

Handling the Test Data and Executing It in Multiple Environments for User Module of an Application

Description

To create a framework for validating the test data using the web services for the authentication module

Background of the problem statement:

As a part of developing an authentication module for healthcare platform web applications, the back-end admin requires a module that can store the details of users and perform registration and login operations.

You must use the following:

- Eclipse IDE
- Java 1.8
- TestNG
- Maven

Following requirements should be met:

- Some of the source code should be tracked on the GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.
- The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link to the repository in the document.
- The step-by-step process involved in completing this task should be documented.

URL Endpoint:

Base URL: <https://reqres.in/api>

Register User URL: <https://reqres.in/api/register>

Get Users URL: <https://reqres.in/api/users>

Logging Test URL: <https://reqres.in/api/unknown>

Step1: Create a maven project and add dependencies:

```
<dependencies>
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-api</artifactId>
    <version>2.14.1</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core -->
  <dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.14.1</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
  <dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

```

<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured-common -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured-common</artifactId>
    <version>5.3.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured-all -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured-all</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.json/json -->
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20180813</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest -->
<dependency>
    <groupId>org.hamcrest</groupId>
    <artifactId>hamcrest</artifactId>
    <version>2.2</version>
    <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/io.rest-assured/json-path -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>json-path</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -
->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.15.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.7.1</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
</dependency>

```

```

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-java</artifactId>
  <version>7.10.1</version>
</dependency>

```

```

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-junit</artifactId>
  <version>7.10.1</version>
  <scope>test</scope>
</dependency>

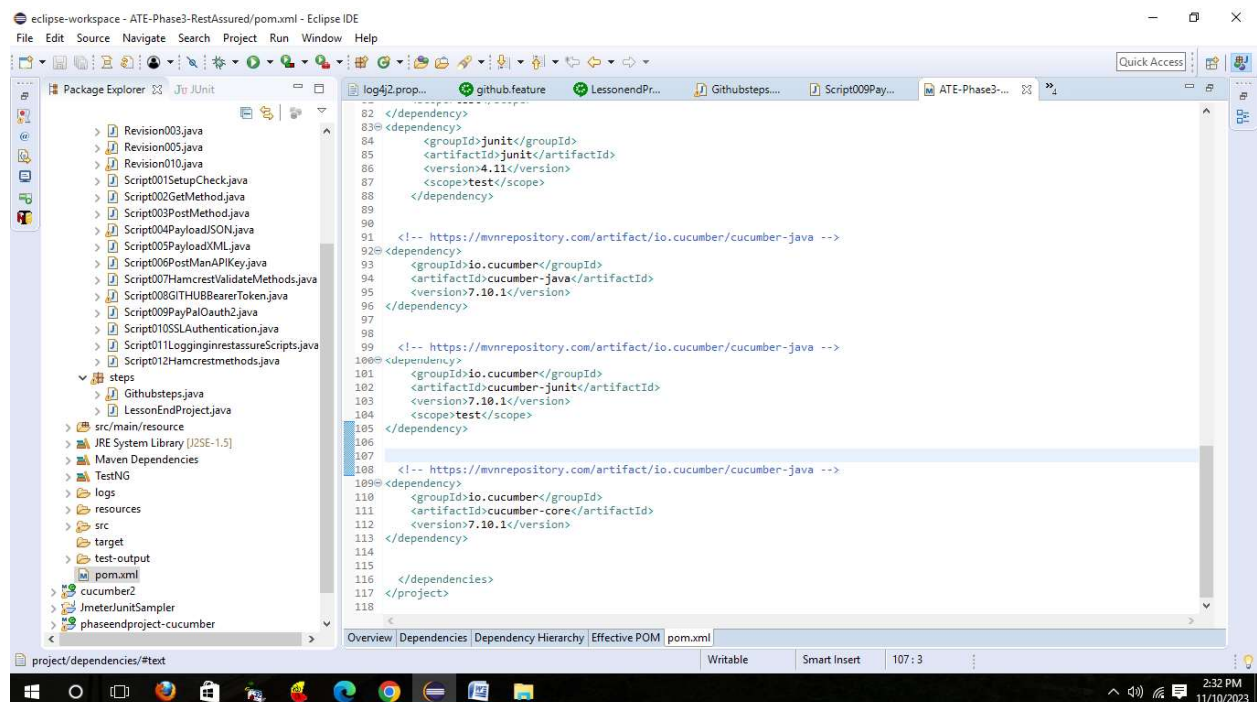
```

```

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-core</artifactId>
  <version>7.10.1</version>
</dependency>

