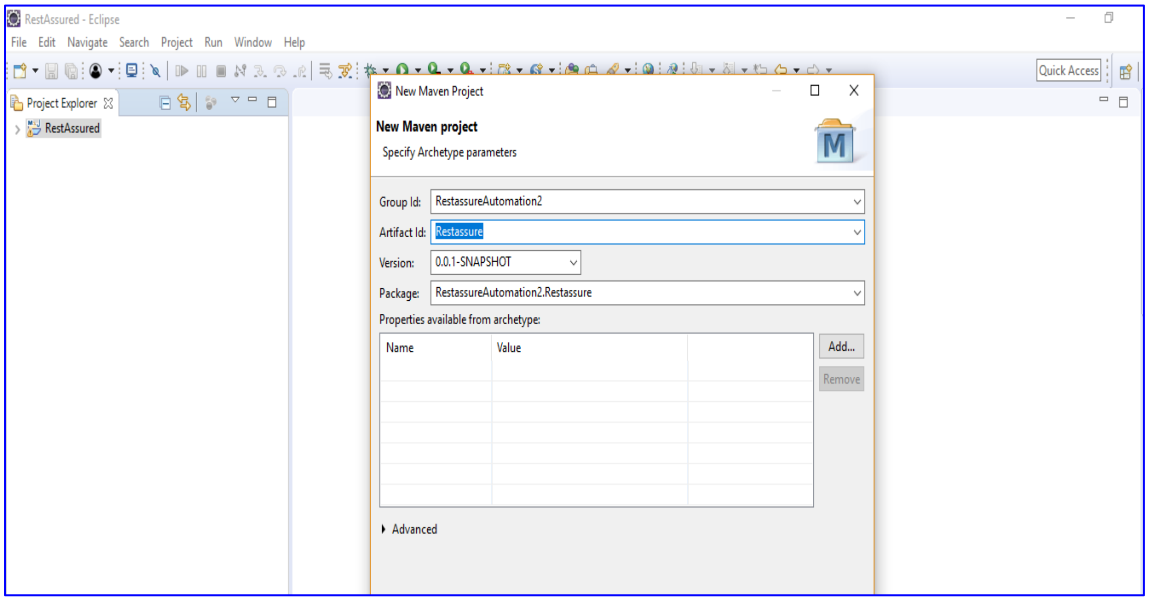
PHASE-3 ASSISTED PRACTICE PROJECT

EMP ID: 2587325

NAME: HIRANMAI

1) Set Up or Configure REST Assured.



New Maven project

Specify Archetype parameters

Group Id: RestassureAutomation2

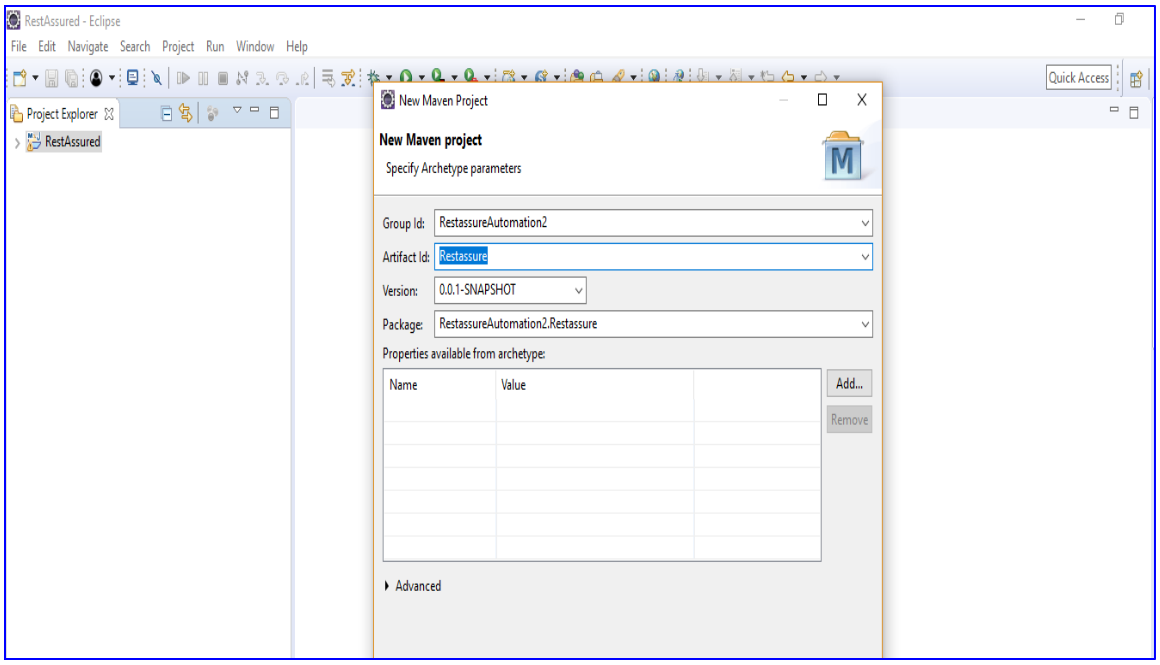
Artifact Id: Restassure

Version: 0.0.1-SNAPSHOT

Package: RestassureAutomation2.Restassure

Properties available from archetype:

Name	Value



New Maven project

Specify Archetype parameters

Group Id: RestassureAutomation2

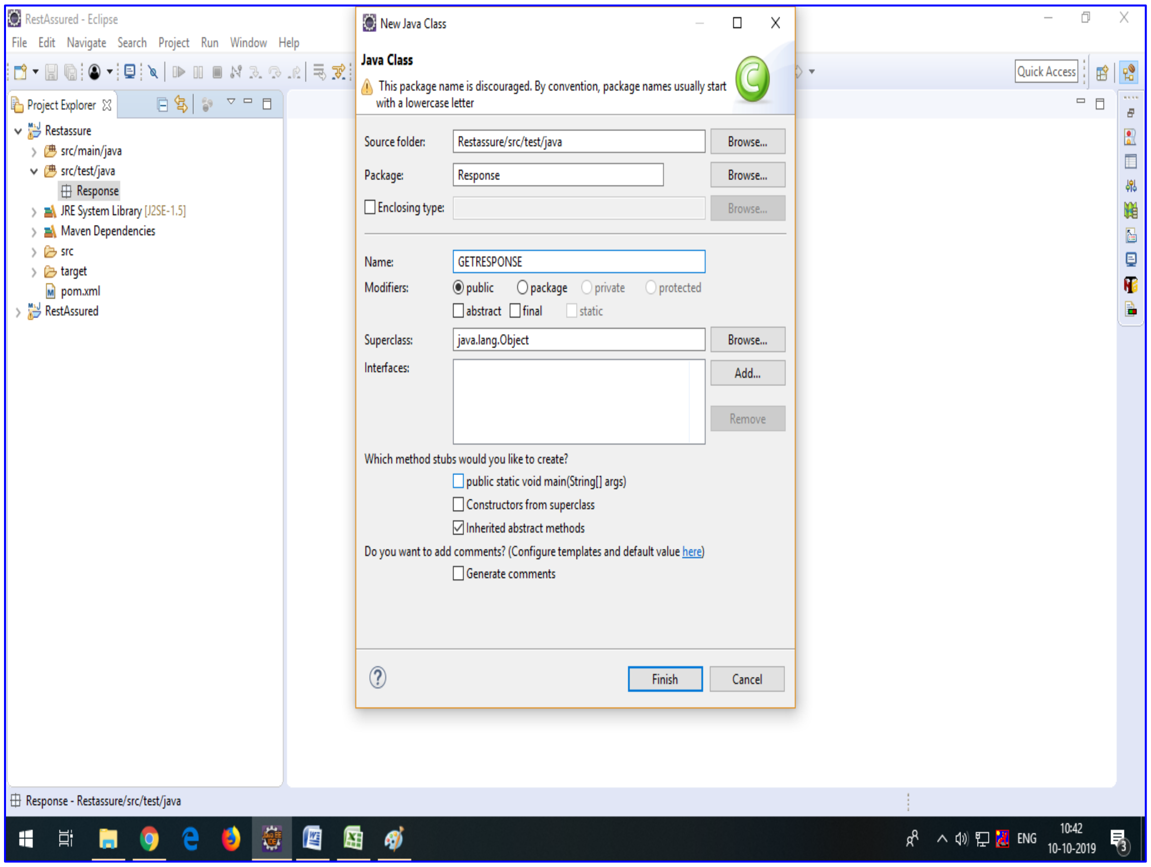
Artifact Id: Restassure

Version: 0.0.1-SNAPSHOT

Package: RestassureAutomation2.Restassure

Properties available from archetype:

Name	Value


New Java Class

Java Class

This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: Restassure/src/test/java

Package: Response

Enclosing type: [ ]

Name: GETRESPONSE

Modifiers: [✓ public] package private protected

[ ] abstract [ ] final [ ] static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

[ ] public static void main(String[] args)

