# Phase-3 Practice Project-Rest Assured

Repository: https://github.com/Joshnitha/Phase2_Practice-Projects/

Repository link of Assisted Practices :
https://github.com/Joshnitha/Phase1_Practice_Projects/tree/main

## 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

## Source code:

**Feature File:**

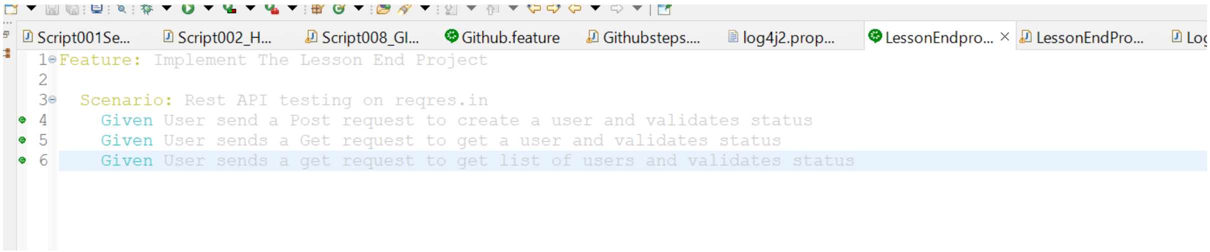
```
Feature: Implement The Lesson End Project

Scenario: Rest API testing on reqres.in

Given User send a Post request to create a user and validates status

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

Given User sends a get request to get list of users and validates status
```

**Program:**

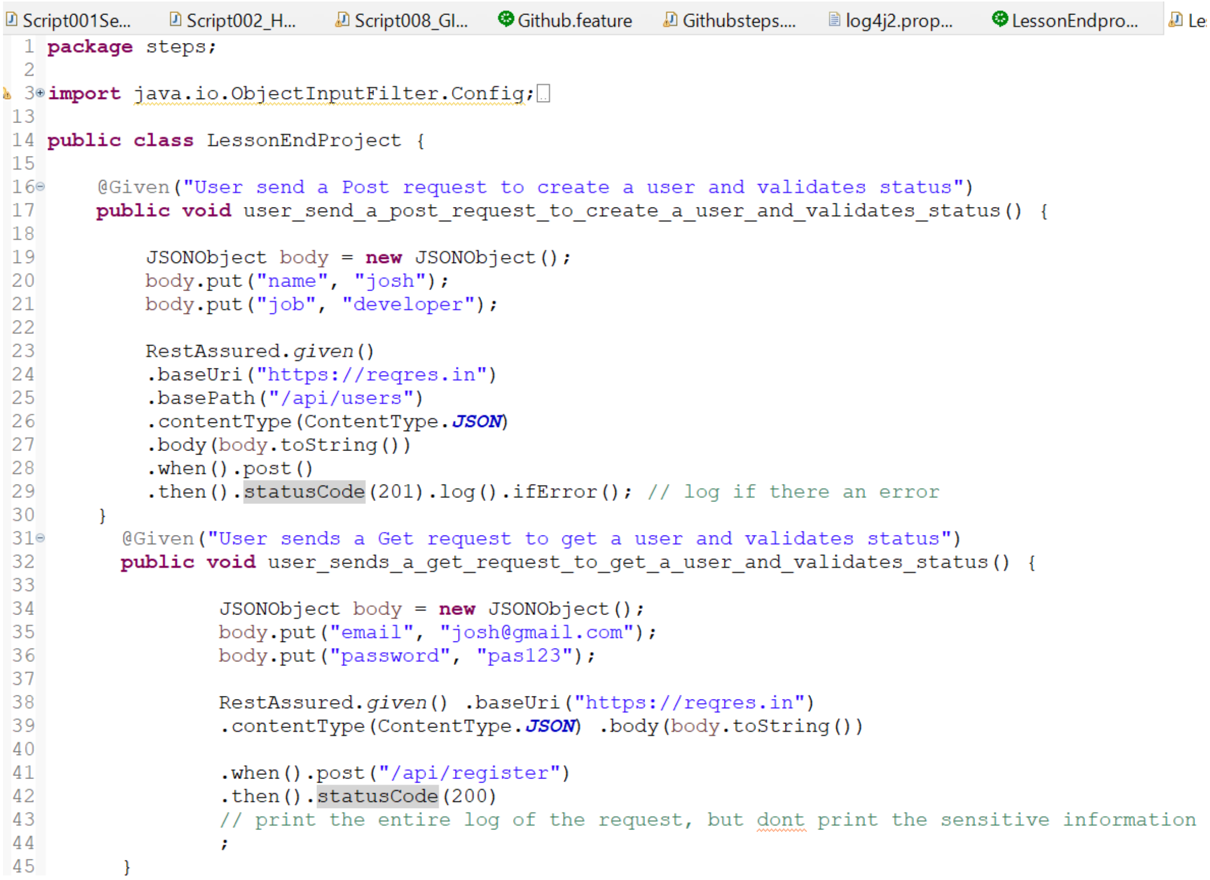
```
package steps;


import java.io.ObjectInputFilter.Config;


import org.json.JSONObject;


import io.cucumber.java.en.Given;
import io.restassured.RestAssured;
import io.restassured.config.LogConfig;
import io.restassured.http.ContentType;
import io.restassured.internal.http.AuthConfig;
import static io.restassured.RestAssured.config;

public class LessonEndProject {

	@Given("User send a Post request to create a user and validates status")
	public void user_send_a_post_request_to_create_a_user_and_validates_status() {
```

```
                JSONObject body = new JSONObject();
                body.put("name", "josh");
                body.put("job", "developer");

                RestAssured.given()
                .baseUri("https://reqres.in")
                .basePath("/api/users")
                .contentType(ContentType.JSON)
                .body(body.toString())
                .when().post()
                .then().statusCode(201).log().ifError(); // log if there an error
        }
         @Given("User sends a Get request to get a user and validates status")
         public void user_sends_a_get_request_to_get_a_user_and_validates_status() {

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

                        RestAssured.given() .baseUri("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
                        ;
         }
        @Given("User sends a get request to get list of users and validates status")
        public void user_sends_a_get_request_to_get_list_of_users_and_validates_status() {



                RestAssured.given()
                .baseUri("https://reqres.in")
                .basePath("/api/unknown")
                .when().get()
                .then().statusCode(200).log().all();
        }
}
```

