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

**Project Aims:**

**Rest Assured:**

**Rest Assured** enables to test REST APIs using java libraries and integrates well with Maven. It has very efficient matching techniques, so asserting your expected results is also pretty straight forward. Rest Assured has methods to fetch data from almost every part of the request and response no matter how complex the JSON structures are.

**TestNg:**

TestNG ("Testing, the Next Generation") is a Java unit testing framework that aims to overcome many limitations of JUnit. TestNG introduces some new functionalities that make it more powerful and easier to use.

**JAVA:**

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA),meaning that compiled Java code can run on all platforms that support Java without the need to recompile.

**I used the following:**

* Eclipse  IDE
* Java 1.8
* TestNG
* RestAssured
* Maven

**I Used these URL Endpoint:**

**Base URL:** <https://reqres.in/api>

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

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

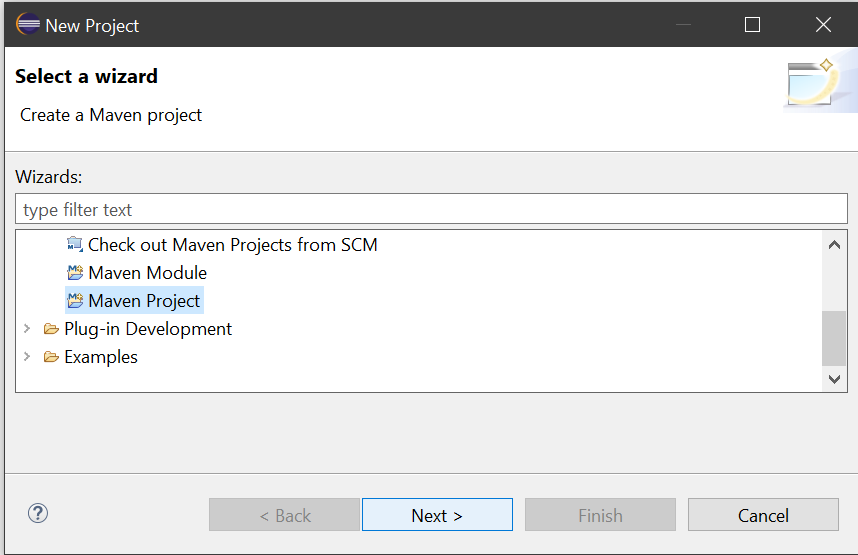
**Login User URL:** <https://reqres.in/api/login>

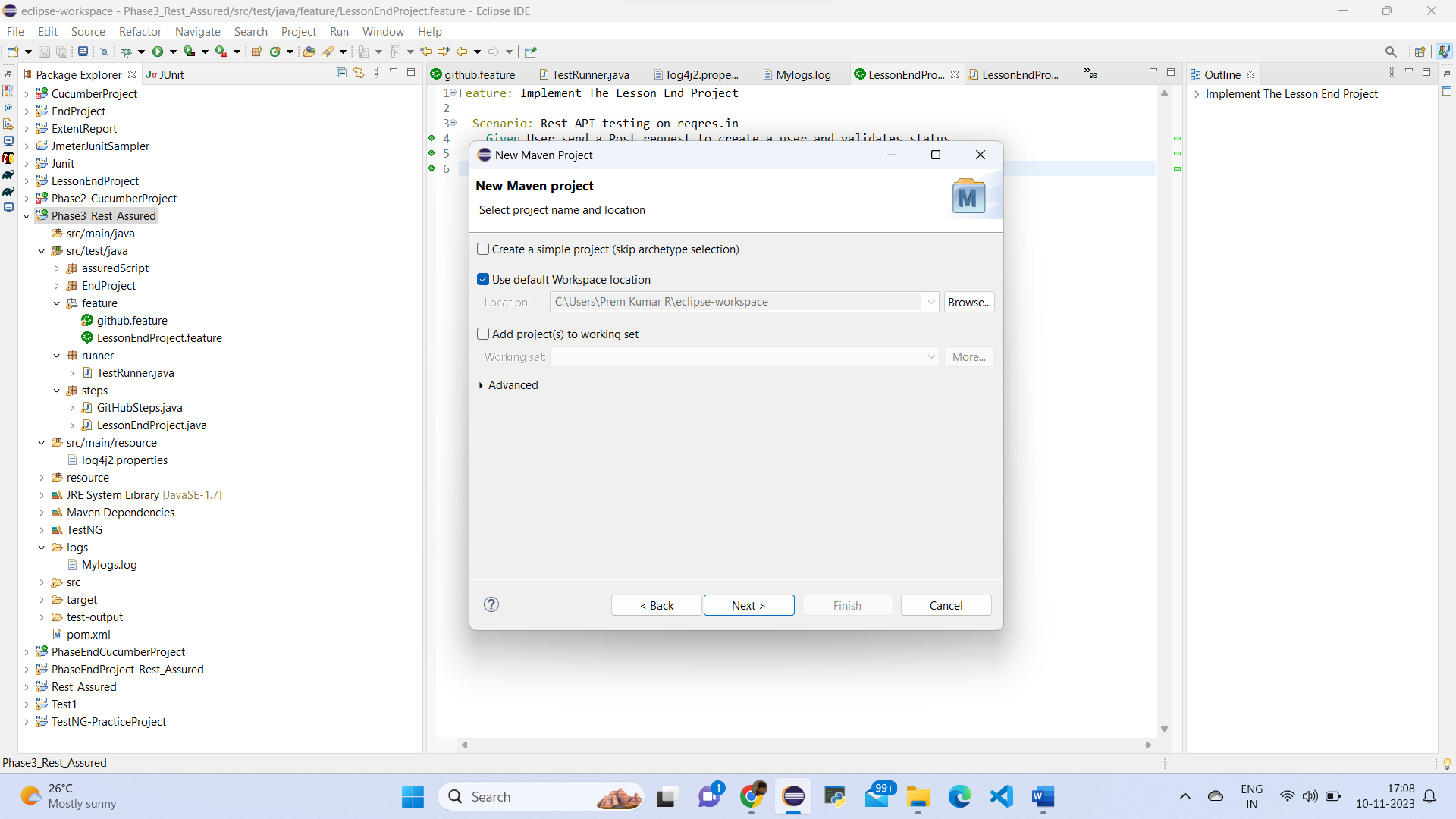
**Logging Test URL:** <https://reqres.in/api/unknown>