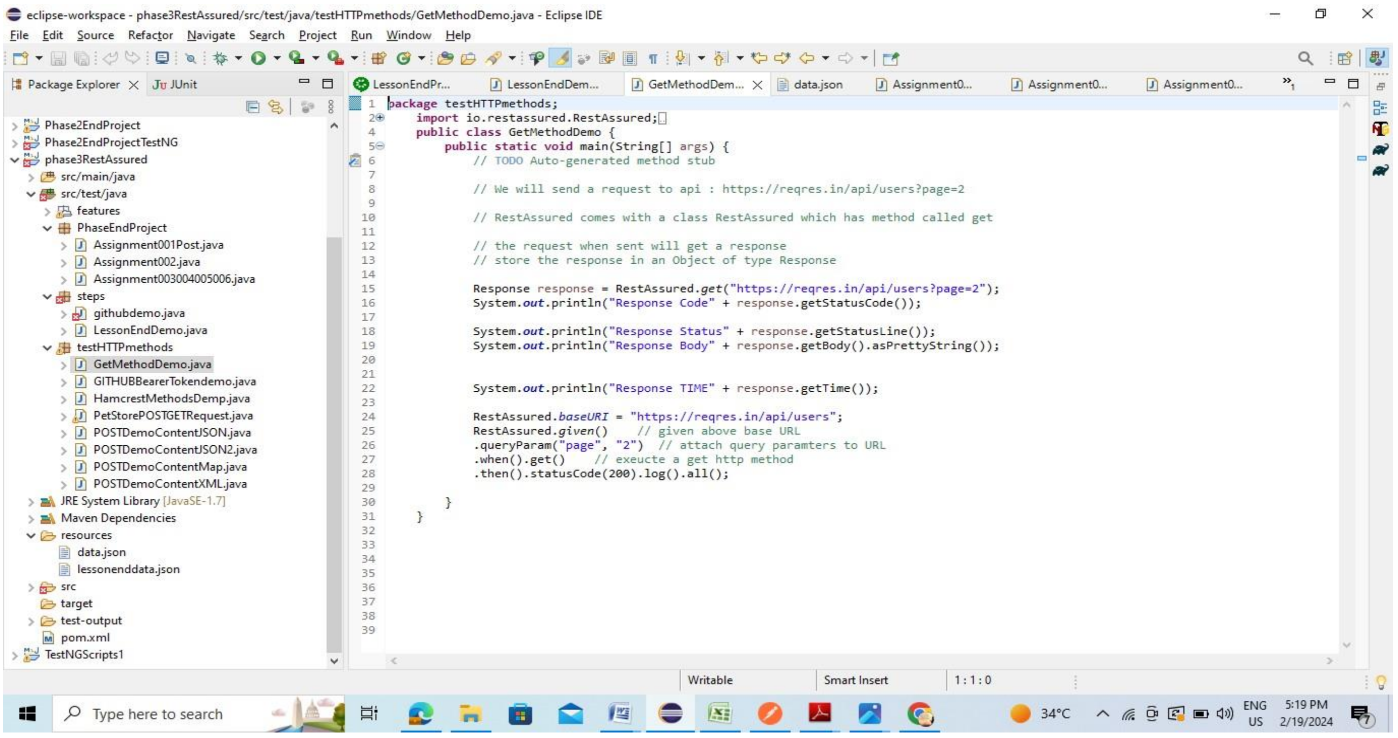
[ ] Constructors from superclass

[✓] Inherited abstract methods

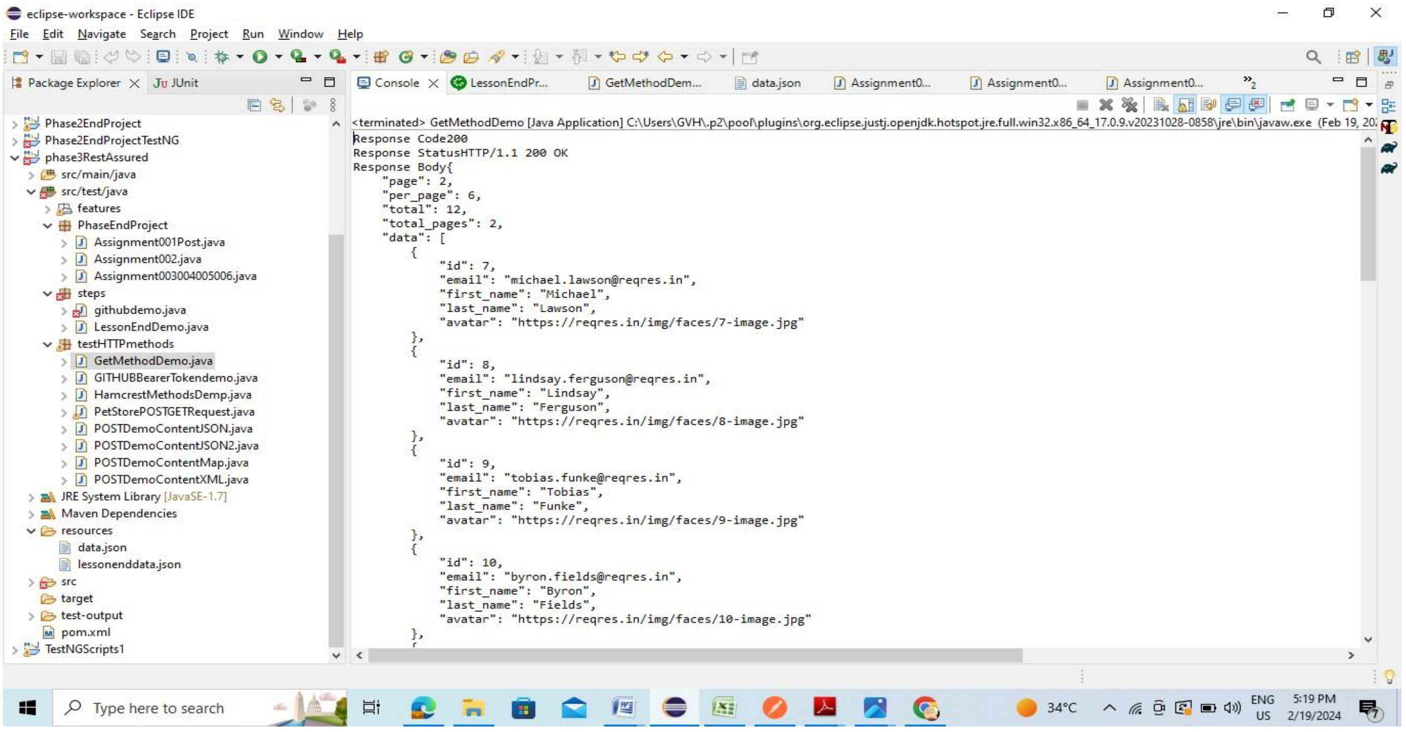
Do you want to add comments? (Configure templates and default value here)

[ ] Generate comments

2) **GET Request and Response Automation.**

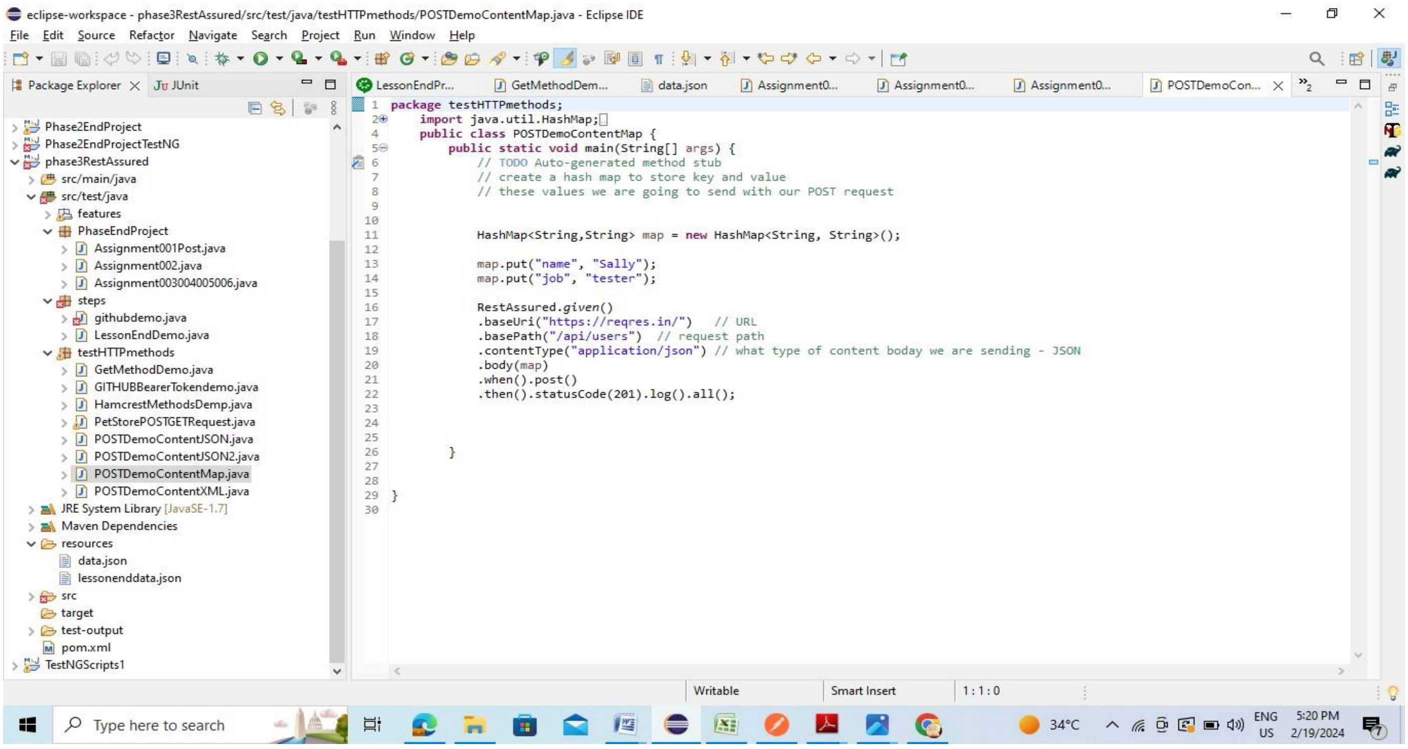


```java
package testHTTPmethods;
	import io.restassured.RestAssured;
	public class GetMethodDemo {
		public static void main(String[] args) {
			// TODO Auto-generated method stub

			// We will send a request to api : https://reqres.in/api/users?page=2

			// RestAssured comes with a class RestAssured which has method called get

			// the request when sent will get a response
			// store the response in an Object of type Response

			Response response = RestAssured.get("https://reqres.in/api/users?page=2");
			System.out.println("Response Code" + response.getStatusCode());

			System.out.println("Response Status" + response.getStatusLine());
			System.out.println("Response Body" + response.getBody().asPrettyString());


			System.out.println("Response TIME" + response.getTime());

			RestAssured.baseURI = "https://reqres.in/api/users";
			RestAssured.given()    // given above base URL
			.queryParam("page", "2")  // attach query paramters to URL
			.when().get()    // exeucte a get http method
			.then().statusCode(200).log().all();

		}
	}
```



