**

**public void testPUT() {**

**given().contentType("application/json").body(map)**

**.when().put()**

**.then().statusCode(200).log().all(); }**

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

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

**import static org.hamcrest.Matchers.\*;**

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

**public class Demo4\_DELETE\_Request {**

**@Test public void deleteEmployeeRecord() {**

**RestAssured.baseURI="http://dummy.restapiexample.com/api/v1";**

**RestAssured.basePath="/delete/11500";**

**Response response= convert response to Json**

**given().when().delete().then()**

**.statusCode(200).statusLine("HTTP/1.1 200 OK")**

**.log().all().extract().response();**

**String jsonAsString=response.asString();**

**Assert.assertEquals(jsonAsString.contains("successfully! deleted Records"),true);}**

**1) Test Status Code**

**2) Log Response**

**3) Verifying Single content in response body**

**4) Verifying Multiple contents in response body**

**5) Setting parameters & headers**

**Public void testStatusCode(){**

**given()**

**.param(“MyName”, “Nurul”)**

**.header(“MyHeader”, “USA”)**

**.when()**

**.get(“http://jsonplaceholder.typicode.com/posts/5”)**

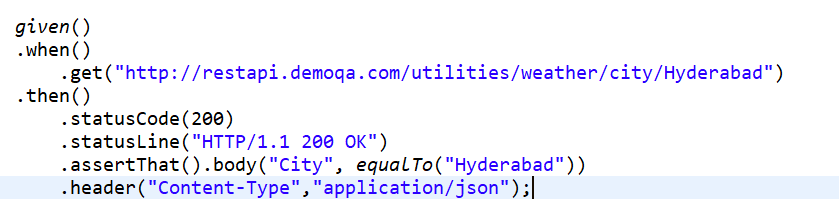
**.then()**

**.statusCode(200)**

**.log().all(); JsonPathFinder**

**.body(“RestResonse.result.name”, equalTo(“India”));**

**.body(“RestResonse.result.name”, hasItems(“India”,BD));**

****

**READ RESPONSE**

**String response = get(/weather/city/Hyderabad").asString();**

**System.out.println("My Response is:"+response);**

**@Test(priority=2) 🡺Extract Specific Value**

**public void testExtractDetailsUsingPath() {**

**String href = given().when()**

**.get("http://jsonplaceholder.typicode.com/photos/1")**

**.then().contentType(ContentType.JSON)**

**.body("albumId", is(1)).extract().path("url");**

**System.out.println("Extracted URL value is:"+href);**

**String href1=get(photos/1").path("thumbnailUrl");**

**System.out.println("Fetched Thumbnail URL1 : --->"+href1);**

**String href2 = get(/1").andReturn().jsonPath(). getString("Url");**

**System.out.println("Fetched Thumbnail URL2 : --->"+href2);**

**when().get(href1).then().statusCode(200); }**

**//Extract details as Response for Further use**

**Response response=given().when()**

**.get("http://jsonplaceholder.typicode.com/photos/1")**

**.then().extract().response();**

**System.out.println(response.contentType());**

**System.out.println(response.path("title"));**

**System.out.println(response.getStatusCode());**

**1) Verifying Single content in XML Response**

**2) Verifying Multiple contents in XML Response**

**3) Verifying all the content in XML response in one go**

**4) Find values using XML Path in XML response**

**5) XPath with text() function**

**//1) Verifying Single content in XML Response**

**@Test(priority=1) void testSingleContent(){**

**given().when().get("http://thomas-bayer.com/sqlrest/CUSTOMER/15/")**

**.then().body("CUSTOMER.ID",equalTo("15")).log().all();}**

**//2) Verifying Multiple contents in XML Response**

**@Test(priority=2) void testMultipleContents(){**

**given().when().get("http://thomas-bayer.com/sqlrest/CUSTOMER/15/")**

**.then().body("CUSTOMER.ID",equalTo("15"))**

**.body("CUSTOMER.FIRSTNAME",equalTo("Bill"))**

**.body("CUSTOMER.CITY",equalTo("Seattle"));}**

**//3) Verifying all the content in XML response in one go**

**@Test(priority=3)id testMultipleContentsInOneGo() {**

**given().when().get("http://thomas-bayer.com/sqlrest/CUSTOMER/15/")**

**.then().body("CUSTOMER.text()",equalTo("15BillClancy319 Upland Pl.Seattle"));**

**// Find values using XML Path in XML response**

**@Test(priority=4) void testUsingXPath1(){**

**given().when()**

**.get("http://thomas-bayer.com/sqlrest/CUSTOMER/15/")**

**.then()**

**.body(hasXPath("/CUSTOMER/FIRSTNAME",containsString("Bill")));**

**.log().all();**