Phase-3 Practice Project: Assisted Practice

2.Write a program to demonstrate automation of GET request and response.

```
package phase3_Lesson_end_project;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class GetDemo {

            @Test

            public void VerifyMessage()

  RestAssured

  .given()

  .contentType("application/json")

  .header("Authorization", "Bearer
44292b35ce5612b1c16008ee8c47e256e5de9e0caeb8cdd653d167f9ca93a800 ")

  .when()

  .get("https://reqres.in/api/users/2")

  .then()

  .statusCode(200)

  .log().all();

  }

}
```

3.Write a program to demonstrate automation of POST request and response.

```
package rest_assured_demo_testscript;

import java.util.HashMap;

import org.testng.annotations. BeforeMethod;

import org.testng.annotations.Test;

import io.restassured. RestAssured;
```

```
public class POSTdemo {


    HashMap<String, String> map = new HashMap<String, String>();


    @BeforeMethod


    public void createPayLoad() {
        map.put("name", "morpheus");
        map.put("job", "painter");
        RestAssured.baseURI = "https://reqres.in/";
        RestAssured.basePath = "/api/users";


    }


    @Test
    public void createResource() {


        RestAssured
        .given()
        .contentType("application/json")
        .body (map)
        .when()
        .post()


        .then()
        .statusCode (201)
        .log().all();
    }
}
```

4.Write a program to send XML payload in REST assured, parse XML response in REST assured, parse JSON response in REST assured, and explore native logging of REST assured.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CapitalCity xmlns="http://www.oorsprong.org/websamples.countryinfo">
      <sCountryISOCode>US</sCountryISOCode>
    </CapitalCity>
  </soap:Body>
</soap:Envelope>
```

```java
package rest_assured_demo_testscript;

      import org.testng.annotations. Test;

      import io.restassured. RestAssured;

      public class GETdemo {

      @Test

      public void getServerResponse() {

      RestAssured

      .when()

      .get("https://restcountries.com/v3.1/all")

      .then()

      .assertThat()

      .statusCode (200)

      .log().all();

}

      }
```

5.Write a program to demonstrate GET, POST, XML, and JSON in REST Assured.

```
package rest_assured_demo_testscript;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class GetBearerTokenDemo {

        @Test

        public void VerifyMessage() {

  RestAssured

  .given()

  .contentType("application/json")

  .header("Authorization", "Bearer 44292b35ce5612b1c16008ee8c47e256e5de9e0caeb8cdd653d167f9ca93a800 ")

  .when()

  .get("https://gorest.co.in/public/v2/users/4589310")

  .then()

  .statusCode(200)

  .log().all();

  }

}
```

```
package rest_assured_demo_testscript;

import java.util.HashMap;

import org.testng.annotations. BeforeMethod;

import org.testng.annotations.Test;

import io.restassured. RestAssured;

public class POSTdemo {

        HashMap<String, String> map = new HashMap<String, String>();

        @BeforeMethod

        public void createPayLoad() {

                map.put("name", "morpheus");

                map.put("job", "painter");

                RestAssured.baseURI = "https://reqres.in/";
                RestAssured.basePath = "/api/users";
        }
        @Test
        public void createResource() {
                RestAssured
                .given()
                .contentType("application/json")
                .body (map)
                .when()
                .post()
                .then()
                .statusCode (201)
                .log().all();
        }
}
```

6.Write a program to demonstrate OAuth authorization in REST API using REST assured.

```
package rest_assured_demo_testscript;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class ApiKeyDemo {
        @Test
 public void GetwheatherInfo(){
        RestAssured
        .given()
        .param("q","chennai")
        .param("appid","2f15b0744ac5f3aa170bf81ec4803102")
        .when()
        .get("https://api.openweathermap.org/data/2.5/weather")
        .then()
        .statusCode(200)
        .log().all();
 }
}
```

8.Write a program to configure Log4j in Eclipse Maven project.

```
package phase3_Lesson_end_project;
import org.apache.log4j.Level;
import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;
import org.testng.annotations.BeforeClass;
import io.restassured.response.Response;
import io.restassured.specification.RequestSpecification;

public class TestBase {
	public static RequestSpecification httpRequest;
	public static Response response;
	public Logger logger;
	@BeforeClass
	public void setUp() {
		logger = Logger.getLogger("LogDemo");
		PropertyConfigurator.configure("log4j.properties");
		logger.setLevel(Level.DEBUG);

}
}
```

9.Write a program to generate logs on Eclipse Console using Log4j.

```
package rest_assured_demo_testscript;
import static org.junit.Assert.assertThat;
import java.util.HashMap;
import java.util.UUID;
import org.apache.log4j.Logger;
import org.testng.Assert;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;
import static org.hamcrest.Matchers.is;

public class Create_Delete_update_resource_Demo extends TestBase {
        HashMap<String, String> map = new HashMap<String, String>();
        UUID uuid= UUID.randomUUID();
        int id;
        Response response;
        JsonPath jsonPath;
        @Test(priority=0)
        public void createPayload() {
        map.put("name", "Masha");
        map.put("email", uuid+"@gmail.com");
        map.put("gender", "male");
        map.put("status", "Active");
        RestAssured.baseURI = "https://gorest.co.in";
        RestAssured.basePath = "public/v2/users";
        logger.info("Payload created");

        }
        @Test(priority=1)
        public void createResource() {
                 response =RestAssured
         .given()
        .contentType("application/json")
        .body(map)
        .header("Authorization", "Bearer
77bf8479c2dcb452437e808e06b3bb45fe398da27c56ce2c337598237f64cc0f")
        .when()
        .post()
        .then()
        .log().all()
        .extract().response();
       logger.info("Resource Created Successfully");
         jsonPath = response.jsonPath();
        id =jsonPath.get("id");
        System.out.println(id);
        }

        @Test(priority=2)
        public void verifyResource() {
        System.out.println("Id ]= "+id);
        RestAssured
.given()
.contentType("application/json")
.body(map)
.header("Authorization", "Bearer
77bf8479c2dcb452437e808e06b3bb45fe398da27c56ce2c337598237f64cc0f")
.when()
        .get("https://gorest.co.in/public/v2/users/"+id)
        .then()
        .statusCode(200)
   .log().all();
```

```
    Assert.assertTrue(jsonPath.get("name").equals("Masha"));
    }
@Test(priority=3)
public void updateResource() {
    map.put("name", "Mashaa");
    map.put("email", uuid+"@gmail.com");
    map.put("gender", "male");
    map.put("status", "Active");
    RestAssured.baseURI = "https://gorest.co.in";
    RestAssured.basePath = "public/v2/users/"+id;
    RestAssured
    .given()
    .contentType("application/json")
    .header("Authorization", "Bearer
77bf8479c2dcb452437e808e06b3bb45fe398da27c56ce2c337598237f64cc0f")
    .body(map)
    .when()
    .put()
    .then()
    .statusCode(200)
      .log().all()
    .assertThat()
        .body("name", is("Mashaa"));
        }
@Test(priority=4)

public void deleteResource() {
    RestAssured.baseURI = "https://gorest.co.in";
    RestAssured.basePath = "public/v2/users/"+id;

    RestAssured
    .given()
```

```
        .contentType("application/json")

        .header("Authorization", "Bearer 77bf8479c2dcb452437e808e06b3bb45fe398da27c56ce2c337598237f64cc0f")

        .when()

        .delete()

        .then()

        .statusCode(204);

}}
```