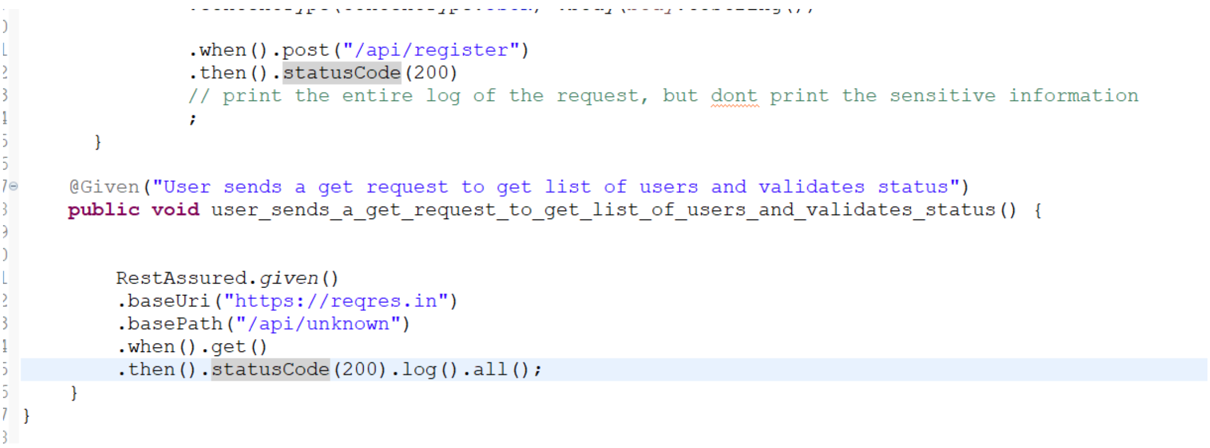
**ScreenShots:**



```
Feature: Implement The Lesson End Project

  Scenario: Rest API testing on reqres.in
    Given User send a Post request to create a user and validates status
    Given User sends a Get request to get a user and validates status
    Given User sends a get request to get list of users and validates status
```

