Source code:

Postman Assignment 001:

Create pet Id and pet name:

First create excel sheet and enter pet names pet id

A	B	C	D	E	F
petid	Pet name				
110	cat				
120	dog				
130	Tiger				
140	bird				
150	Lion				
160	Donkey				
170	cow				
180	doggie				
190	jauguar				
200	hen				
210	goat				
220	horse				
230	bear				
240	sheep				
250	rat				
260	mice				
270	fish				
280	ant				
290	bee				

And save it as project pet id select as csv & delimeted & save as in desktop folder

● Open Postman
● Create a new collection called **Phase end project** and arrange three end-to-end services
as shown below:
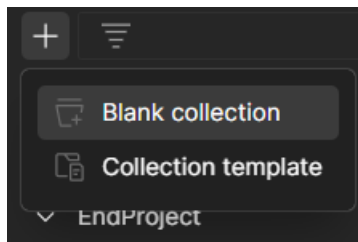
In this blank collection add new request
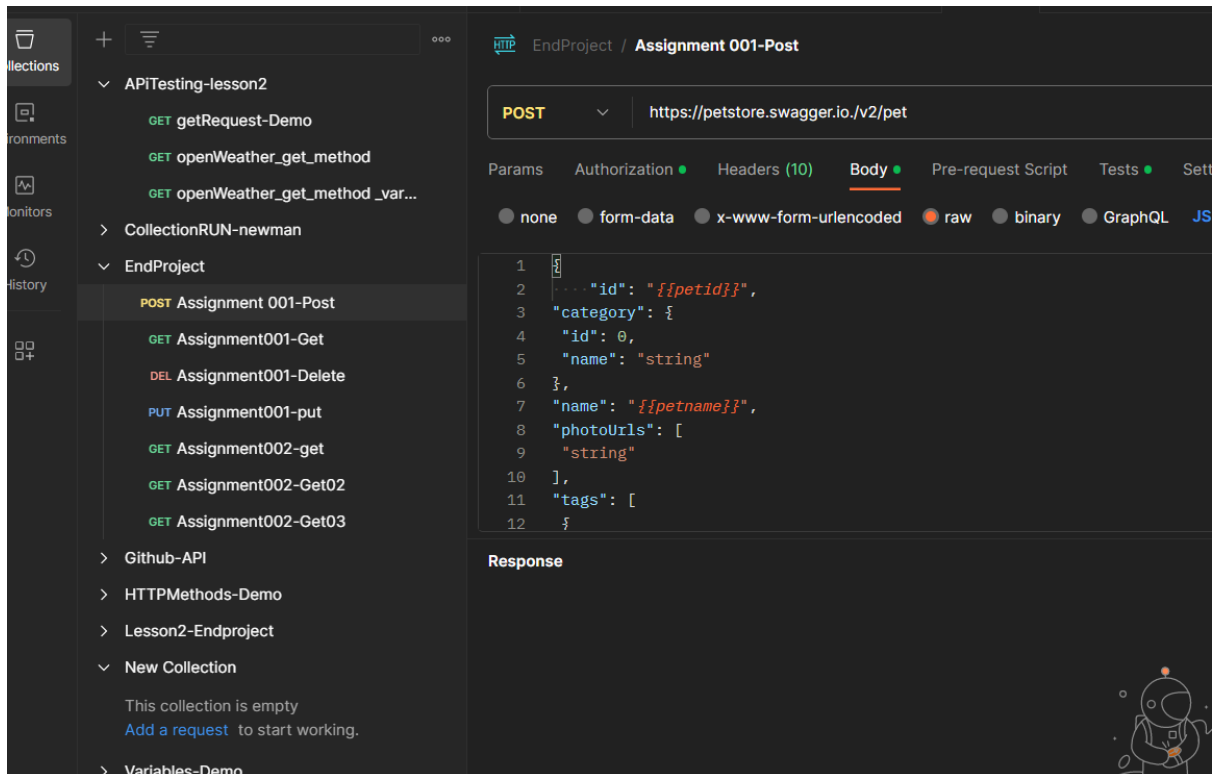
In this request part add Assignment_001-Post
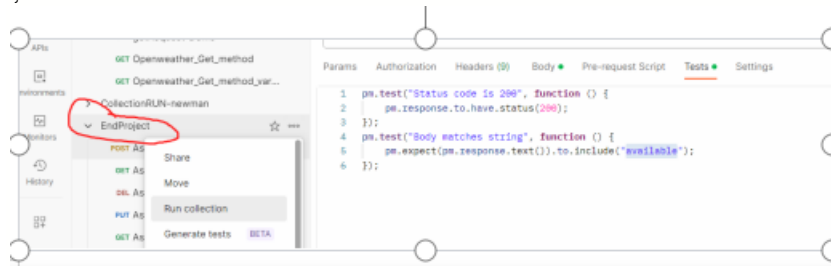
url:https://petstore.swagger.io./v2/pet