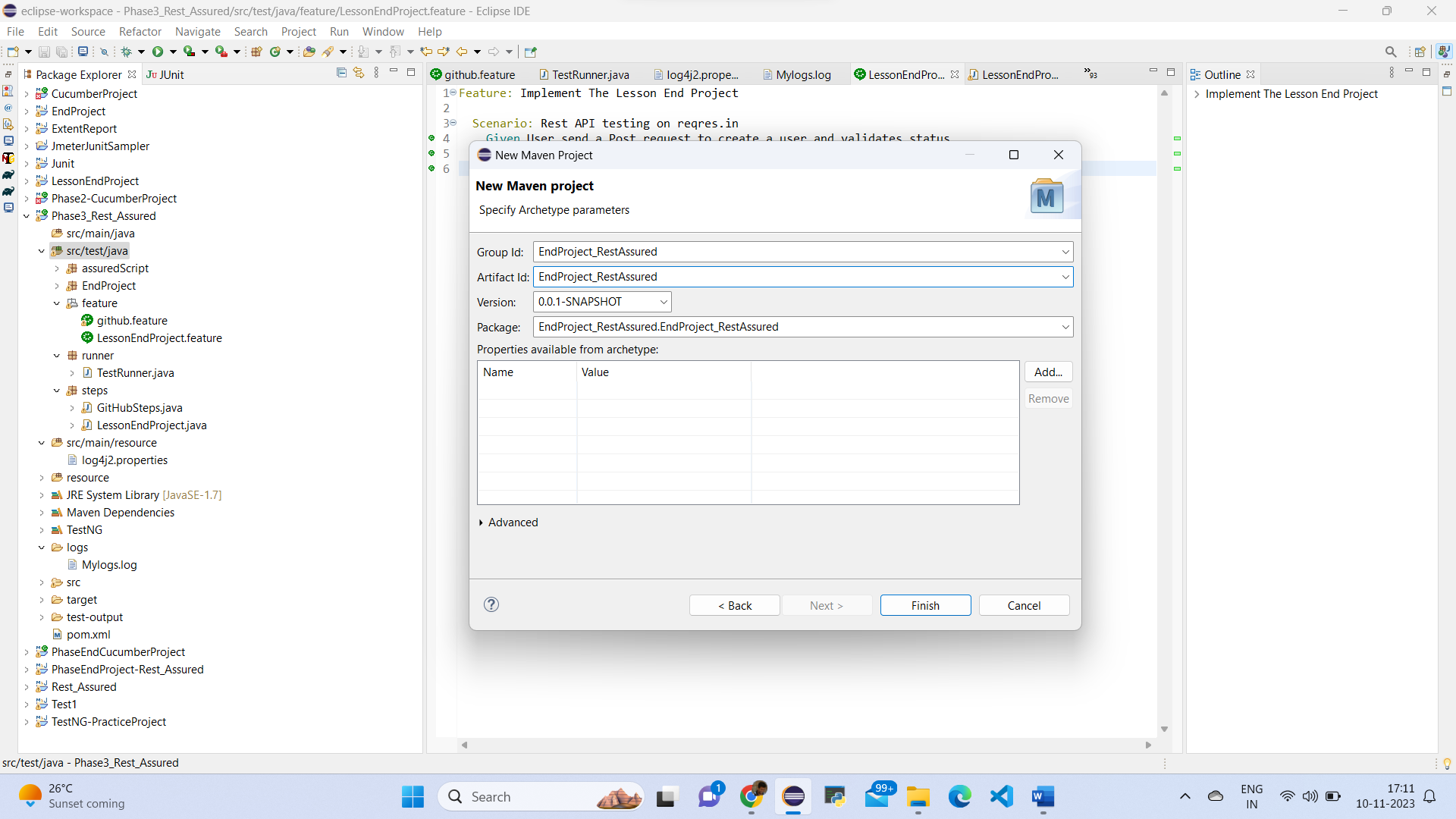
**STEPS**

Step-1: First I open Eclipse IDE

Step-2: Then I create a Maven Project

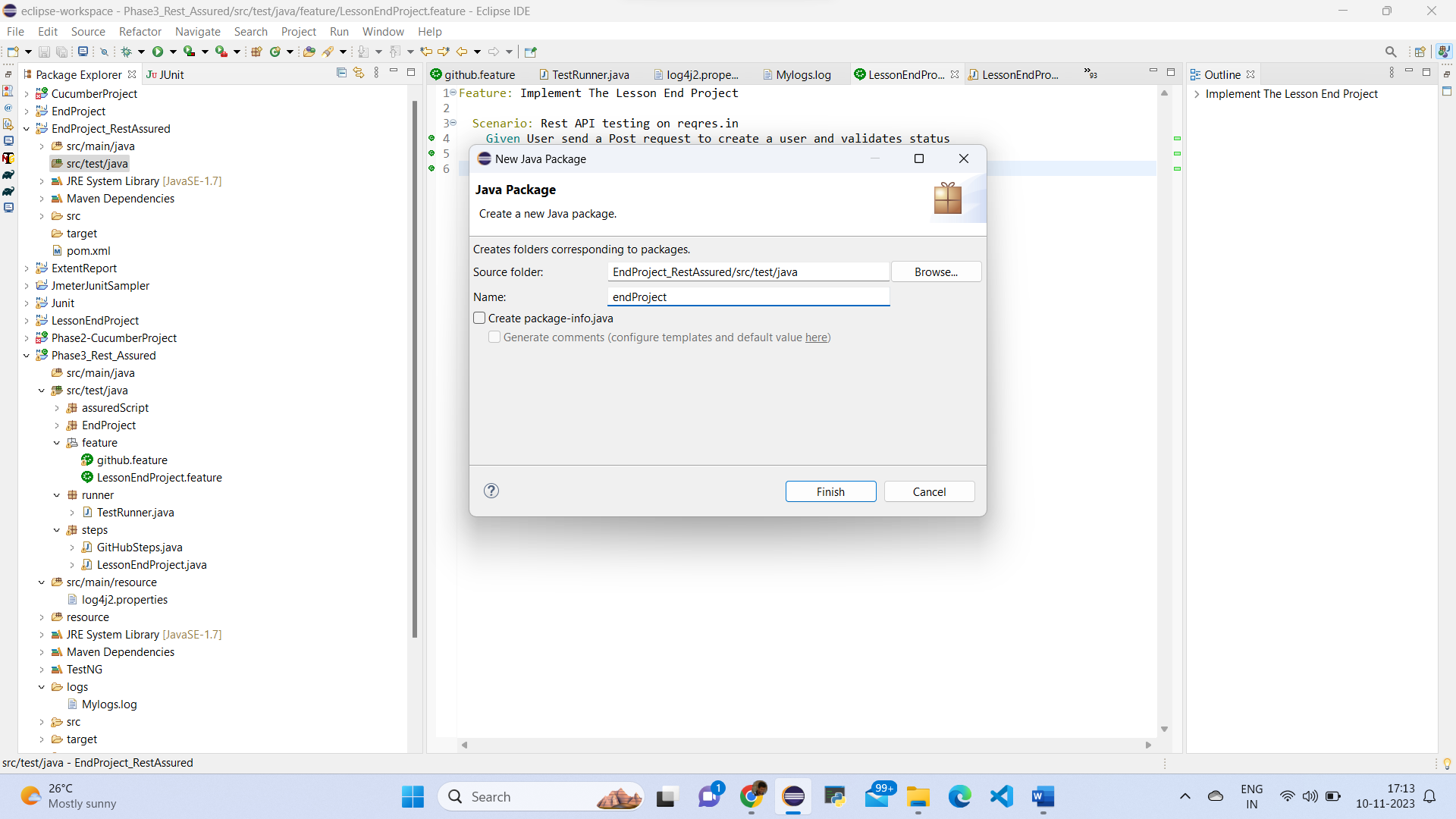