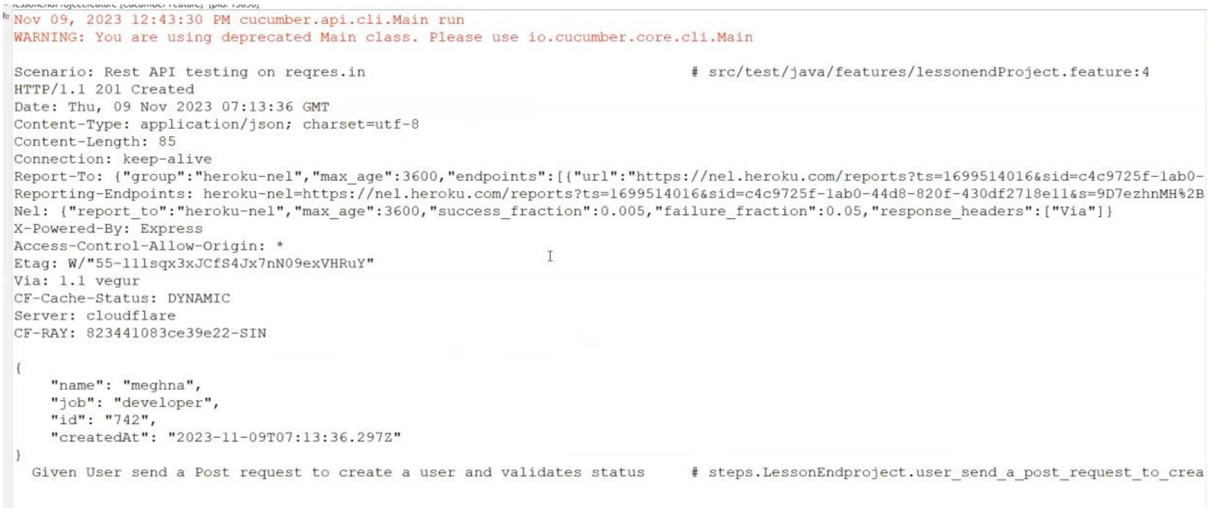


```java
package steps;

import java.io.ObjectInputFilter.Config;

public class LessonEndProject {

    @Given("User send a Post request to create a user and validates status")
    public void user_send_a_post_request_to_create_a_user_and_validates_status() {

        JSONObject body = new JSONObject();
        body.put("name", "josh");
        body.put("job", "developer");

        RestAssured.given()
        .baseUri("https://reqres.in")
        .basePath("/api/users")
        .contentType(ContentType.JSON)
        .body(body.toString())
        .when().post()
        .then().statusCode(201).log().ifError(); // log if there an error
    }
      @Given("User sends a Get request to get a user and validates status")
      public void user_sends_a_get_request_to_get_a_user_and_validates_status() {

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

              RestAssured.given() .baseUri("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
              ;
      }
```