</dependencies>

```



Step2:

Convert the maven project to cucumber project

Step3:

- Create a package features
- Create a file lessonendproject.feature
- Write the BDD code init

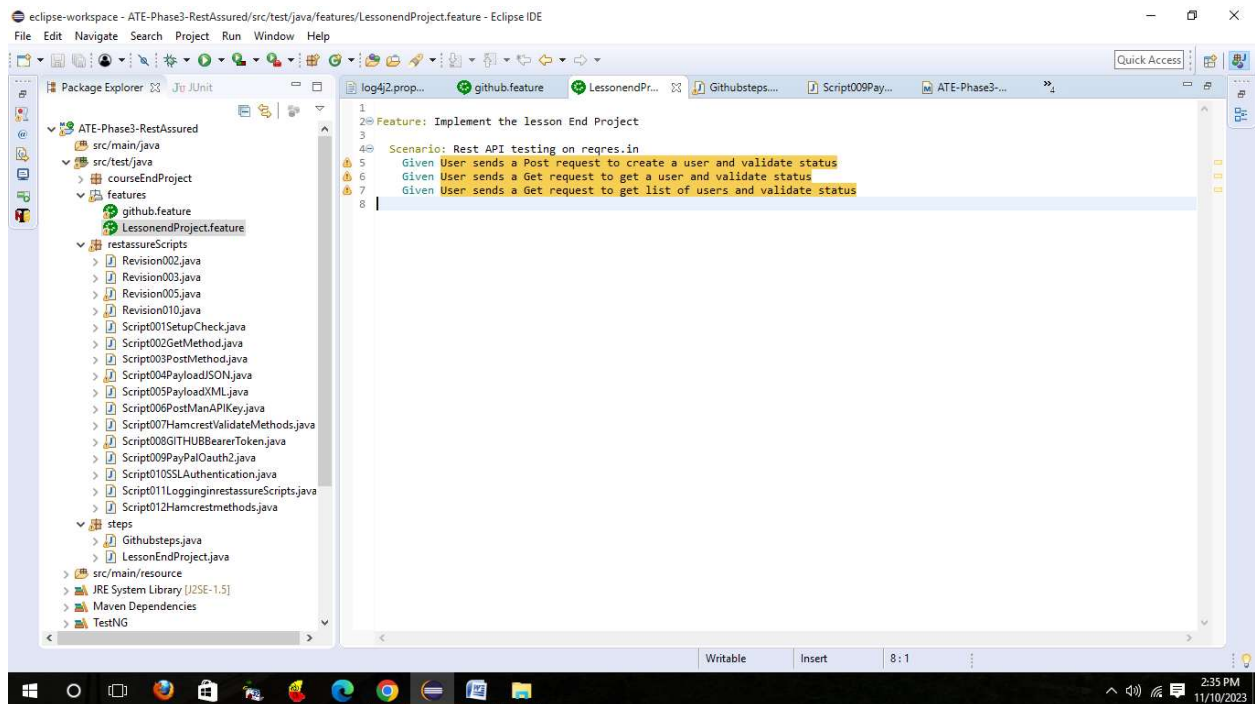
Feature: Implement the lesson End Project

Scenario: Rest API testing on reqres.in

Given User sends a Post request to create a user and validate status

Given User sends a Get request to get a user and validate status

Given User sends a Get request to get list of users and validate status



Step 4:

- Create a package steps
- In that package create step definition file for the above feature file package steps;

```
import org.json.JSONObject;  
import org.testng.annotations.Test;
```

```
import io.cucumber.java.en.Given;  
import io.restassured.RestAssured;
```

```

import io.restassured.http.ContentType;

public class LessonEndProject {
    @Test(priority='1')
    @Given("User sends a Post request to create a user and validate status")
    public void user_sends_a_post_request_to_create_a_user_and_validate_status() {
        JSONObject body = new JSONObject();
        body.put("name", "meghna");
        body.put("job", "developer");

        RestAssured.given()
            .baseUrl("https://reqres.in/")
            .basePath("/api/users")
            .contentType("application/json")
            .body(body.toString())
            .when().post()
            .then().statusCode(201)
            .log().all();
    }
    @Test(priority='2')
    @Given("User sends a Get request to get a user and validate status")
    public void user_sends_a_get_request_to_get_a_user_and_validate_status() {

        JSONObject body = new JSONObject();
        body.put("email", "meghna@gmail.com");
        body.put("password", "pas123");

        RestAssured.given().baseUrl("https://reqres.in")
            .contentType(ContentType.JSON).body(body.toString())