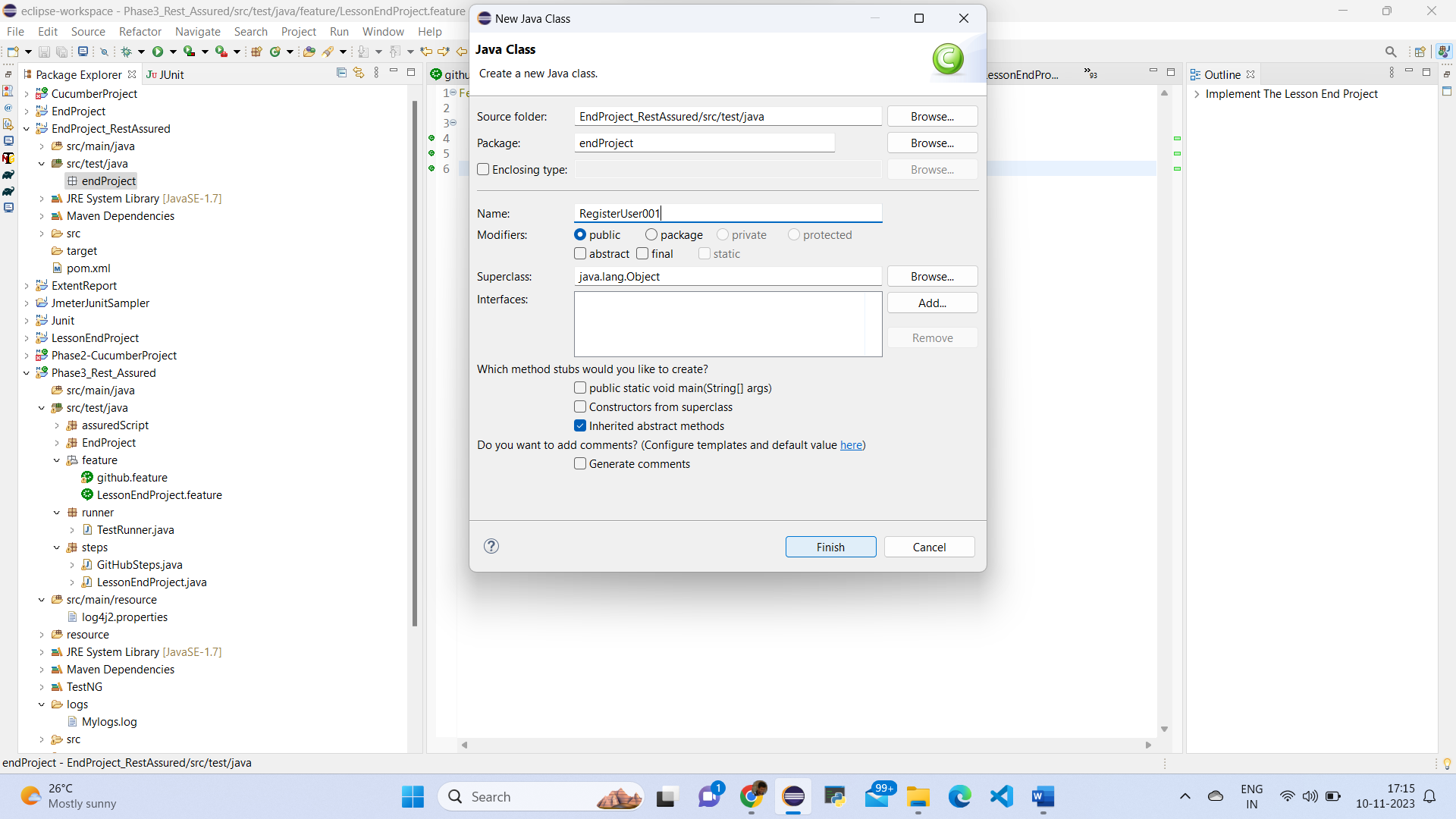


Step-3: Then I add dependencies in pom.xml file

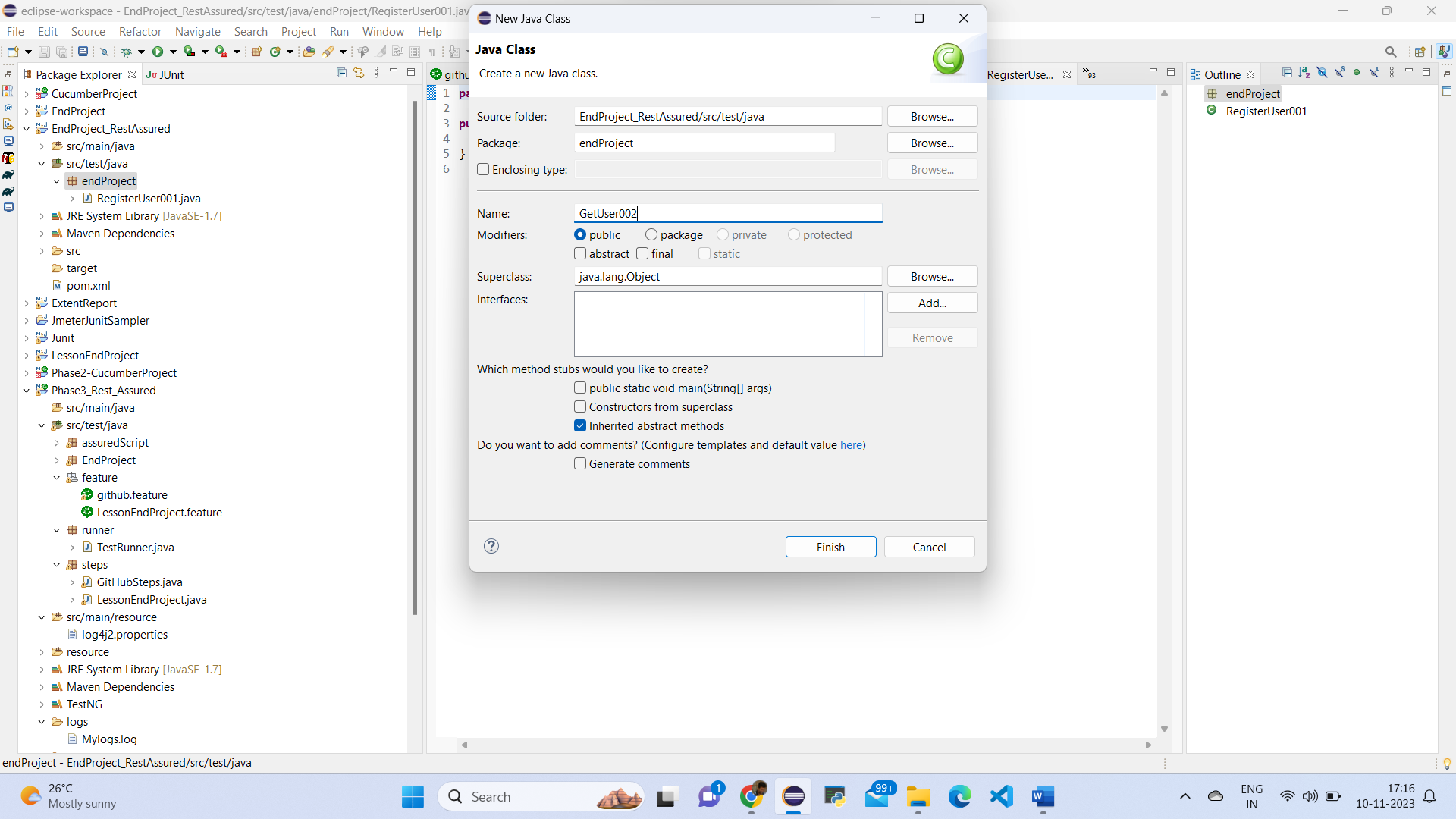
Step-4: Then I create a package in src/test/java