in body part select raw area and select text as json

```json
{
"id": "{{petid}}",
"category": {
 "id": 0,
 "name": "string"
},
"name": "{{petname}}",
"photoUrls": [
 "string"
],
"tags": [
 {
```
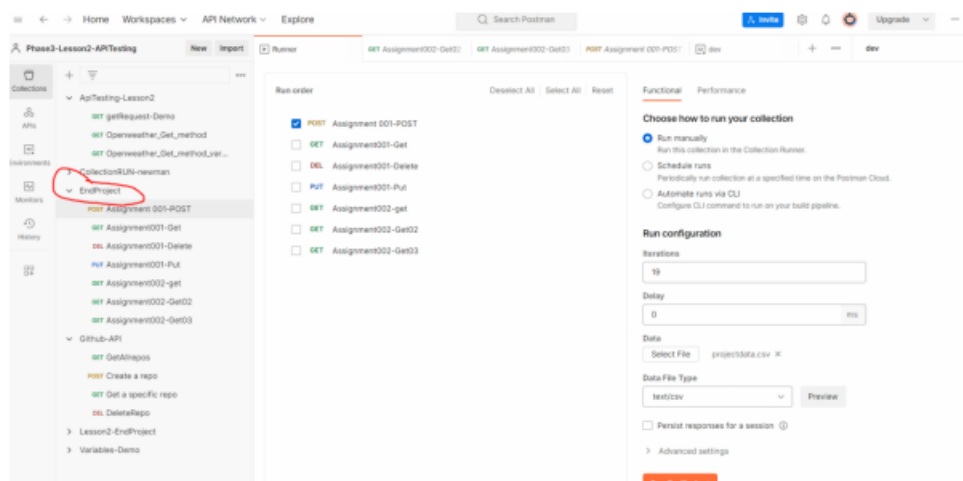
```json
    "id": 0,
    "name": "string"
   }
  ],
  "status": "available"
}
{
"id": "{{petid}}",
"category": {
 "id": 0,
 "name": "string"
},
"name": "{{petname}}",
"photoUrls": [
 "string"
],
"tags": [
 {
 "id": 0,
 "name": "string"
 }
],
"status": "available"
}
```

In this area add body part also

Go to end project click 3 dots select run collection

Next select postman-assignment001-get method

And add body value

```
{
  "id": 110,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "cow",
  "photoUrls": [
    "string"
  ],
```

```
      "tags": [

        {

            "id": 0,

            "name": "string"

        }

      ],

      "status": "available"

  }
```