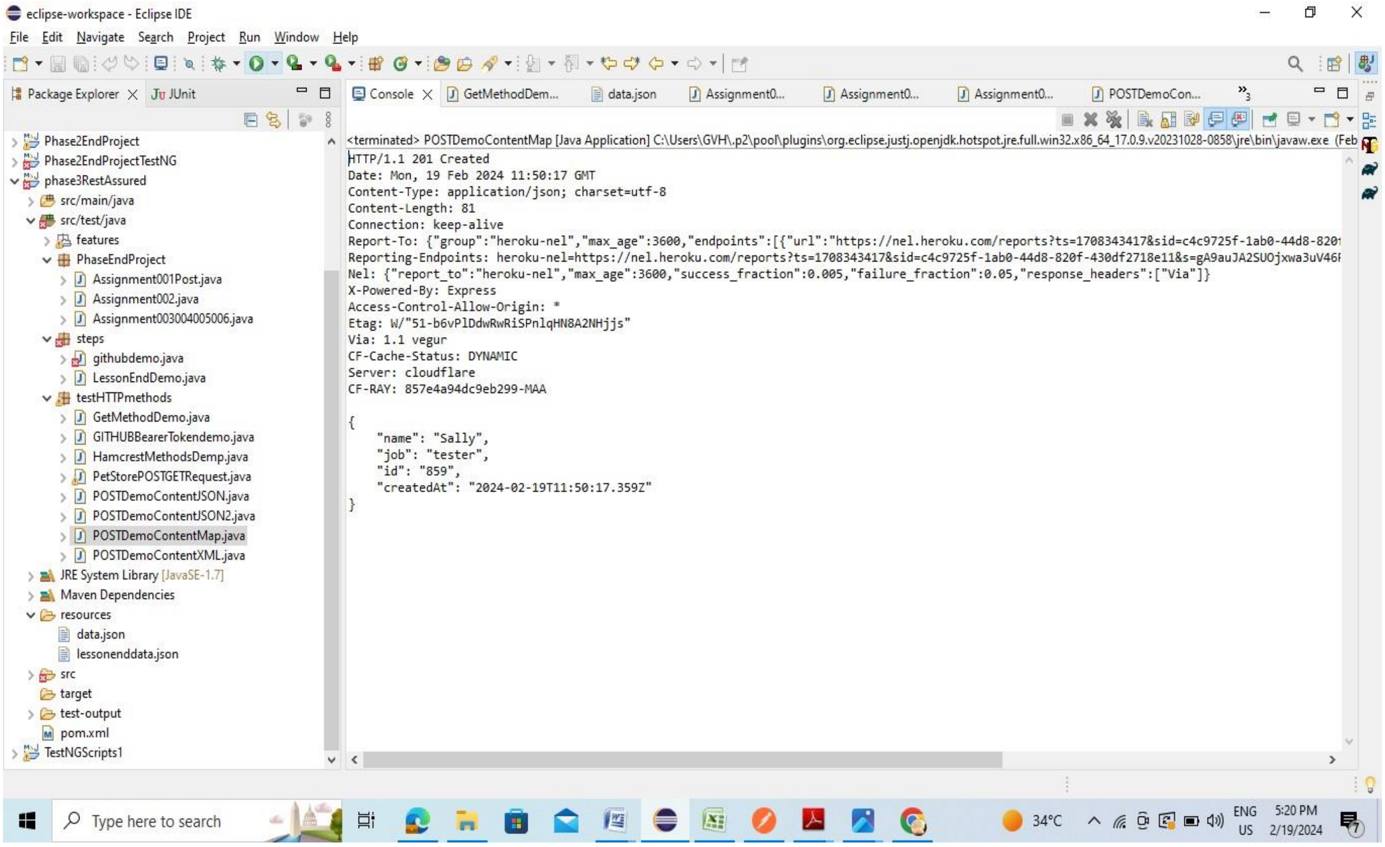
```
<terminated> GetMethodDemo [Java Application] C:\Users\GVH\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (Feb 19, 20
Response Code200
Response StatusHTTP/1.1 200 OK
Response Body{
    "page": 2,
    "per_page": 6,
    "total": 12,
    "total_pages": 2,
    "data": [
        {
            "id": 7,
            "email": "michael.lawson@reqres.in",
            "first_name": "Michael",
            "last_name": "Lawson",
            "avatar": "https://reqres.in/img/faces/7-image.jpg"
        },
        {
            "id": 8,
            "email": "lindsay.ferguson@reqres.in",
            "first_name": "Lindsay",
            "last_name": "Ferguson",
            "avatar": "https://reqres.in/img/faces/8-image.jpg"
        },
        {
            "id": 9,
            "email": "tobias.funke@reqres.in",
            "first_name": "Tobias",
            "last_name": "Funke",
            "avatar": "https://reqres.in/img/faces/9-image.jpg"
        },
        {
            "id": 10,
            "email": "byron.fields@reqres.in",
            "first_name": "Byron",
            "last_name": "Fields",
            "avatar": "https://reqres.in/img/faces/10-image.jpg"
        },
```

3) **POST Request and Response Automation.**



eclipse-workspace - phase3RestAssured/src/test/java/testHTTPmethods/POSTDemoContentMap.java - Eclipse IDE

```java
package testHTTPmethods;
    import java.util.HashMap;
    public class POSTDemoContentMap {
        public static void main(String[] args) {
            // TODO Auto-generated method stub
            // create a hash map to store key and value
            // these values we are going to send with our POST request


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

            map.put("name", "Sally");
            map.put("job", "tester");

            RestAssured.given()
            .baseUri("https://reqres.in/")   // URL
            .basePath("/api/users")  // request path
            .contentType("application/json") // what type of content boday we are sending - JSON
            .body(map)
            .when().post()
            .then().statusCode(201).log().all();



        }

}
```

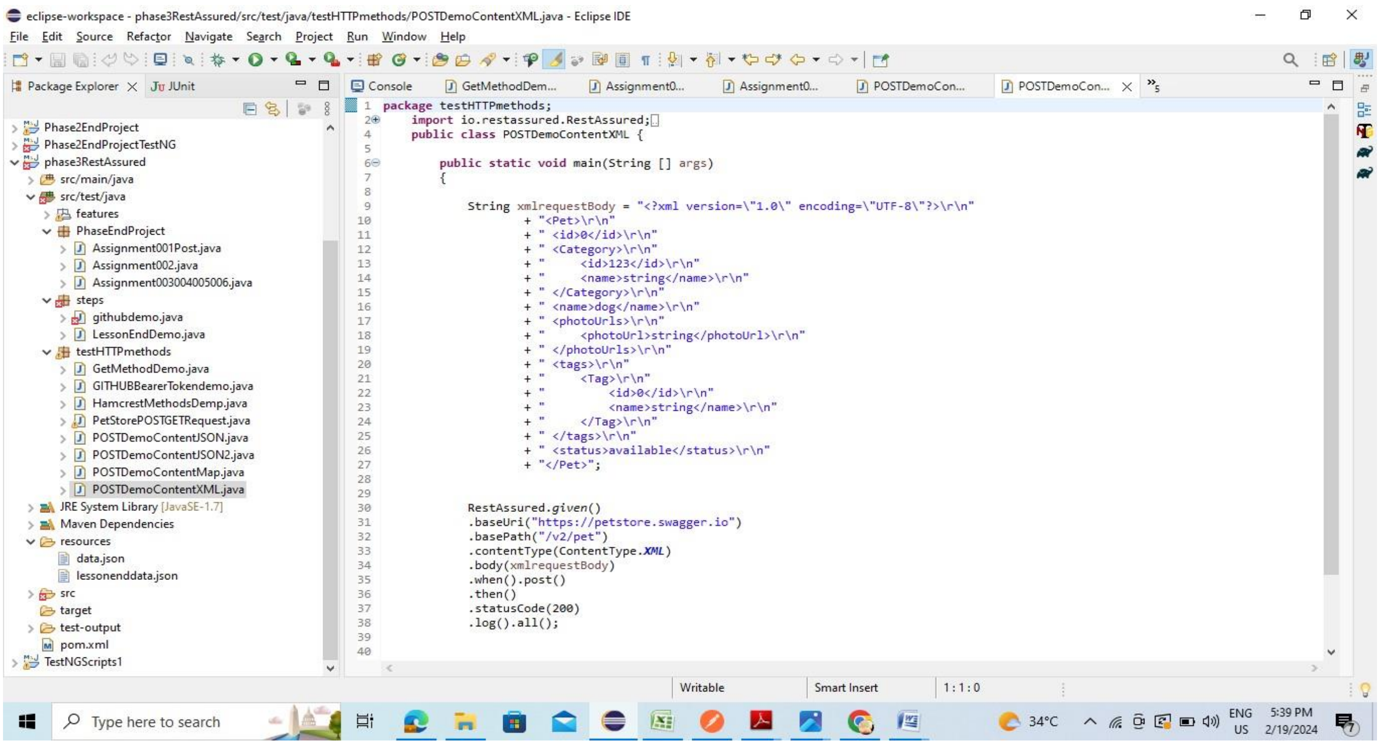


eclipse-workspace - Eclipse IDE

<terminated> POSTDemoContentMap [Java Application] C:\Users\GVH\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (Feb