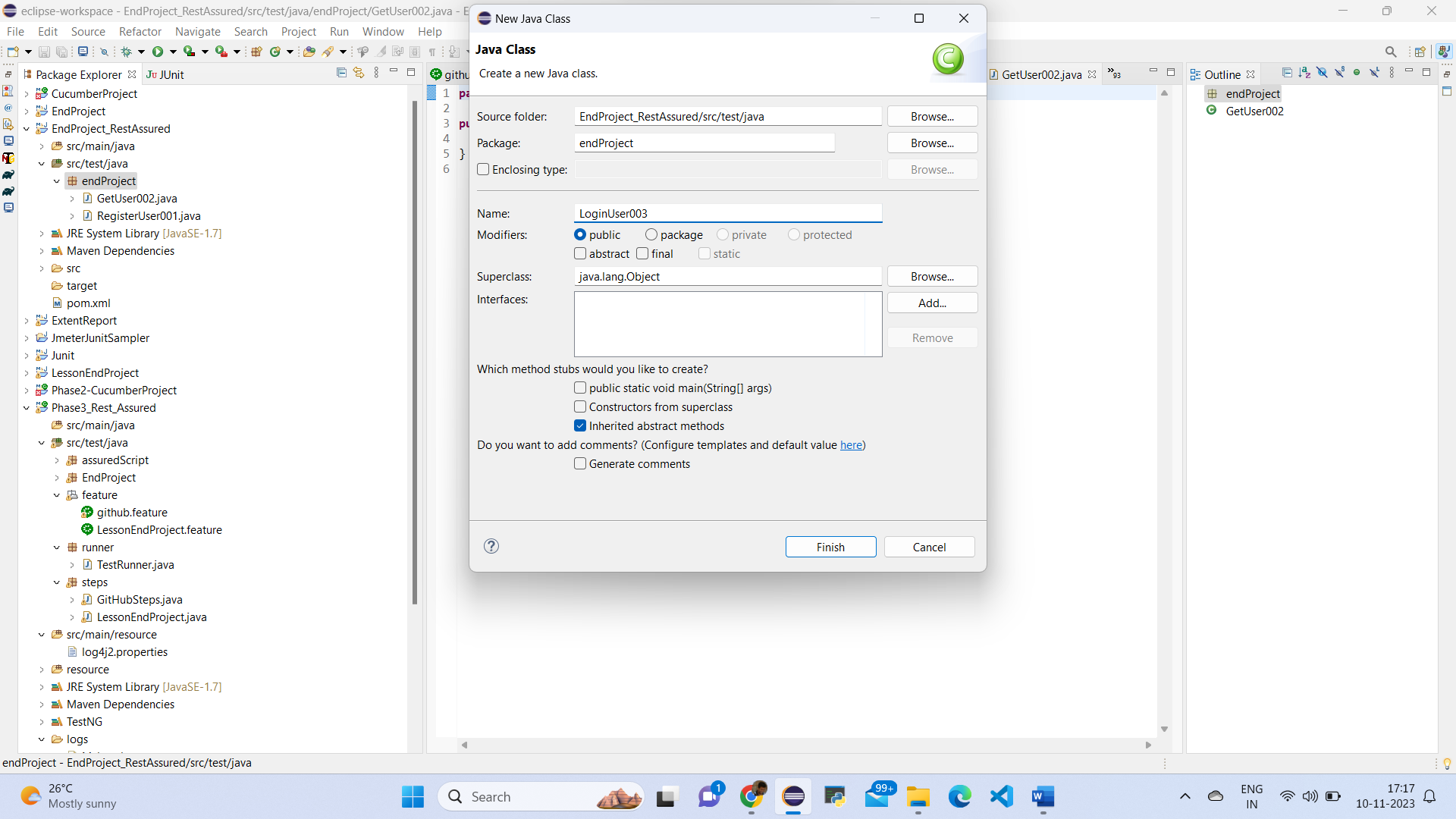
Step-5: Then I create a new class as RegisterUser001.java



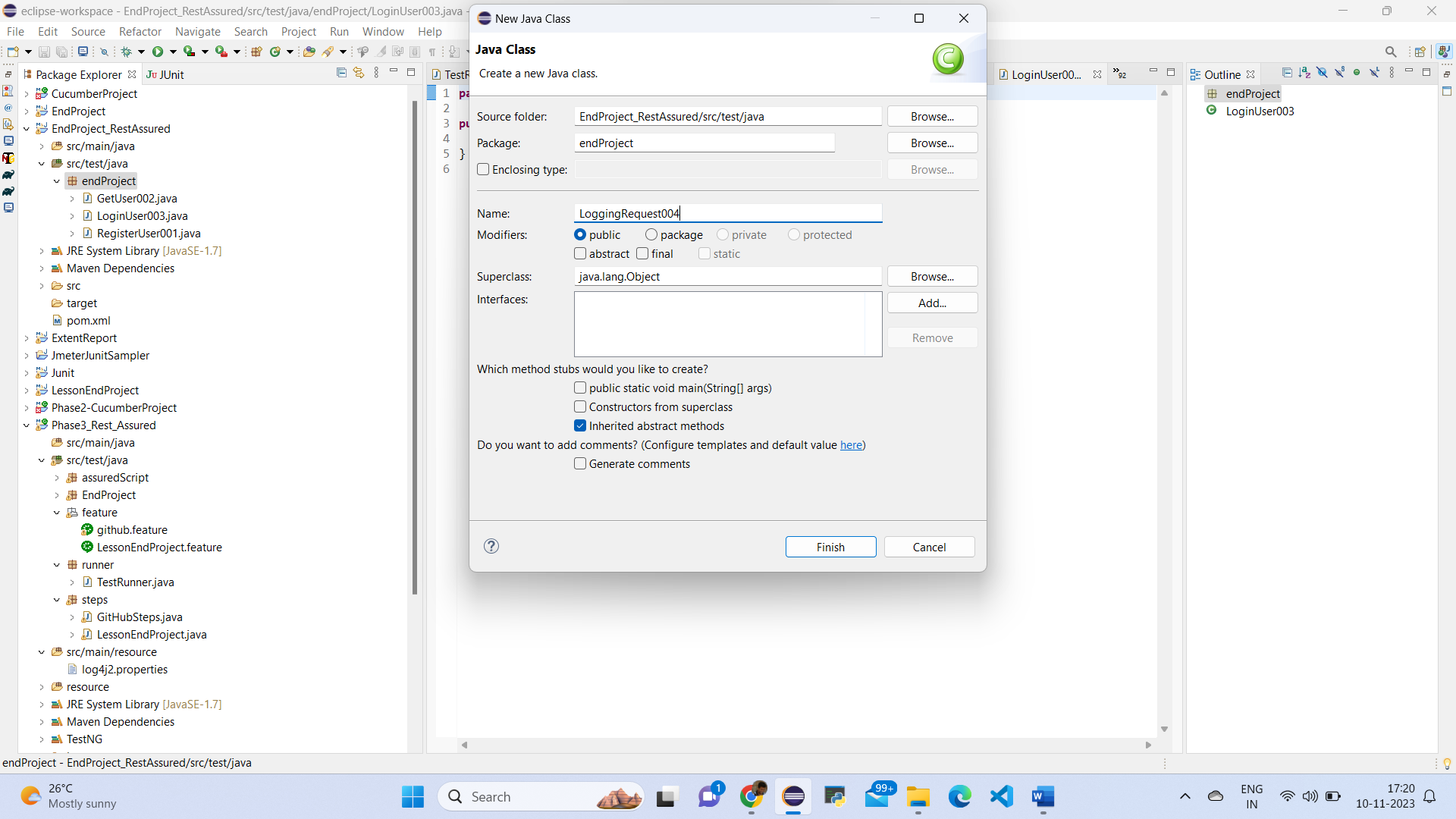
Step-6: Then I create another class named as GetUser002.java