This is output part
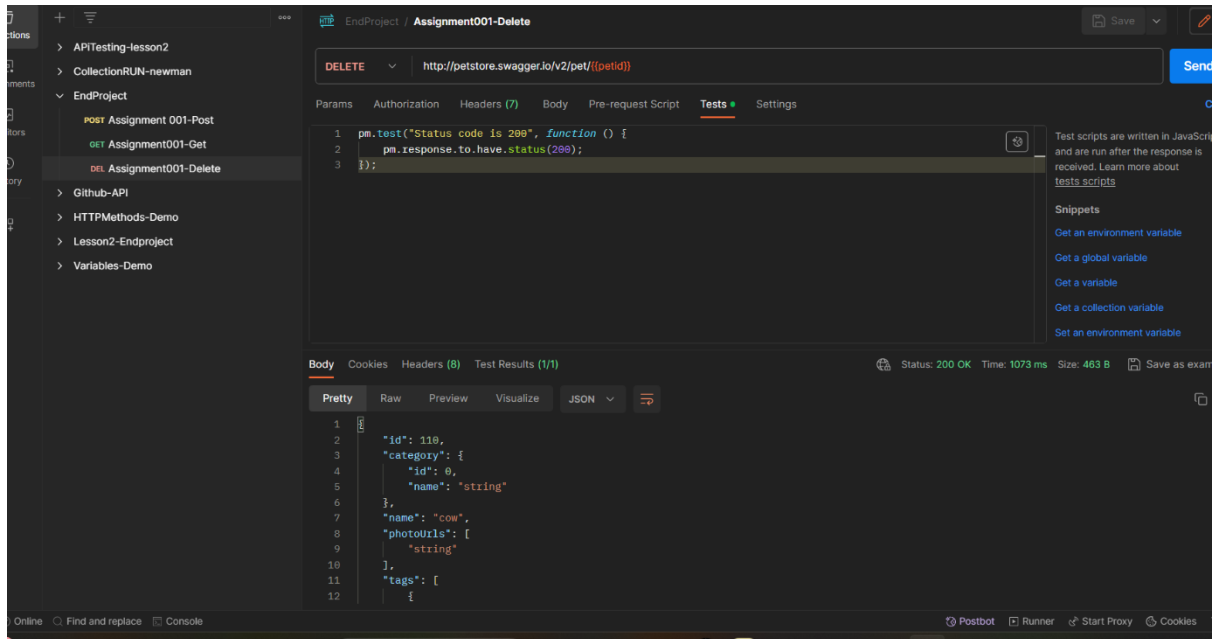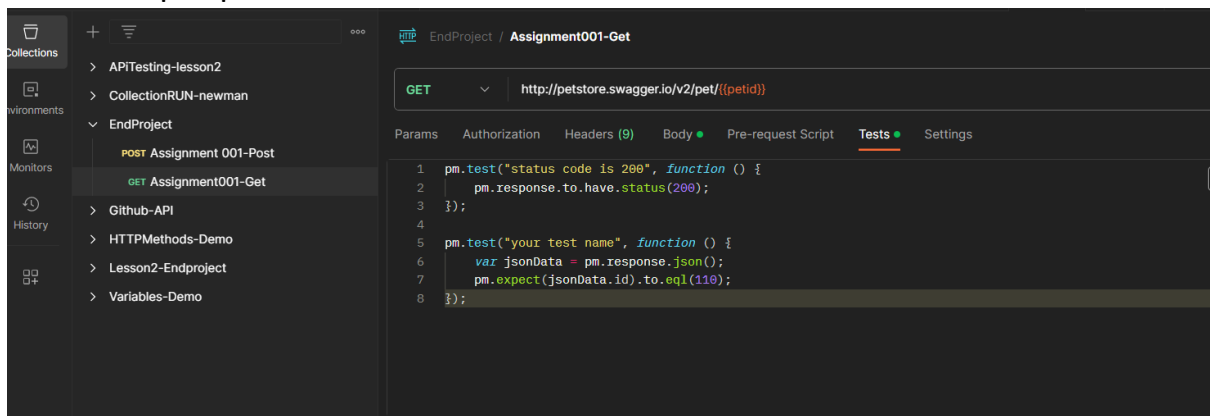
And next Take add request as delete – Assignment001-delete
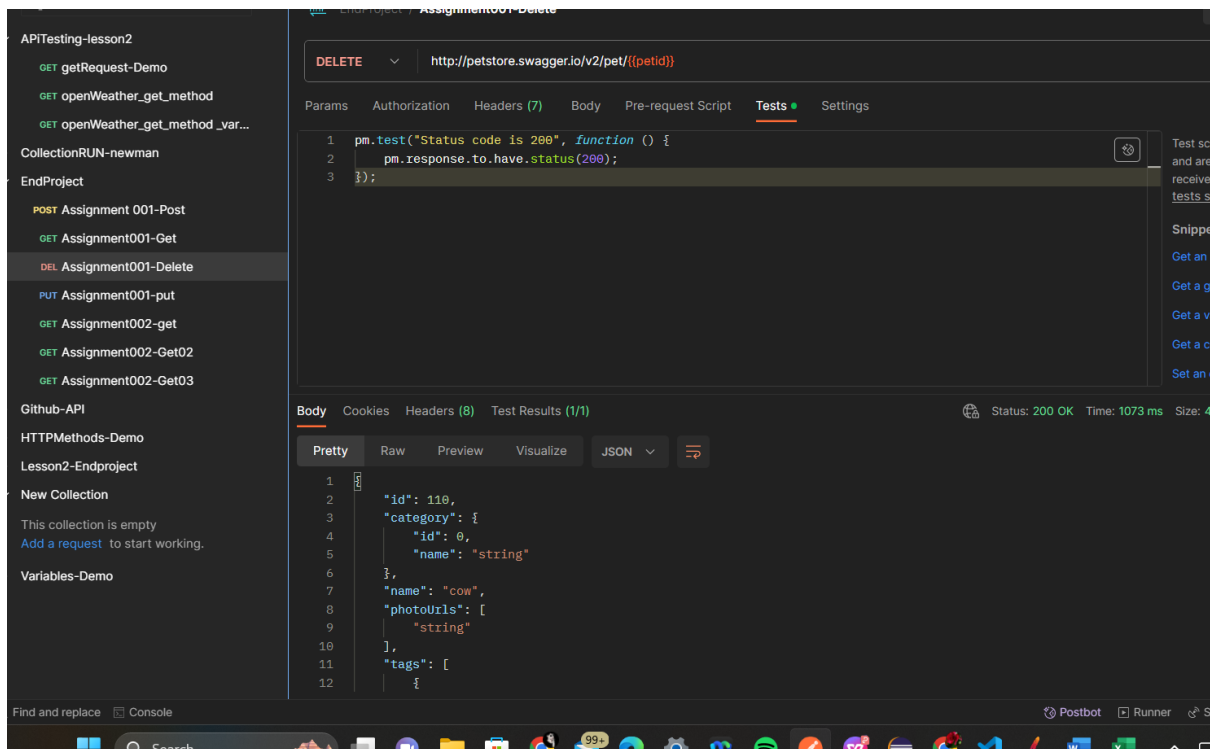
In this environment part select dev enter pet id & pet names