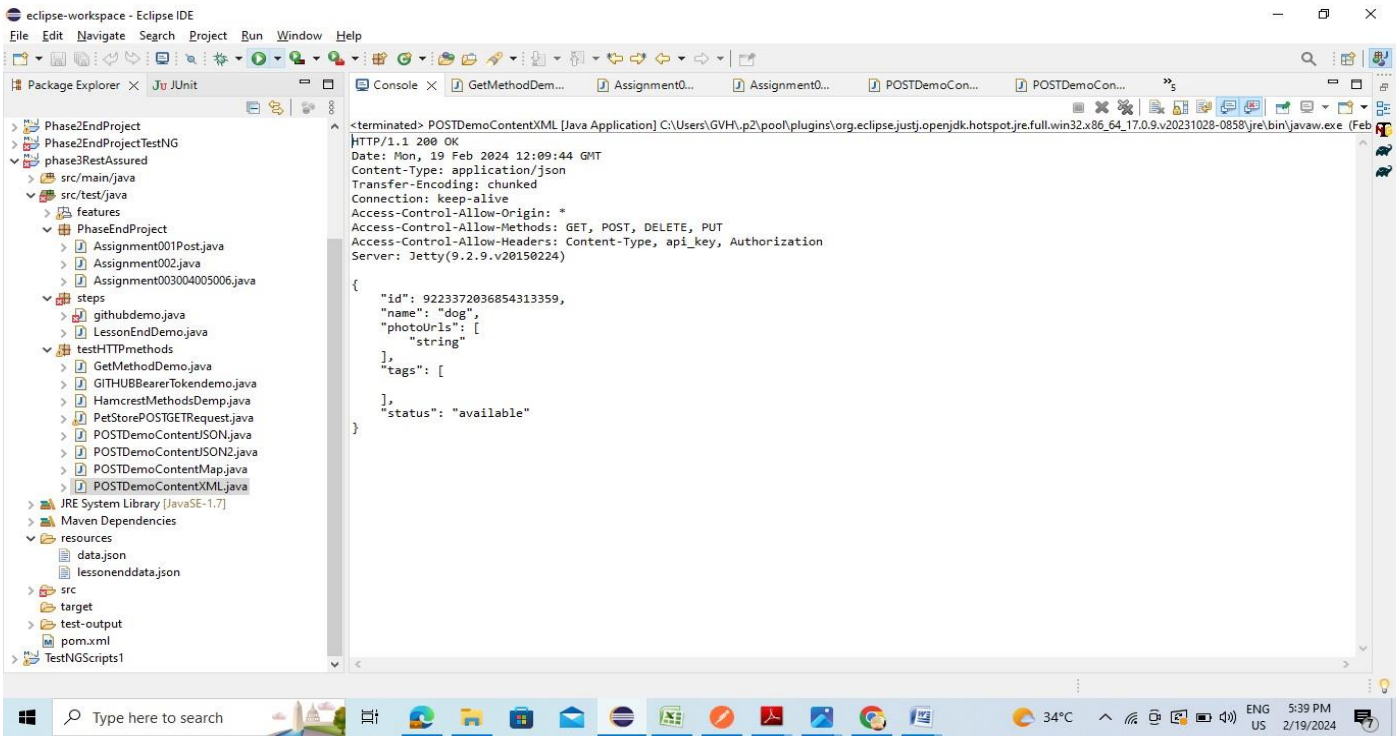
```
HTTP/1.1 201 Created
Date: Mon, 19 Feb 2024 11:50:17 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 81
Connection: keep-alive
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.com/reports?ts=1708343417&sid=c4c9725f-1ab0-44d8-820
Reporting-Endpoints: heroku-nel=https://nel.heroku.com/reports?ts=1708343417&sid=c4c9725f-1ab0-44d8-820f-430df2718e11&s=gA9auJA2SUOjxwa3uV46
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/"51-b6vPlDdwRwRiSPnlqHN8A2NHjjs"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 857e4a94dc9eb299-MAA

{
    "name": "Sally",
    "job": "tester",
    "id": "859",
    "createdAt": "2024-02-19T11:50:17.359Z"
}
```

4) Send XML Payload in REST Assured, Parse XML Response in REST Assured, Parse JSON Response in REST Assured, Explore Native Logging of REST Assured.



eclipse-workspace - phase3RestAssured/src/test/java/testHTTPmethods/POSTDemoContentXML.java - Eclipse IDE

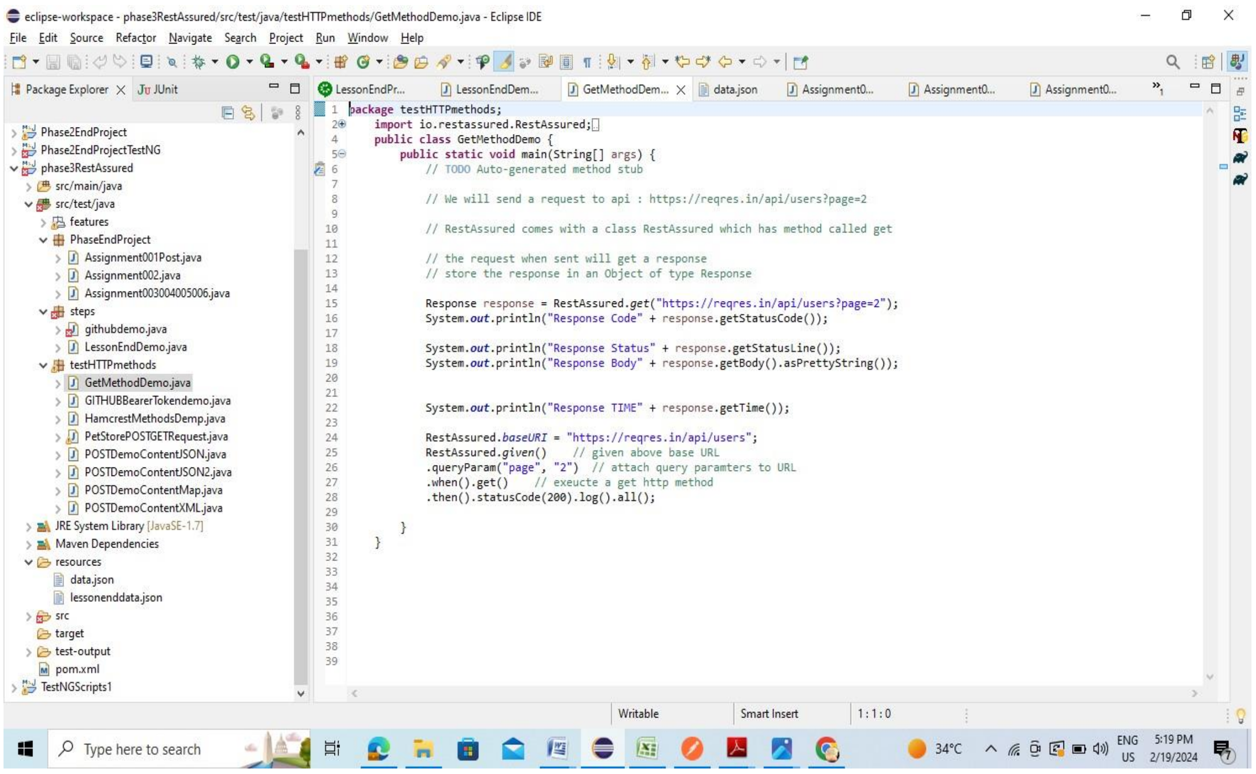
```java
package testHTTPmethods;
    import io.restassured.RestAssured;
    public class POSTDemoContentXML {

        public static void main(String [] args)
        {

            String xmlrequestBody = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n"
                    + "<Pet>\r\n"
                    + " <id>0</id>\r\n"
                    + " <Category>\r\n"
                    + "     <id>123</id>\r\n"
                    + "     <name>string</name>\r\n"
                    + " </Category>\r\n"
                    + " <name>dog</name>\r\n"
                    + " <photoUrls>\r\n"
                    + "     <photoUrl>string</photoUrl>\r\n"
                    + " </photoUrls>\r\n"
                    + " <tags>\r\n"
                    + "     <Tag>\r\n"
                    + "         <id>0</id>\r\n"
                    + "         <name>string</name>\r\n"
                    + "     </Tag>\r\n"
                    + " </tags>\r\n"
                    + " <status>available</status>\r\n"
                    + "</Pet>";


            RestAssured.given()
            .baseUri("https://petstore.swagger.io")
            .basePath("/v2/pet")
            .contentType(ContentType.XML)
            .body(xmlrequestBody)
            .when().post()
            .then()
            .statusCode(200)
            .log().all();
```