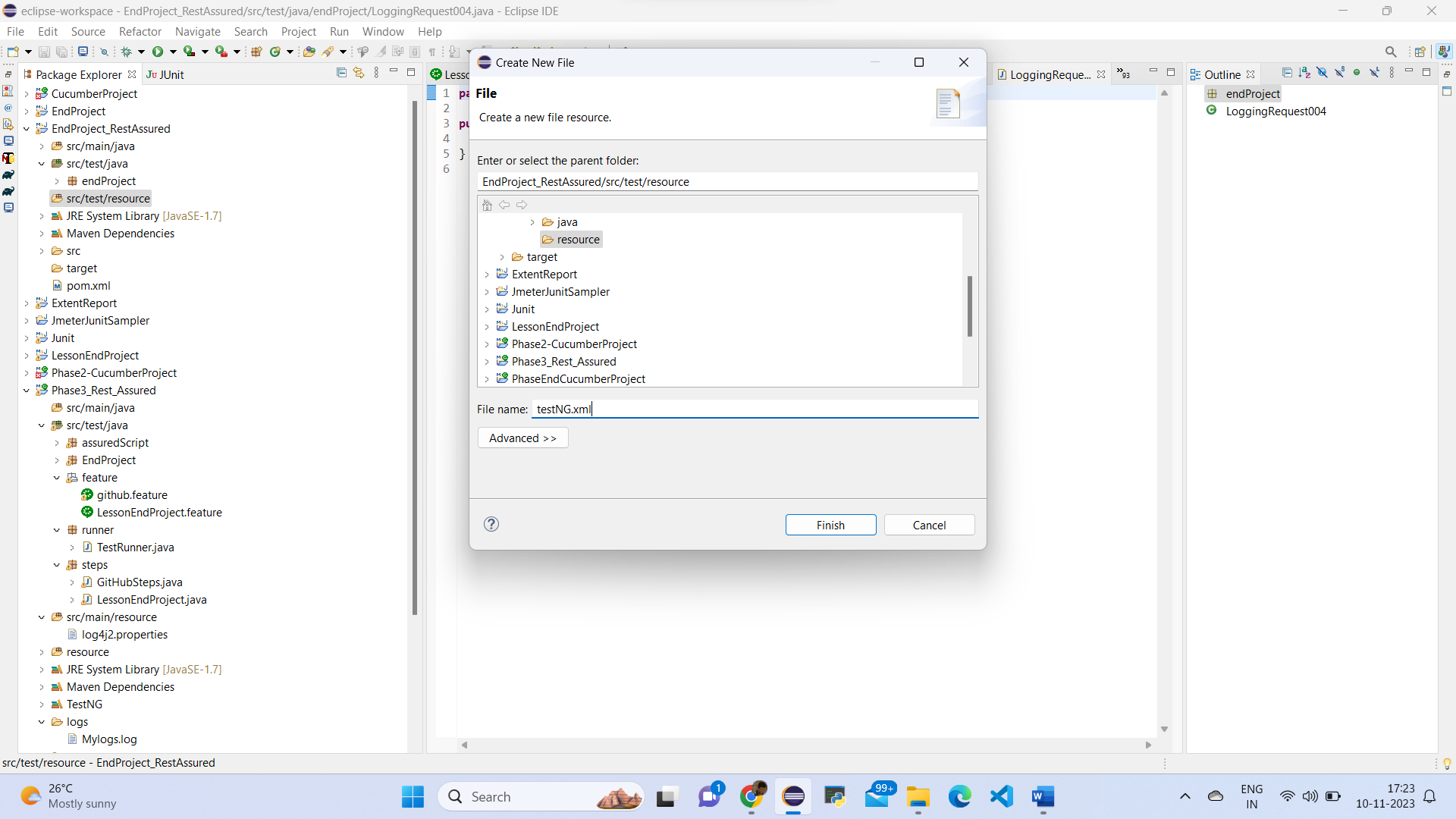
Step-7:Then I create another class named asLoginUser003.java