            .when().post("/api/register")
            .then().statusCode(200)

            // print the entire log of the request, but dont print the sensitive information

            ;
    }
    @Test(priority='3')
    @Given("User sends a Get request to get list of users and validate status")
    public void user_sends_a_get_request_to_get_list_of_users_and_validate_status() {
        RestAssured.given()
            .baseUrl("https://reqres.in")
            .basePath("/api/unknown")
            .when().get()
            .then().statusCode(200).log().all();
    }
}

```

- In the above code there are three methods
- `public void user_sends_a_post_request_to_create_a_user_and_validate_status()`
 1. in this method the user send the post request to create a user
 2. and validate the user created
 3. link to create is mentioned in the problem statement
 4. validates the status code
- `public void user_sends_a_get_request_to_get_a_user_and_validate_status()`
- `public void User_sends a Get request to get list of users and validate status()`
 1. In this method the user sends get request
 2. To fetch the users
 3. And validate the status code
 4. The link is given in the problem statement
 5. There are two methods the second method is to fetch the user created in the first method
 6. The third method is to get all the user

```

1 package steps;
2
3 import org.json.JSONObject;
4
5
6
7
8
9
10 public class LessonEndProject {
11     @Test(priority="1")
12     @Given("User sends a Post request to create a user and validate status")
13     public void user_sends_a_post_request_to_create_a_user_and_validate_status() {
14         JSONObject body = new JSONObject();
15         body.put("name", "meghna");
16         body.put("job", "developer");
17
18         RestAssured.given()
19             .baseUrl("https://reqres.in/")
20             .basePath("/api/users")
21             .contentType("application/json")
22             .body(body.toString())
23             .when().post()
24             .then().statusCode(201)
25             .log().all();
26     }
27
28     @Test(priority="2")
29     @Given("User sends a Get request to get a user and validate status")
30     public void user_sends_a_get_request_to_get_a_user_and_validate_status() {
31         JSONObject body = new JSONObject();
32         body.put("email", "meghna@gmail.com");
33         body.put("password", "pas123");
34
35         RestAssured.given().baseUrl("https://reqres.in")
36             .contentType(ContentType.JSON)
37             .body(body.toString())
38             .when().post("/api/register")
39             .then().statusCode(200)
40
41         // print the entire log of the request, but don't print the sensitive information
42     }
43
44 }

```

Output:

```

Scenario: Rest API testing on reqres.in
src/test/java/features/LessonendProject.feature:4
HTTP/1.1 201 Created
Date: Fri, 10 Nov 2023 09:14:47 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 85
Connection: keep-alive
Report-To: {"group":"heroku-
nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.com/reports?ts=1699607687
&sid=c4c9725f-1ab0-44d8-820f-
430df2718e11&s=%2BoNb9a0REIFmYGUxiKnnfLt16YSCFKNou7fc7c8%2F4c4%3D"}]}

```

Reporting-Endpoints: heroku-
nel=https://nel.heroku.com/reports?ts=1699607687&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=%2BoNb9a0REIFmYGUxiKnnfLt16YSCFKNou7fc7c8%2F4c4%3D
Nel: {"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fraction":0.05,"response_headers":["Via"]}
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"55-HrbLb+r8sVth8V7tsniCtWa0g9c"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 823d2fee5992938f-BBI

```
{
  "name": "meghna",
  "job": "developer",
  "id": "208",
  "createdAt": "2023-11-10T09:14:47.673Z"
}
{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "name": "sand dollar",
      "year": 2006,
      "color": "#DECDBE",
      "pantone_value": "13-1106"
    },
    {
      "id": 8,
      "name": "chili pepper",
      "year": 2007,
      "color": "#9B1B30",
      "pantone_value": "19-1557"
    },
    {
      "id": 9,
      "name": "blue iris",
      "year": 2008,
      "color": "#5A5B9F",
      "pantone_value": "18-3943"
    },
    {
      "id": 10,
      "name": "mimosa",

```

```
    "year": 2009,  
    "color": "#F0C05A",  
    "pantone_value": "14-0848"  
  },  
  {  
    "id": 11,  
    "name": "turquoise",  
    "year": 2010,  
    "color": "#45B5AA",  
    "pantone_value": "15-5519"  
  },  
  {  
    "id": 12,  
    "name": "honeysuckle",  
    "year": 2011,  
    "color": "#D94F70",  
    "pantone_value": "18-2120"  
  }  
],  
"support": {  
  "url": "https://reqres.in/#support-heading",  
  "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
}  
}
```