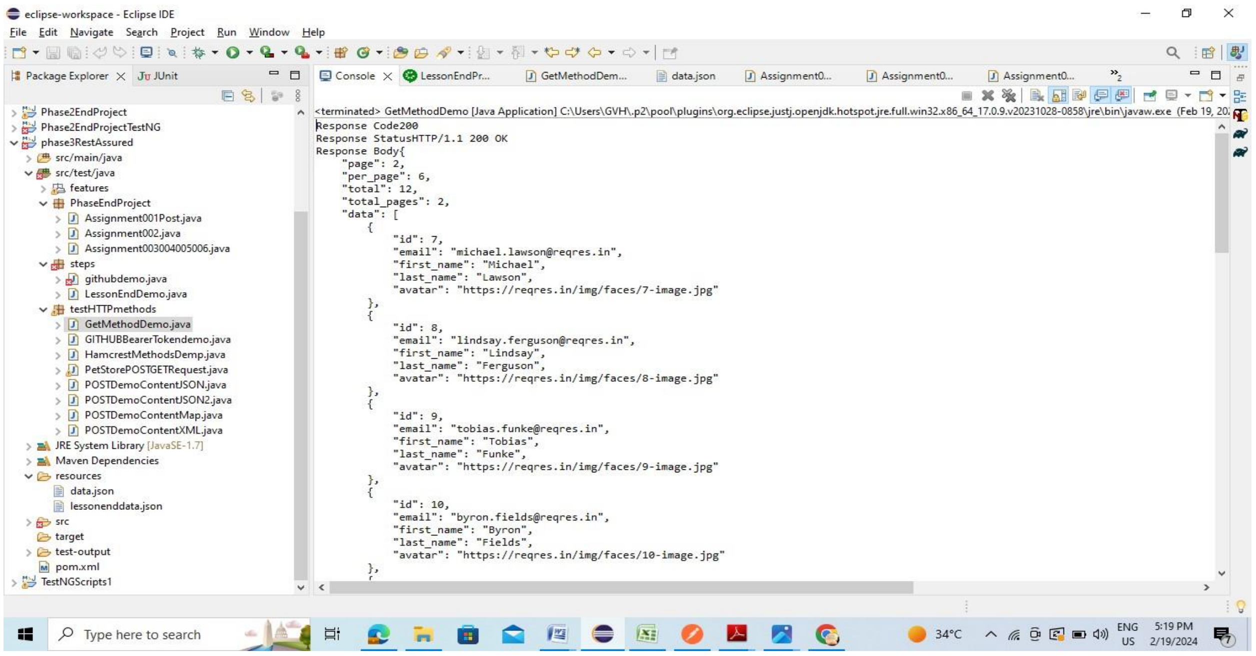
eclipse-workspace - Eclipse IDE

```
<terminated> POSTDemoContentXML [Java Application] C:\Users\GVH\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (Feb
HTTP/1.1 200 OK
Date: Mon, 19 Feb 2024 12:09:44 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Server: Jetty(9.2.9.v20150224)

{
    "id": 9223372036854313359,
    "name": "dog",
    "photoUrls": [
        "string"
    ],
    "tags": [

    ],
    "status": "available"
}
```

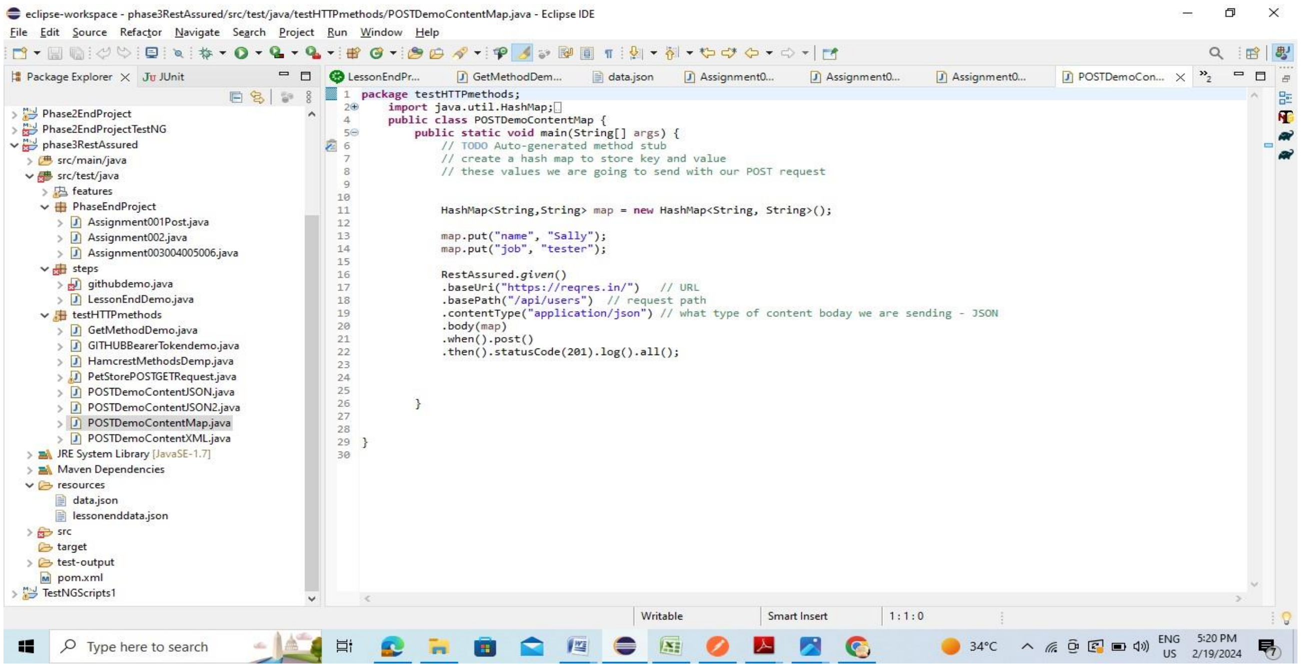
5) GET, POST, XML, and JSON.



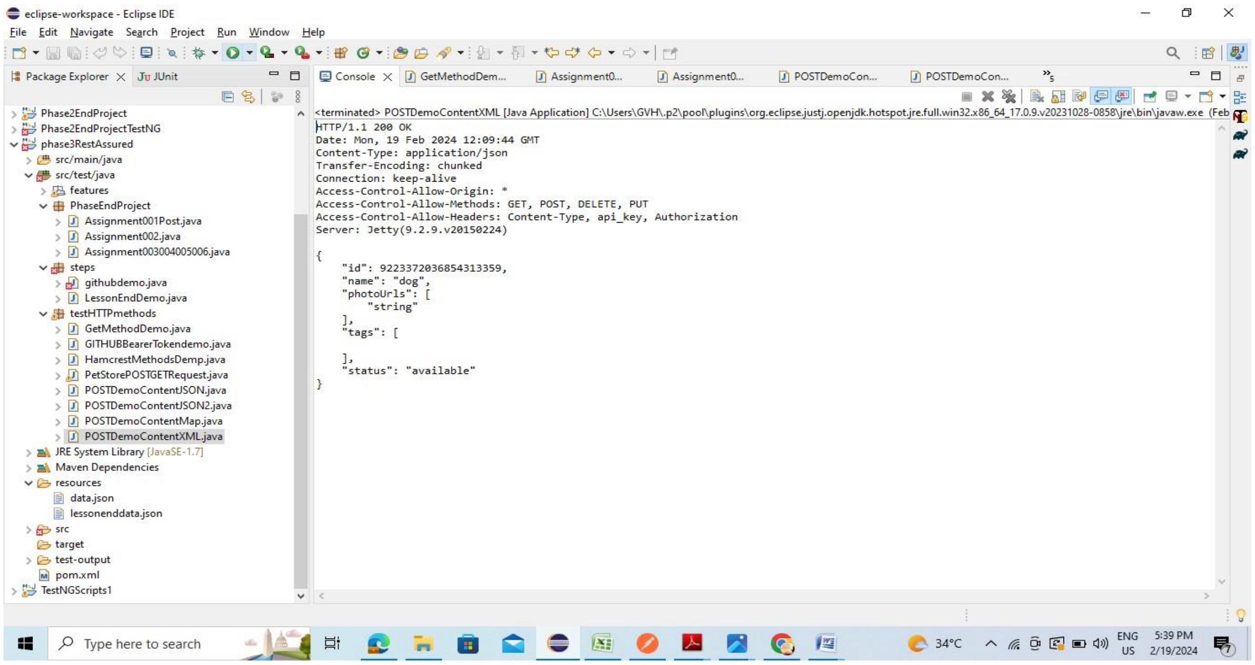
```java
package testHTTPmethods;
    import io.restassured.RestAssured;
    public class GetMethodDemo {
        public static void main(String[] args) {
            // TODO Auto-generated method stub

            // We will send a request to api : https://reqres.in/api/users?page=2

            // RestAssured comes with a class RestAssured which has method called get

            // the request when sent will get a response
            // store the response in an Object of type Response

            Response response = RestAssured.get("https://reqres.in/api/users?page=2");
            System.out.println("Response Code" + response.getStatusCode());

            System.out.println("Response Status" + response.getStatusLine());
            System.out.println("Response Body" + response.getBody().asPrettyString());


            System.out.println("Response TIME" + response.getTime());

            RestAssured.baseURI = "https://reqres.in/api/users";
            RestAssured.given()    // given above base URL
            .queryParam("page", "2")  // attach query paramters to URL
            .when().get()    // exeucte a get http method
            .then().statusCode(200).log().all();

        }
    }
```