Step-8: Then I create a new class Script004LoggingRequest.java

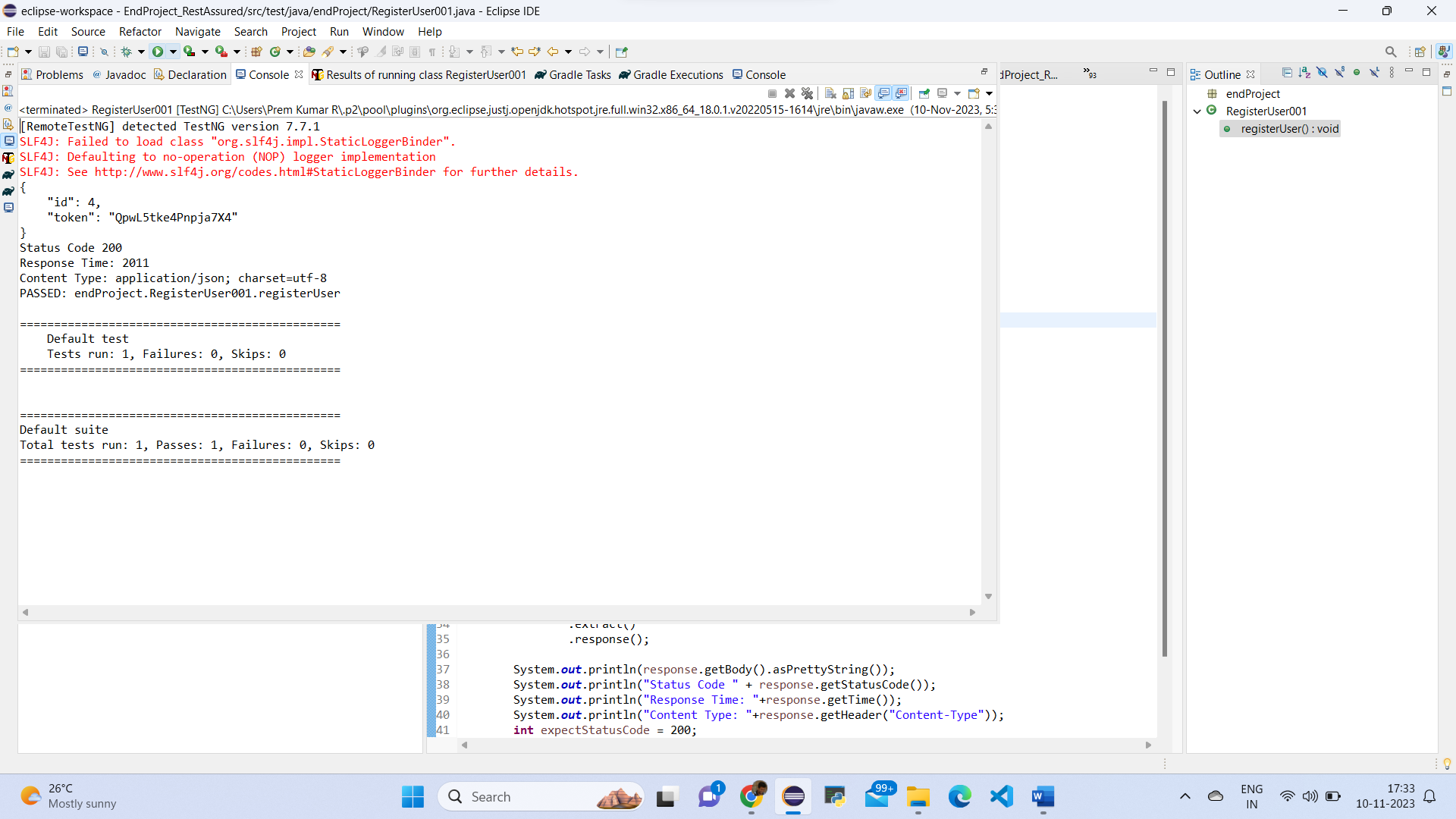


Step-9: Then I Create a new xml file as testNg.xml in src/test/resources folder

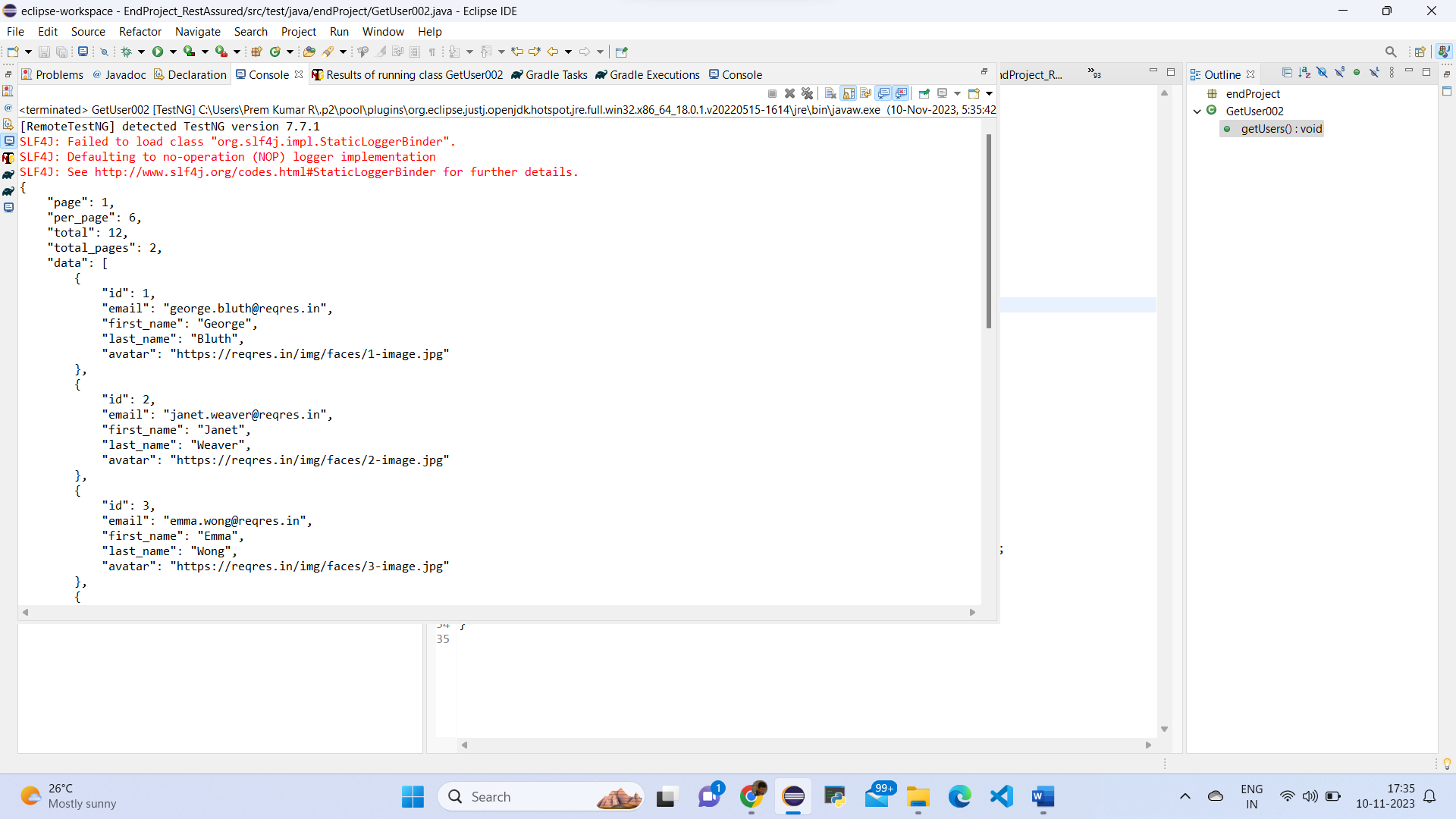


Step-10: Then I run it separately and via testNG.xml also

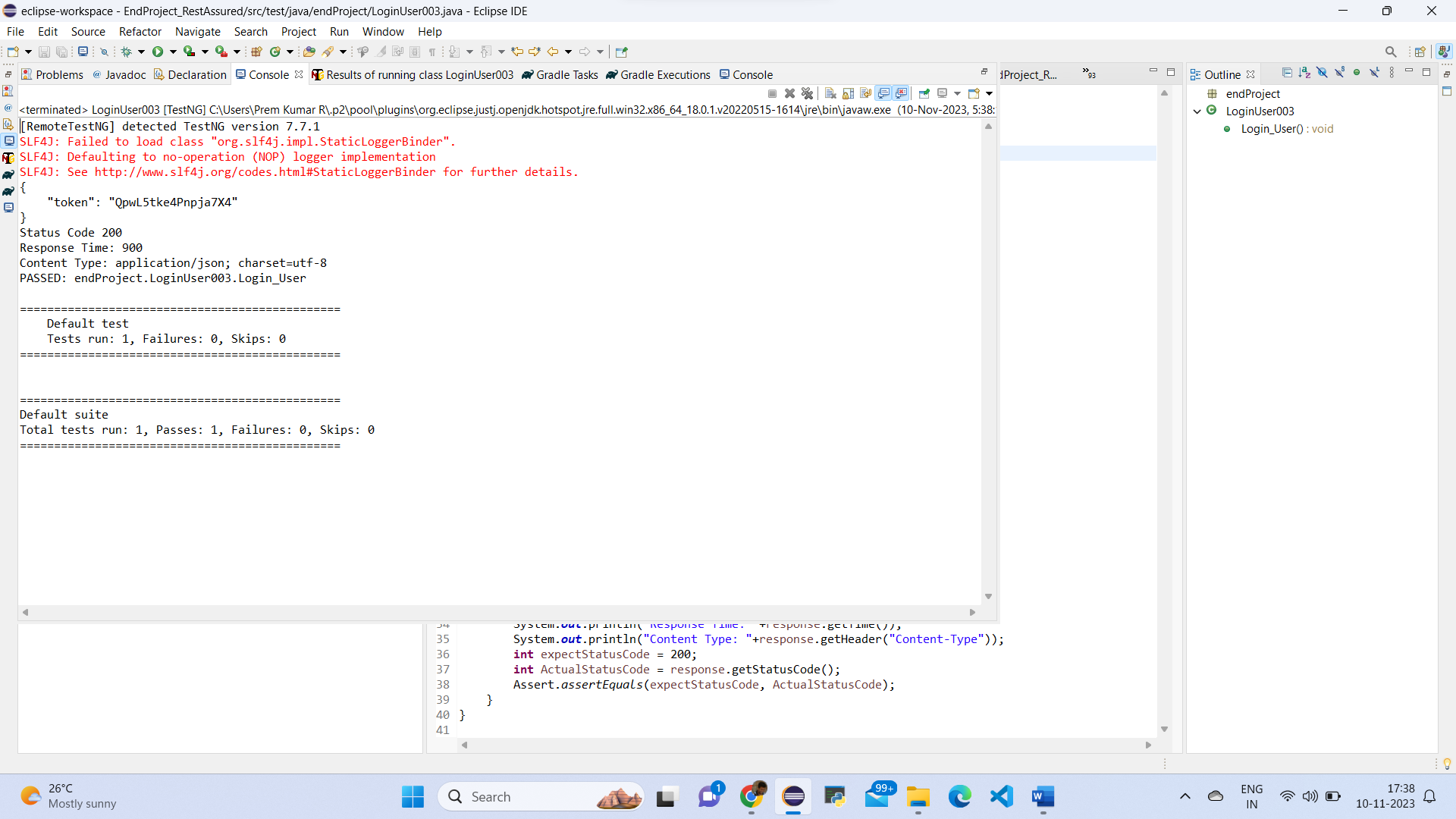
**RegisterUser001.java**



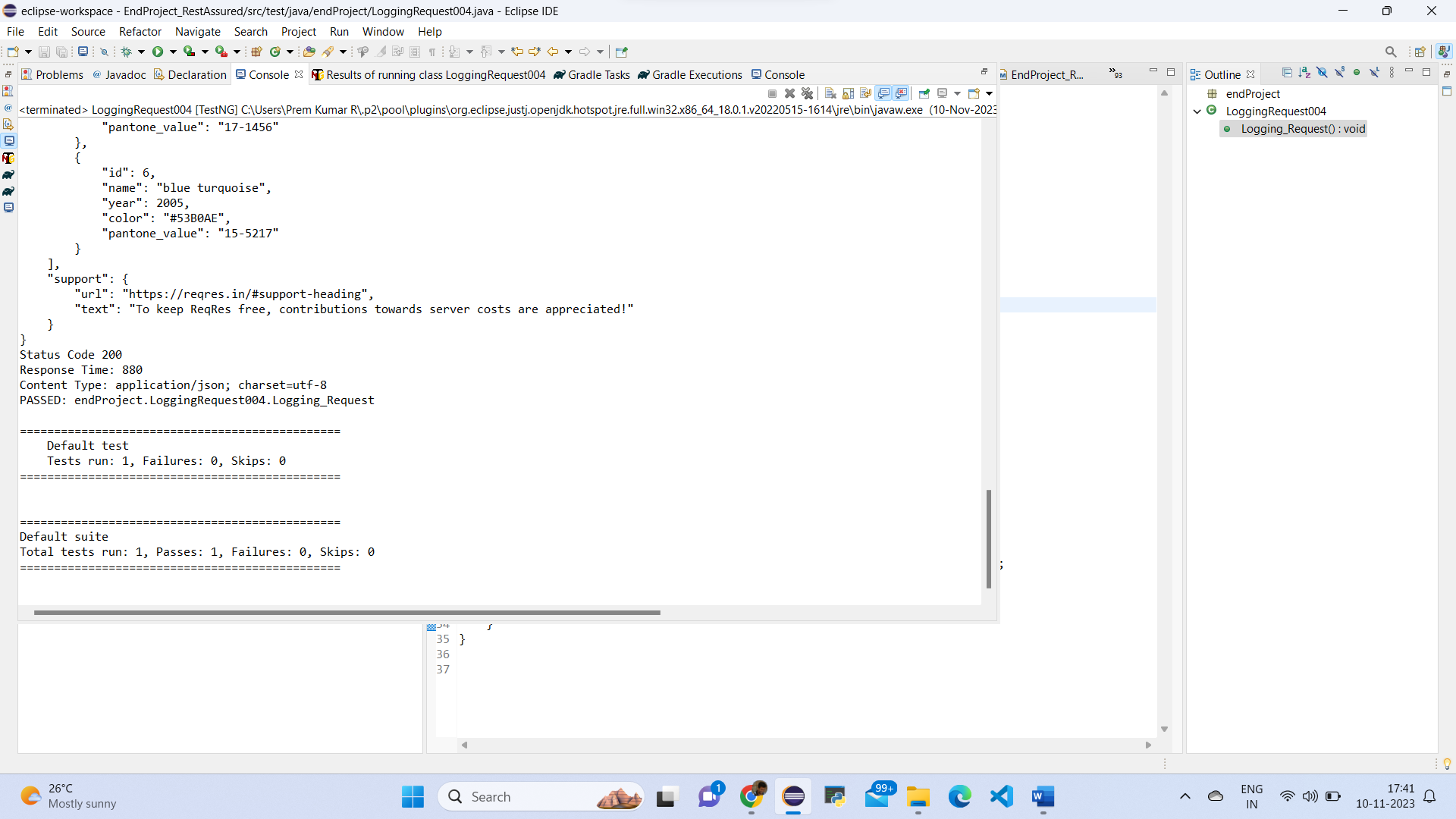
**GetUser002.java**



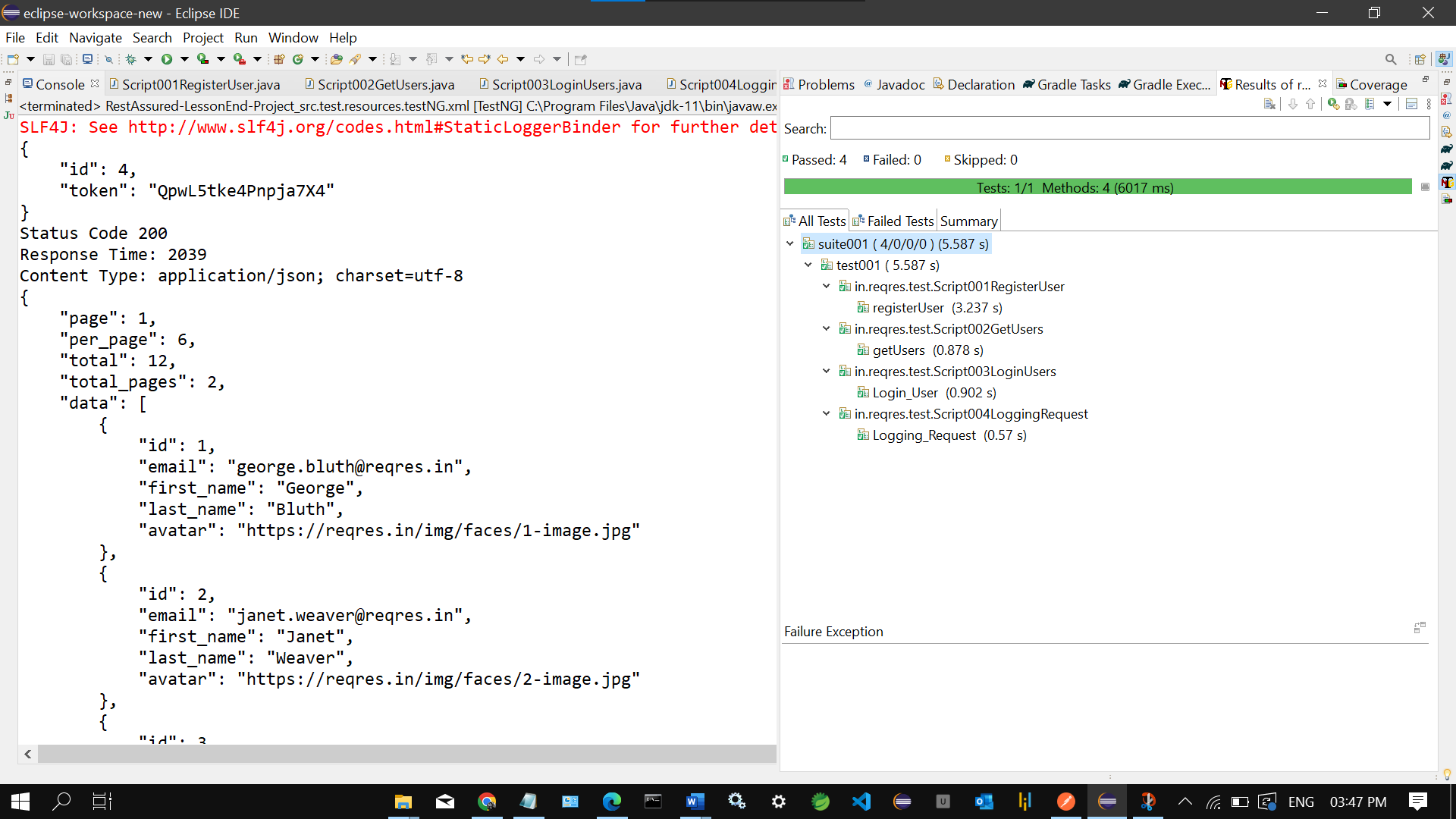
**LoginUser003.java**



**LoggingRequest004.java**



**Run From testNG.xml**



**Source Code**

**RegisterUser001.java**

**package** endProject;

**import** org.json.JSONObject;

**import** org.testng.Assert;

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

**import** io.restassured.RestAssured;

**import** io.restassured.response.Response;

**public** **class** RegisterUser001 {

@Test

**public** **void** registerUser() {

JSONObject body = **new** JSONObject();

body.put("email", "eve.holt@reqres.in");

body.put("password", "pistol");

String jsonBody=body.toString();

Response response = RestAssured

.*given*()

.baseUri("https://reqres.in")

.basePath("/api/register")

.contentType("application/json")

.body(jsonBody)

.when()

.post()

.then()

.statusCode(200)

.extract()

.response();

System.***out***.println(response.getBody().asPrettyString());

System.***out***.println("Status Code " + response.getStatusCode());

System.***out***.println("Response Time: "+response.getTime());

System.***out***.println("Content Type: "+response.getHeader("Content-Type"));

**int** expectStatusCode = 200;

**int** ActualStatusCode = response.getStatusCode();

Assert.*assertEquals*(expectStatusCode, ActualStatusCode);

}

}

**GetUser002.java**

**package** endProject;

**import** org.testng.Assert;

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

**import** io.restassured.RestAssured;

**import** io.restassured.response.Response;

**public** **class** GetUser002 {

@Test

**public** **void** getUsers() {

Response response = RestAssured

.*given*()

.baseUri("https://reqres.in")

.basePath("/api/users")