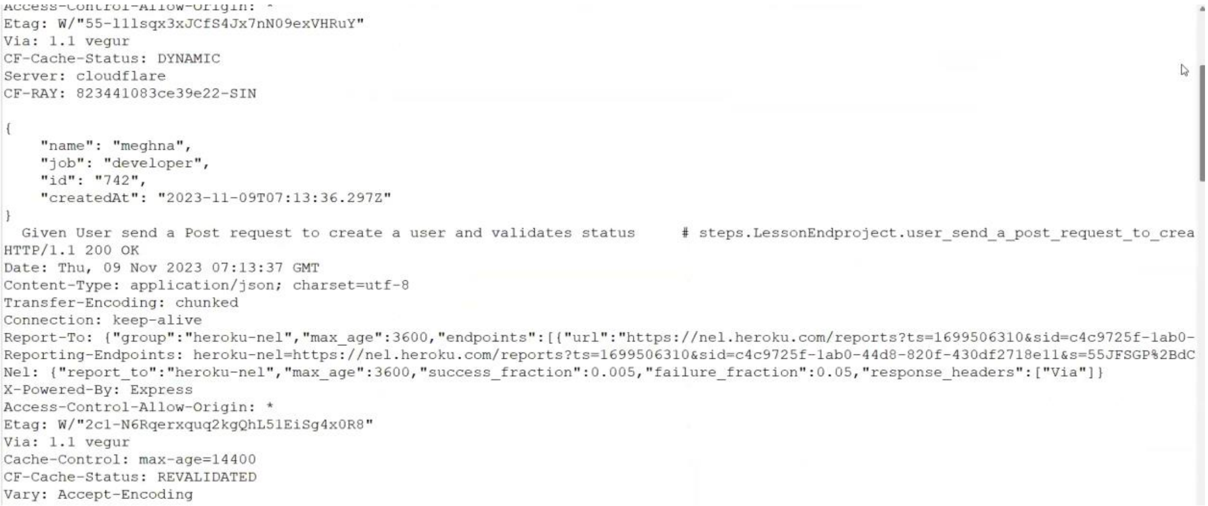


```java
              .when().post("/api/register")
              .then().statusCode(200)
              // print the entire log of the request, but dont print the sensitive information
              ;
      }

    @Given("User sends a get request to get list of users and validates status")
    public void user_sends_a_get_request_to_get_list_of_users_and_validates_status() {


        RestAssured.given()
        .baseUri("https://reqres.in")
        .basePath("/api/unknown")
        .when().get()
        .then().statusCode(200).log().all();
    }
}
```