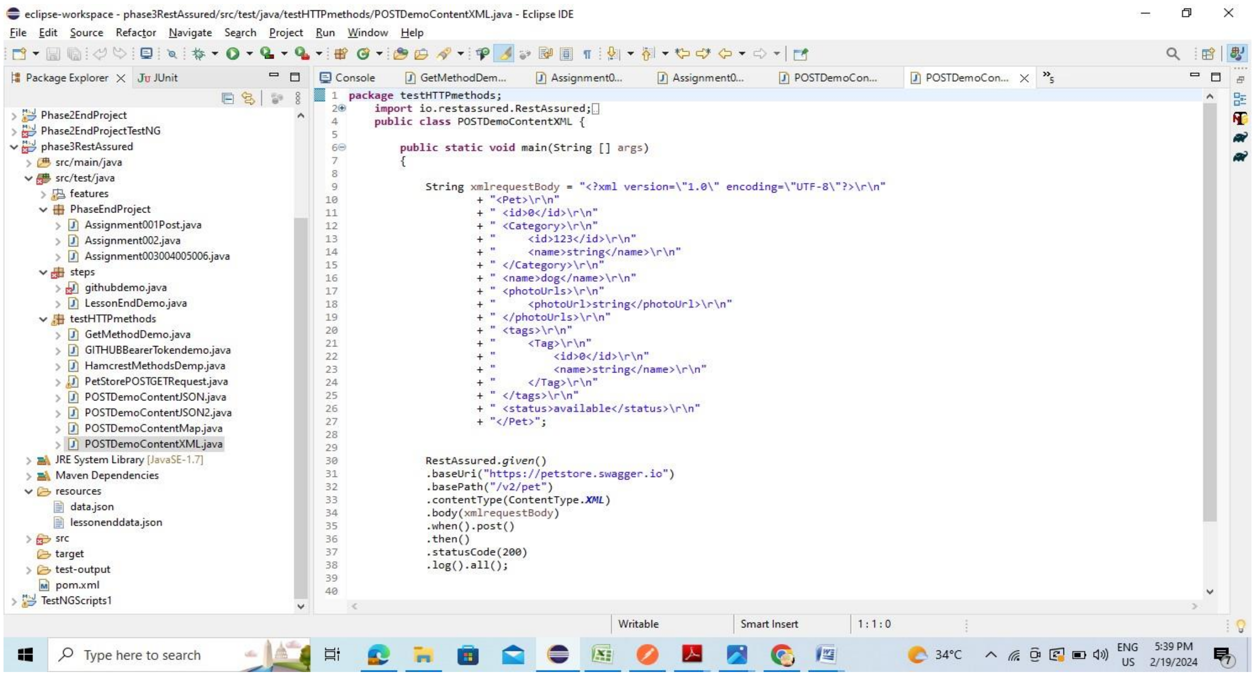
```
<terminated> GetMethodDemo [Java Application] C:\Users\GVH\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (Feb 19, 20
Response Code200
Response StatusHTTP/1.1 200 OK
Response Body{
    "page": 2,
    "per_page": 6,
    "total": 12,
    "total_pages": 2,
    "data": [
        {
            "id": 7,
            "email": "michael.lawson@reqres.in",
            "first_name": "Michael",
            "last_name": "Lawson",
            "avatar": "https://reqres.in/img/faces/7-image.jpg"
        },
        {
            "id": 8,
            "email": "lindsay.ferguson@reqres.in",
            "first_name": "Lindsay",
            "last_name": "Ferguson",
            "avatar": "https://reqres.in/img/faces/8-image.jpg"
        },
        {
            "id": 9,
            "email": "tobias.funke@reqres.in",
            "first_name": "Tobias",
            "last_name": "Funke",
            "avatar": "https://reqres.in/img/faces/9-image.jpg"
        },
        {
            "id": 10,
            "email": "byron.fields@reqres.in",
            "first_name": "Byron",
            "last_name": "Fields",
            "avatar": "https://reqres.in/img/faces/10-image.jpg"
        },
```


eclipse-workspace - phase3RestAssured/src/test/java/testHTTPmethods/POSTDemoContentMap.java - Eclipse IDE

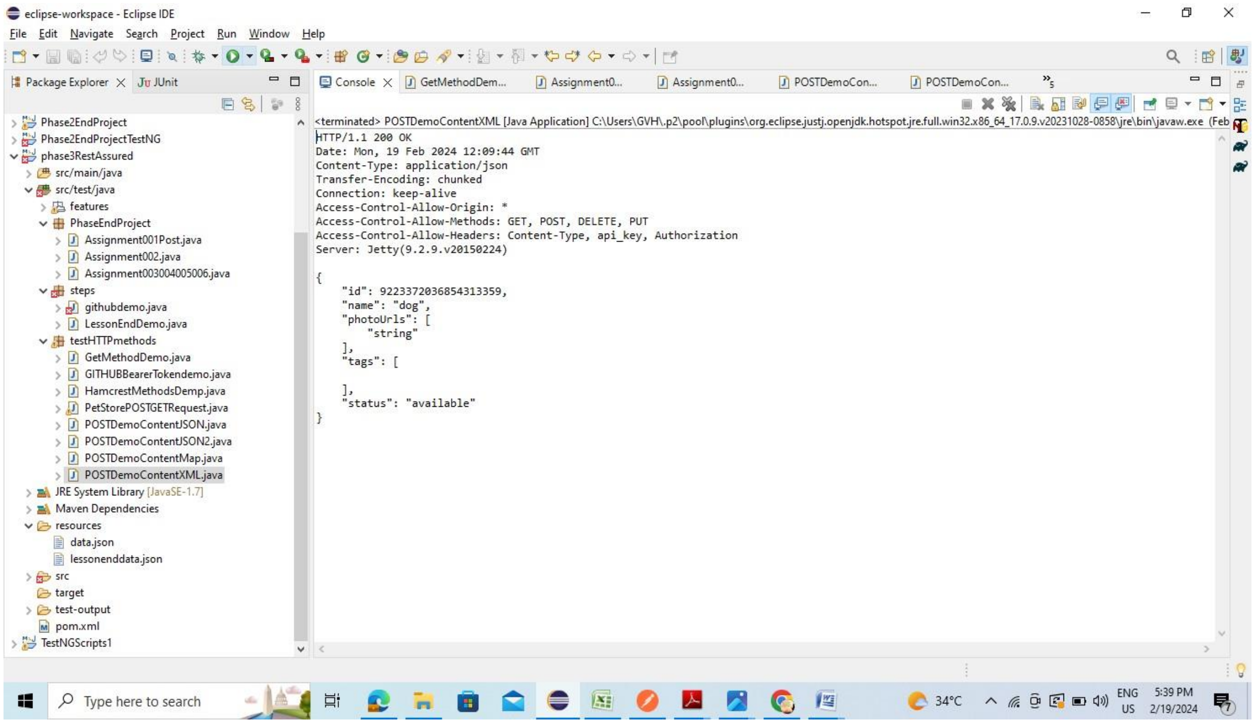
```java
package testHTTPmethods;
    import java.util.HashMap;
    public class POSTDemoContentMap {
        public static void main(String[] args) {
            // TODO Auto-generated method stub
            // create a hash map to store key and value
            // these values we are going to send with our POST request


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

            map.put("name", "Sally");
            map.put("job", "tester");

            RestAssured.given()
            .baseUri("https://reqres.in/")   // URL
            .basePath("/api/users")  // request path
            .contentType("application/json") // what type of content boday we are sending - JSON
            .body(map)
            .when().post()
            .then().statusCode(201).log().all();



        }


}
```


eclipse-workspace - Eclipse IDE

```
<terminated> POSTDemoContentXML [Java Application] C:\Users\GVH\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (Feb
HTTP/1.1 200 OK
Date: Mon, 19 Feb 2024 12:09:44 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Server: Jetty(9.2.9.v20150224)

{
    "id": 9223372036854313359,
    "name": "dog",
    "photoUrls": [
        "string"
    ],
    "tags": [

    ],
    "status": "available"
}
```

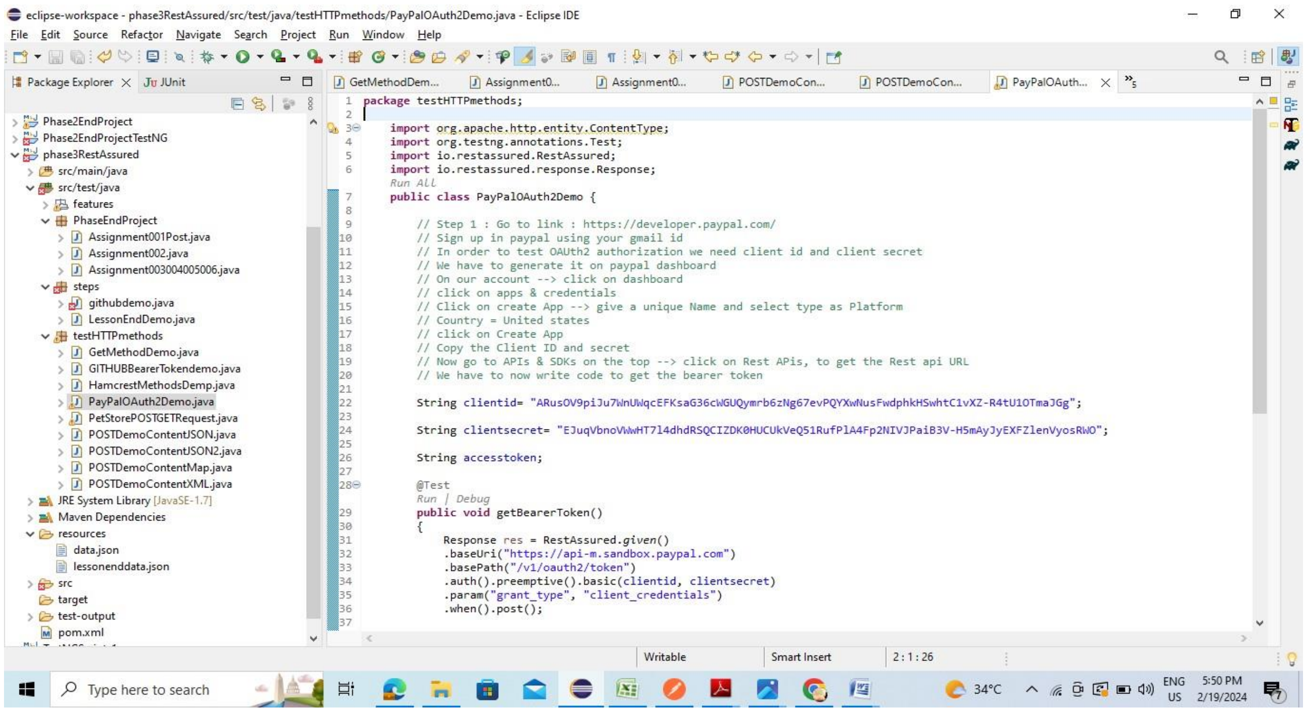

eclipse-workspace - phase3RestAssured/src/test/java/testHTTPmethods/POSTDemoContentXML.java - Eclipse IDE

```java
package testHTTPmethods;
    import io.restassured.RestAssured;
    public class POSTDemoContentXML {

        public static void main(String [] args)
        {

            String xmlrequestBody = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n"
                    + "<Pet>\r\n"
                    + " <id>0</id>\r\n"
                    + " <Category>\r\n"
                    + "     <id>123</id>\r\n"
                    + "     <name>string</name>\r\n"
                    + " </Category>\r\n"
                    + " <name>dog</name>\r\n"
                    + " <photoUrls>\r\n"
                    + "     <photoUrl>string</photoUrl>\r\n"
                    + " </photoUrls>\r\n"
                    + " <tags>\r\n"
                    + "     <Tag>\r\n"
                    + "         <id>0</id>\r\n"
                    + "         <name>string</name>\r\n"
                    + "     </Tag>\r\n"
                    + " </tags>\r\n"
                    + " <status>available</status>\r\n"
                    + "</Pet>";


            RestAssured.given()
            .baseUri("https://petstore.swagger.io")
            .basePath("/v2/pet")
            .contentType(ContentType.XML)
            .body(xmlrequestBody)
            .when().post()
            .then()
            .statusCode(200)
            .log().all();
```