.contentType("application/json")

.when()

.get()

.then()

.statusCode(200)

.extract()

.response();

System.***out***.println(response.getBody().asPrettyString());

System.***out***.println("Status Code " + response.getStatusCode());

System.***out***.println("Response Time: "+response.getTime());

System.***out***.println("Content Type: "+response.getHeader("Content-Type"));

**int** expectStatusCode = 200;

**int** ActualStatusCode = response.getStatusCode();

Assert.*assertEquals*(expectStatusCode, ActualStatusCode);

}

}

**LoginUser003.java**

**package** endProject;

**import** org.json.JSONObject;

**import** org.testng.Assert;

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

**import** io.restassured.RestAssured;

**import** io.restassured.response.Response;

**public** **class** LoginUser003 {

@Test

**public** **void** Login\_User() {

JSONObject body = **new** JSONObject();

body.put("email", "eve.holt@reqres.in");

body.put("password", "cityslicka");

String jsonBody=body.toString();

Response response = RestAssured

.*given*()

.baseUri("https://reqres.in")

.basePath("/api/login")

.contentType("application/json")

.body(jsonBody)

.when()

.post()

.then()

.statusCode(200)

.extract()

.response();

System.***out***.println(response.getBody().asPrettyString());

System.***out***.println("Status Code " + response.getStatusCode());

System.***out***.println("Response Time: "+response.getTime());

System.***out***.println("Content Type: "+response.getHeader("Content-Type"));

**int** expectStatusCode = 200;

**int** ActualStatusCode = response.getStatusCode();

Assert.*assertEquals*(expectStatusCode, ActualStatusCode);

}

}

**LoggingRequest004.java**

**package** endProject;

**import** org.testng.Assert;

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

**import** io.restassured.RestAssured;

**import** io.restassured.response.Response;

**public** **class** LoggingRequest004 {

@Test

**public** **void** Logging\_Request() {

Response response = RestAssured

.*given*()

.baseUri("https://reqres.in")