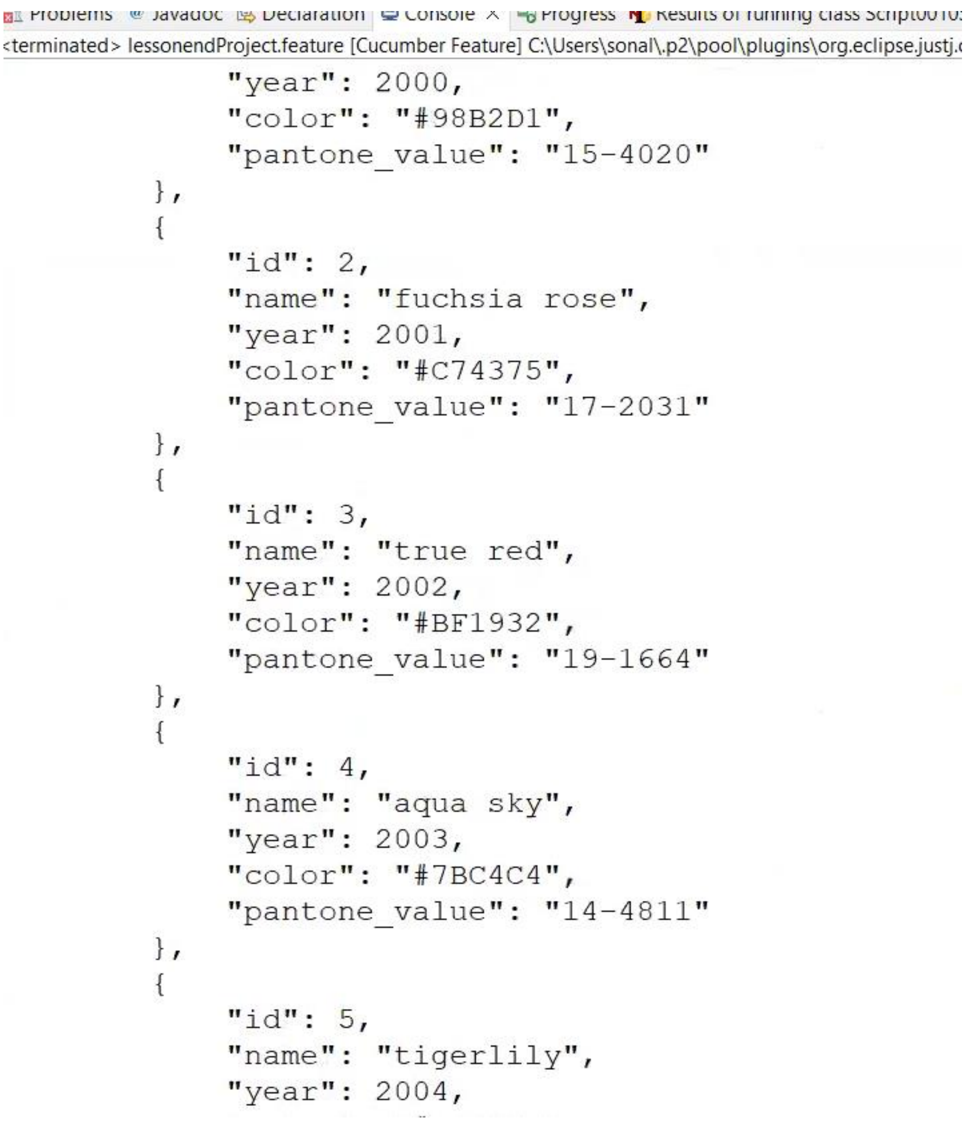
**Output:**



```
Nov 09, 2023 12:43:30 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

Scenario: Rest API testing on reqres.in                                    # src/test/java/features/lessonendProject.feature:4
HTTP/1.1 201 Created
Date: Thu, 09 Nov 2023 07:13:36 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=1699514016&sid=c4c9725f-1ab0-
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1699514016&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=9D7ezhnMH%2B
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/"55-111sqx3xJCfS4Jx7nN09exVHRuY"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 823441083ce39e22-SIN

{
    "name": "meghna",
    "job": "developer",
    "id": "742",
    "createdAt": "2023-11-09T07:13:36.297Z"
}
  Given User send a Post request to create a user and validates status     # steps.LessonEndproject.user_send_a_post_request_to_crea
```



```
Access-Control-Allow-Origin: *
Etag: W/"55-111sqx3xJCfS4Jx7nN09exVHRuY"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 823441083ce39e22-SIN

{
    "name": "meghna",
    "job": "developer",
    "id": "742",
    "createdAt": "2023-11-09T07:13:36.297Z"
}
  Given User send a Post request to create a user and validates status     # steps.LessonEndproject.user_send_a_post_request_to_crea
HTTP/1.1 200 OK
Date: Thu, 09 Nov 2023 07:13:37 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.com/reports?ts=1699506310&sid=c4c9725f-1ab0-
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1699506310&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=55JFSGP%2BdC
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/"2c1-N6Rqerxquq2kgQhL51EiSg4x0R8"
Via: 1.1 vegur
Cache-Control: max-age=14400
CF-Cache-Status: REVALIDATED
Vary: Accept-Encoding
```



<terminated> lessonendProject.feature [Cucumber Feature] C:\Users\sonal\.p2\pool\plugins\org.eclipse.justj.

```
        "year": 2000,
        "color": "#98B2D1",
        "pantone_value": "15-4020"
    },
    {
        "id": 2,
        "name": "fuchsia rose",
        "year": 2001,
        "color": "#C74375",
        "pantone_value": "17-2031"
    },
    {
        "id": 3,
        "name": "true red",
        "year": 2002,
        "color": "#BF1932",
        "pantone_value": "19-1664"
    },
    {
        "id": 4,
        "name": "aqua sky",
        "year": 2003,
        "color": "#7BC4C4",
        "pantone_value": "14-4811"
    },
    {
        "id": 5,
        "name": "tigerlily",
        "year": 2004,
```