eclipse-workspace - Eclipse IDE

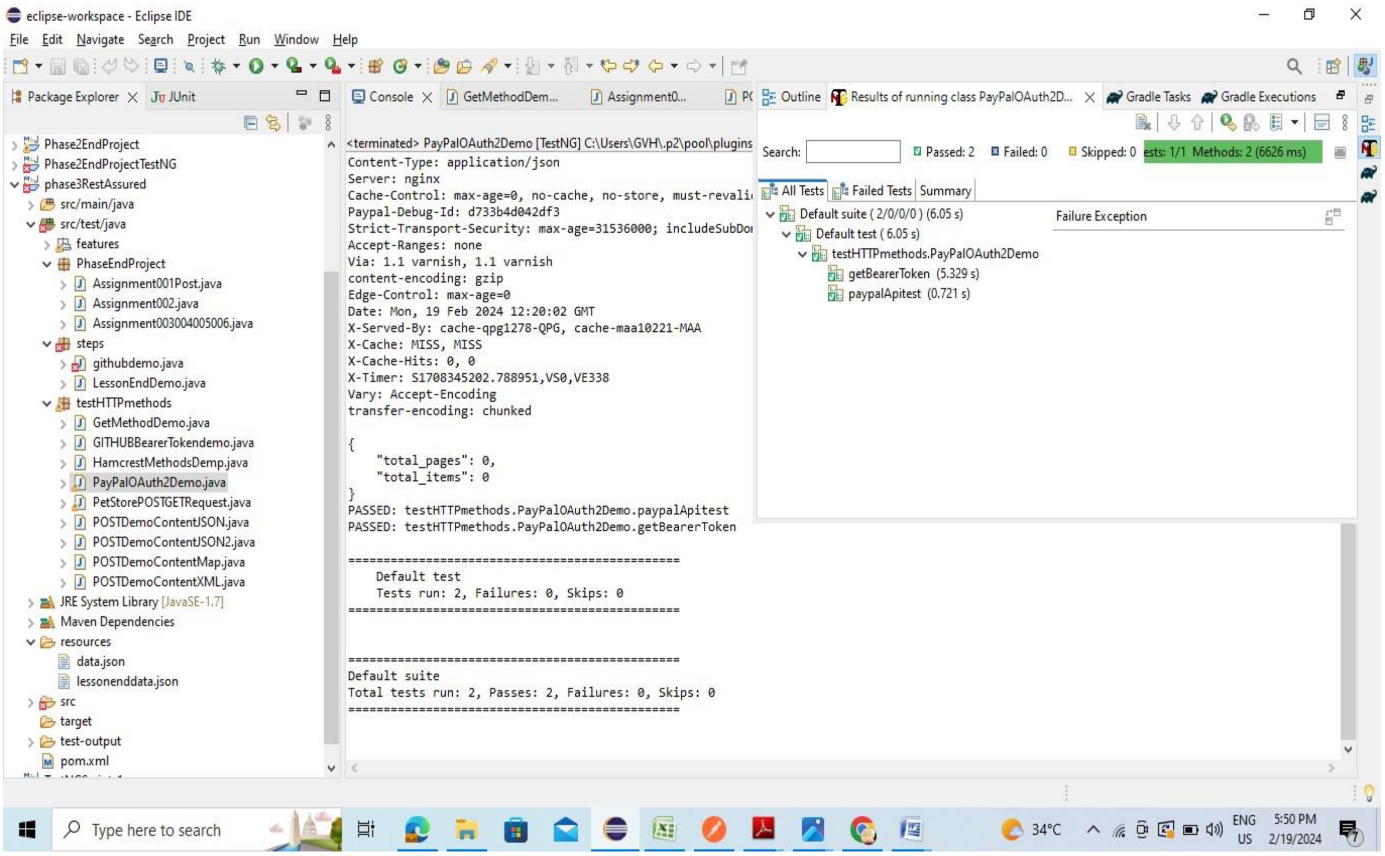
```
<terminated> POSTDemoContentXML [Java Application] C:\Users\GVH\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (Feb
HTTP/1.1 200 OK
Date: Mon, 19 Feb 2024 12:09:44 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET, POST, DELETE, PUT
Access-Control-Allow-Headers: Content-Type, api_key, Authorization
Server: Jetty(9.2.9.v20150224)

{
    "id": 9223372036854313359,
    "name": "dog",
    "photoUrls": [
        "string"
    ],
    "tags": [

    ],
    "status": "available"
}
```

6) Handle OAuth Authorization in REST API Using REST Assured.



eclipse-workspace - phase3RestAssured/src/test/java/testHTTPmethods/PayPalOAuth2Demo.java - Eclipse IDE

```java
package testHTTPmethods;

import org.apache.http.entity.ContentType;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
import io.restassured.response.Response;
Run All
public class PayPalOAuth2Demo {

    // Step 1 : Go to link : https://developer.paypal.com/
    // Sign up in paypal using your gmail id
    // In order to test OAUth2 authorization we need client id and client secret
    // We have to generate it on paypal dashboard
    // On our account --> click on dashboard
    // click on apps & credentials
    // Click on create App --> give a unique Name and select type as Platform
    // Country = United states
    // click on Create App
    // Copy the Client ID and secret
    // Now go to APIs & SDKs on the top --> click on Rest APis, to get the Rest api URL
    // We have to now write code to get the bearer token

    String clientid= "ARusOV9piJu7WnUWqcEFKsaG36cWGUQymrb6zNg67evPQYXwNusFwdphkHSwhtC1vXZ-R4tU1OTmaJGg";

    String clientsecret= "EJuqVbnoVWwHT7l4dhdRSQCIZDK0HUCUkVeQ51RufPlA4Fp2NIVJPaiB3V-H5mAyJyEXFZlenVyosRWO";

    String accesstoken;

    @Test
    Run | Debug
    public void getBearerToken()
    {
        Response res = RestAssured.given()
        .baseUri("https://api-m.sandbox.paypal.com")
        .basePath("/v1/oauth2/token")
        .auth().preemptive().basic(clientid, clientsecret)
        .param("grant_type", "client_credentials")
        .when().post();
```



eclipse-workspace - Eclipse IDE

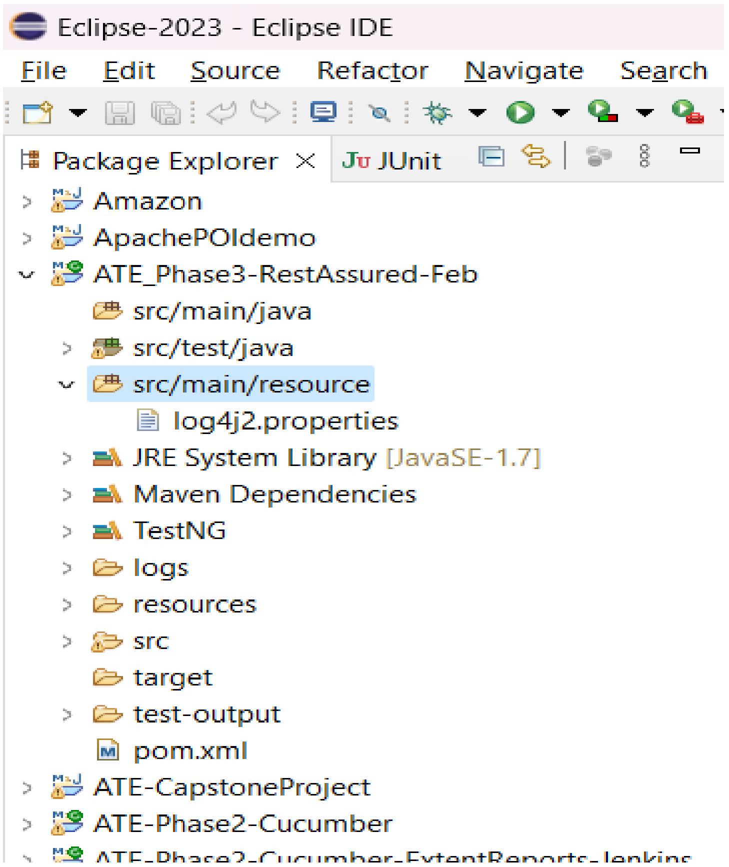
```
<terminated> PayPalOAuth2Demo [TestNG] C:\Users\GVH\.p2\pool\plugins
Content-Type: application/json
Server: nginx
Cache-Control: max-age=0, no-cache, no-store, must-revali
Paypal-Debug-Id: d733b4d042df3
Strict-Transport-Security: max-age=31536000; includeSubDo
Accept-Ranges: none
Via: 1.1 varnish, 1.1 varnish
content-encoding: gzip
Edge-Control: max-age=0
Date: Mon, 19 Feb 2024 12:20:02 GMT
X-Served-By: cache-qpg1278-QPG, cache-maa10221-MAA
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1708345202.788951,VS0,VE338
Vary: Accept-Encoding
transfer-encoding: chunked

{
    "total_pages": 0,
    "total_items": 0
}
PASSED: testHTTPmethods.PayPalOAuth2Demo.paypalApitest
PASSED: testHTTPmethods.PayPalOAuth2Demo.getBearerToken

===============================================
    Default test
    Tests run: 2, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================
```