.basePath("/api/unknown")

.contentType("application/json")

.when()

.get()

.then()

.statusCode(200)

.log().all()

.extract()

.response();

System.***out***.println(response.getBody().asPrettyString());

System.***out***.println("Status Code " + response.getStatusCode());

System.***out***.println("Response Time: "+response.getTime());

System.***out***.println("Content Type: "+response.getHeader("Content-Type"));

**int** expectStatusCode = 200;

**int** ActualStatusCode = response.getStatusCode();

Assert.*assertEquals*(expectStatusCode, ActualStatusCode);

}

}

**testNg.xml**

<suite name=*"suite001"*>

<test name=*"test001"*>

<classes>

<class name=*"in.reqres.test.RegisterUser001"*></class>

<class name=*"in.reqres.test.GetUser002"*></class>

<class name=*"in.reqres.test.LoginUser003"*></class>

<class name=*"in.reqres.test.LoggingRequest004"*></class>

</classes>

</test>

</suite>

**Pom.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>EndProject\_RestAssured</groupId>

<artifactId>EndProject\_RestAssured</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>EndProject\_RestAssured</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<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>

<!-- 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>

<!-- 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/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/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/com.fasterxml.jackson.core/jackson-databind -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.15.2</version>

</dependency>

<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/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.7.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<plugins>

<!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

<!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

<plugin>

<artifactId>maven-site-plugin</artifactId>

<version>3.7.1</version>

</plugin>

<plugin>

<artifactId>maven-project-info-reports-plugin</artifactId>

<version>3.0.0</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>