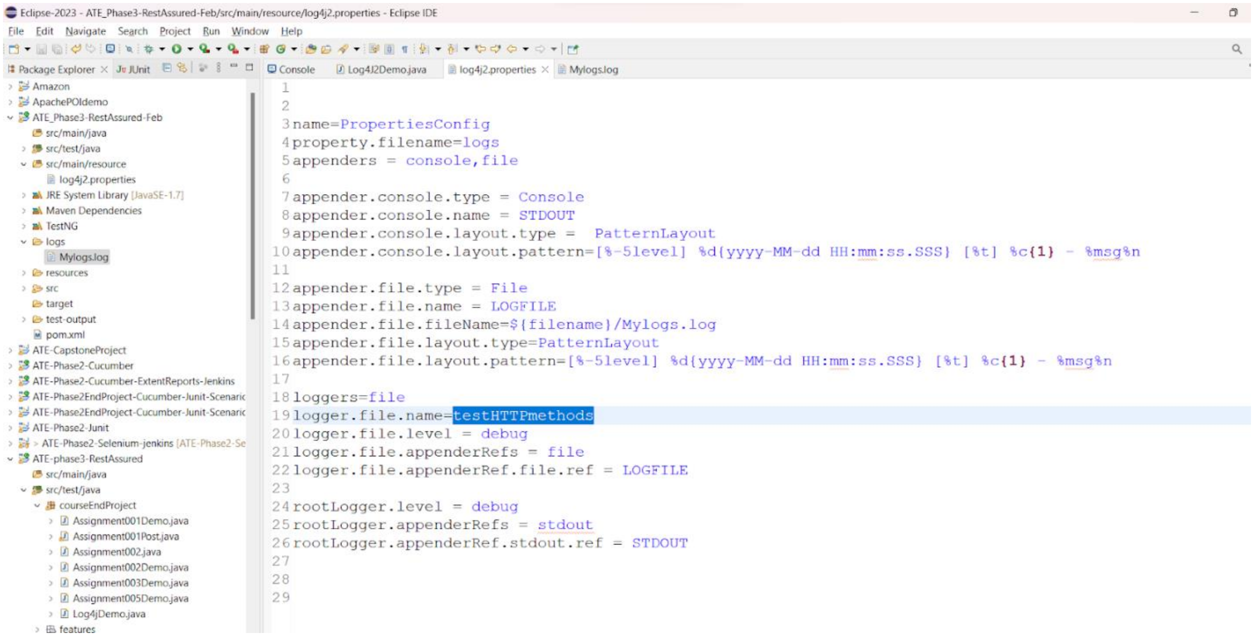
Results of running class PayPalOAuth2Demo

Passed: 2 Failed: 0 Skipped: 0 Tests: 1/1 Methods: 2 (6626 ms)

```
Default suite ( 2/0/0/0 ) (6.05 s)
  Default test ( 6.05 s)
    testHTTPmethods.PayPalOAuth2Demo
      getBearerToken (5.329 s)
      paypalApitest (0.721 s)
```

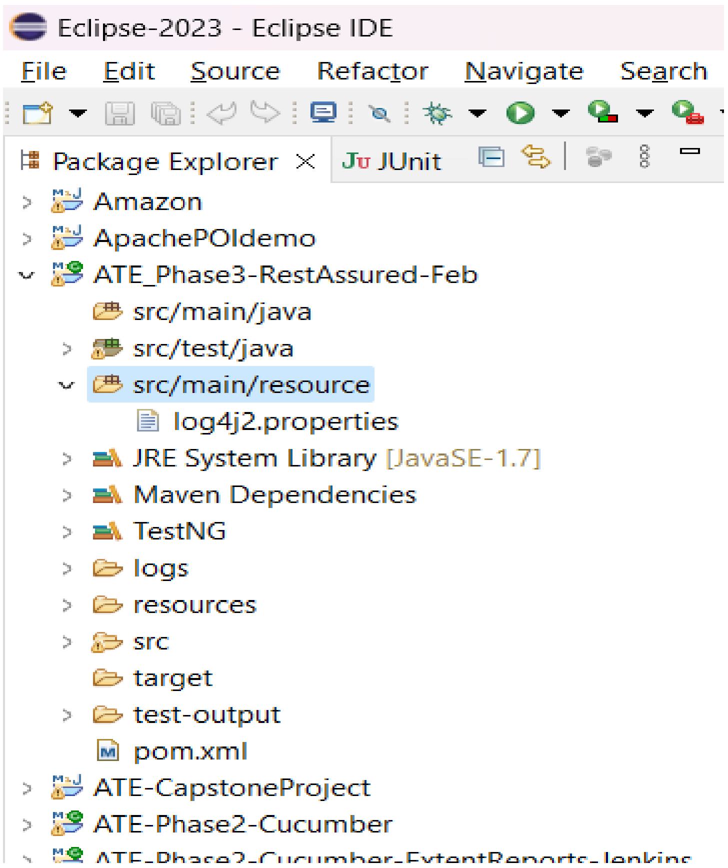
7) Configure Log4j in Eclipse Maven Project.

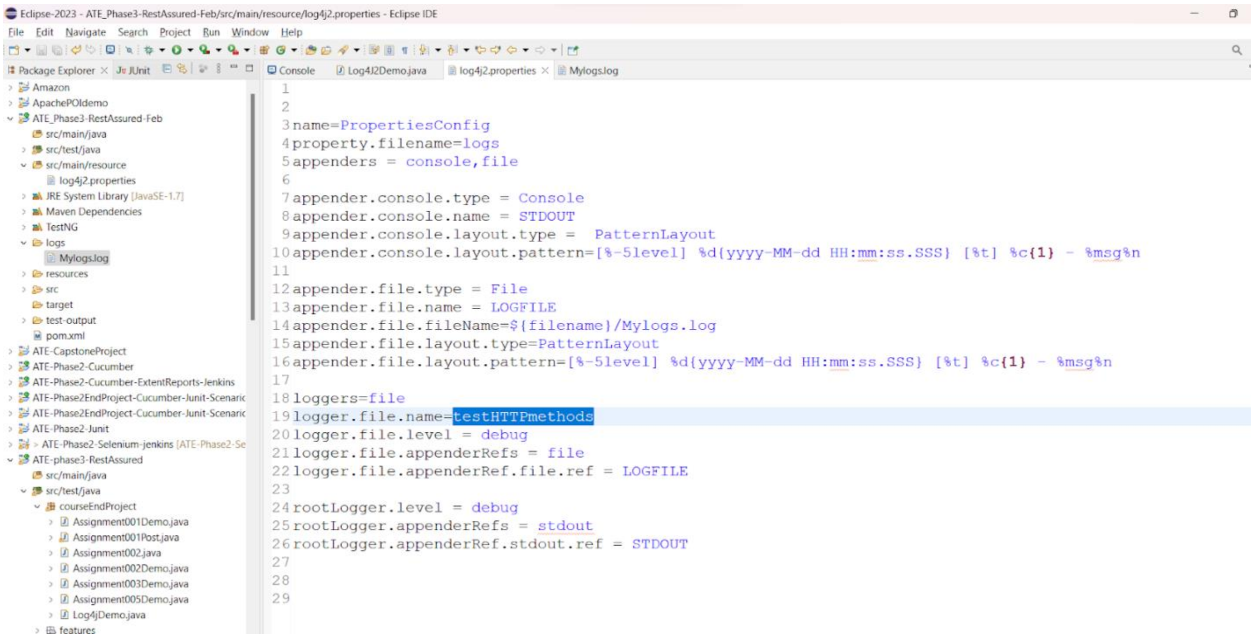



Eclipse-2023 - ATE_Phase3-RestAssured-Feb/src/main/resource/log4j2.properties - Eclipse IDE
Console
Log4J2Demo.java
log4j2.properties
Mylogs.log

```
name=PropertiesConfig
property.filename=logs
appenders = console,file

appender.console.type = Console
appender.console.name = STDOUT
appender.console.layout.type =  PatternLayout
appender.console.layout.pattern=[%-5level] %d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %c{1} - %msg%n

appender.file.type = File
appender.file.name = LOGFILE
appender.file.fileName=${filename}/Mylogs.log
appender.file.layout.type=PatternLayout
appender.file.layout.pattern=[%-5level] %d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %c{1} - %msg%n

loggers=file
logger.file.name=testHTTPmethods
logger.file.level = debug
logger.file.appenderRefs = file
logger.file.appenderRef.file.ref = LOGFILE

rootLogger.level = debug
rootLogger.appenderRefs = stdout
rootLogger.appenderRef.stdout.ref = STDOUT
```

8) OAuth, SSL, and Log4j.




Eclipse-2023 - ATE_Phase3-RestAssured-Feb/src/main/resource/log4j2.properties - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Console
Log4J2Demo.java
log4j2.properties
Mylogs.log

```


name=PropertiesConfig
property.filename=logs
appenders = console,file

appender.console.type = Console
appender.console.name = STDOUT
appender.console.layout.type =  PatternLayout
appender.console.layout.pattern=[%-5level] %d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %c{1} - %msg%n

appender.file.type = File
appender.file.name = LOGFILE
appender.file.fileName=${filename}/Mylogs.log
appender.file.layout.type=PatternLayout
appender.file.layout.pattern=[%-5level] %d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %c{1} - %msg%n

loggers=file
logger.file.name=testHTTPmethods
logger.file.level = debug
logger.file.appenderRefs = file
logger.file.appenderRef.file.ref = LOGFILE

rootLogger.level = debug
rootLogger.appenderRefs = stdout
rootLogger.appenderRef.stdout.ref = STDOUT
```