Next take Put assignment001-Demo
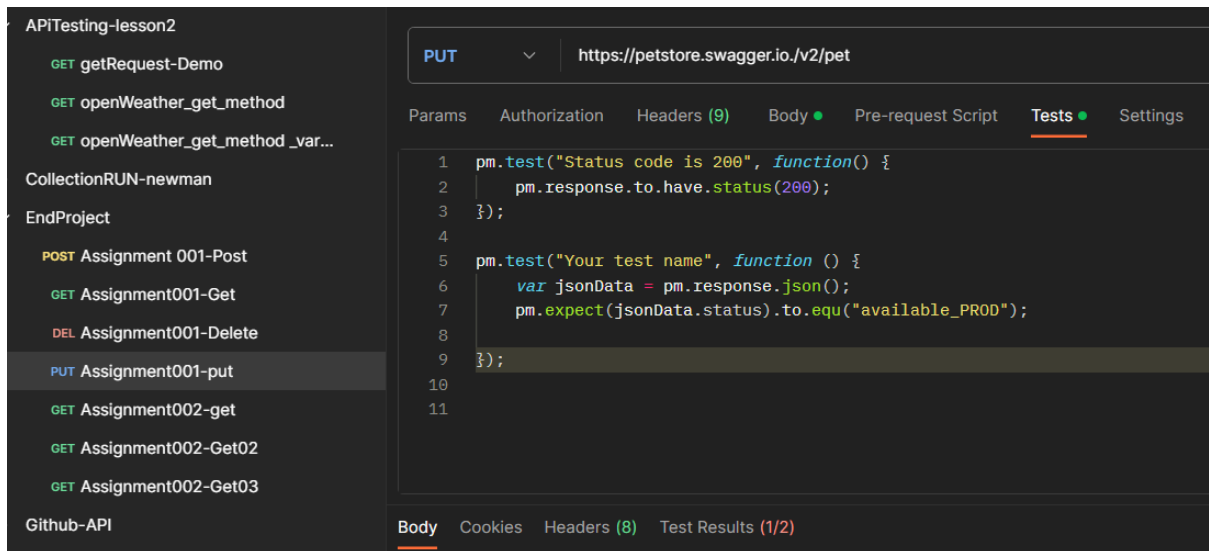
Next take put body part enter script
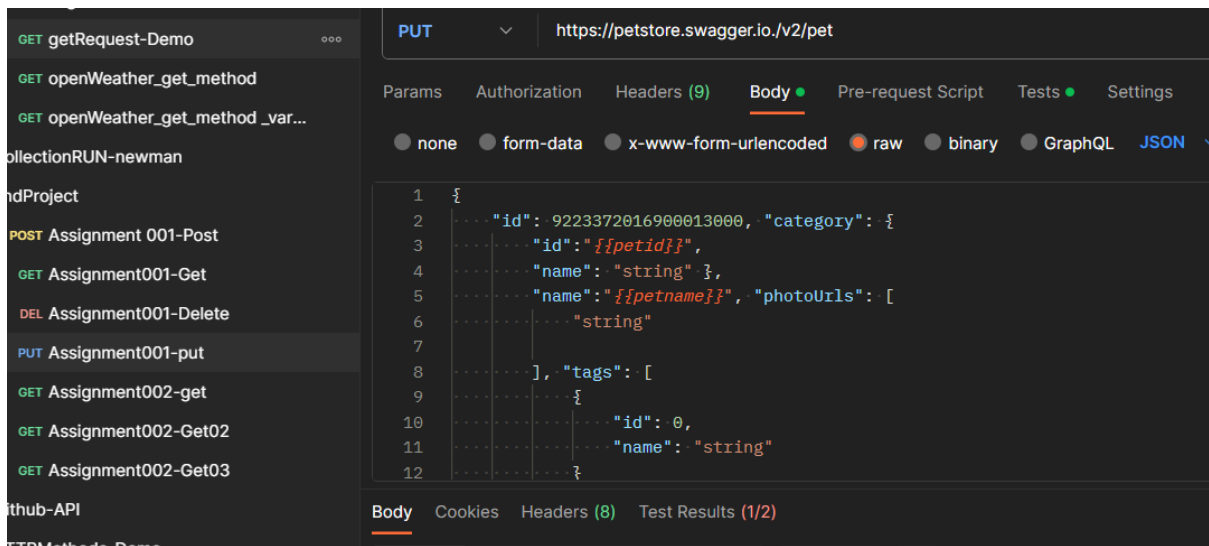
Its output part
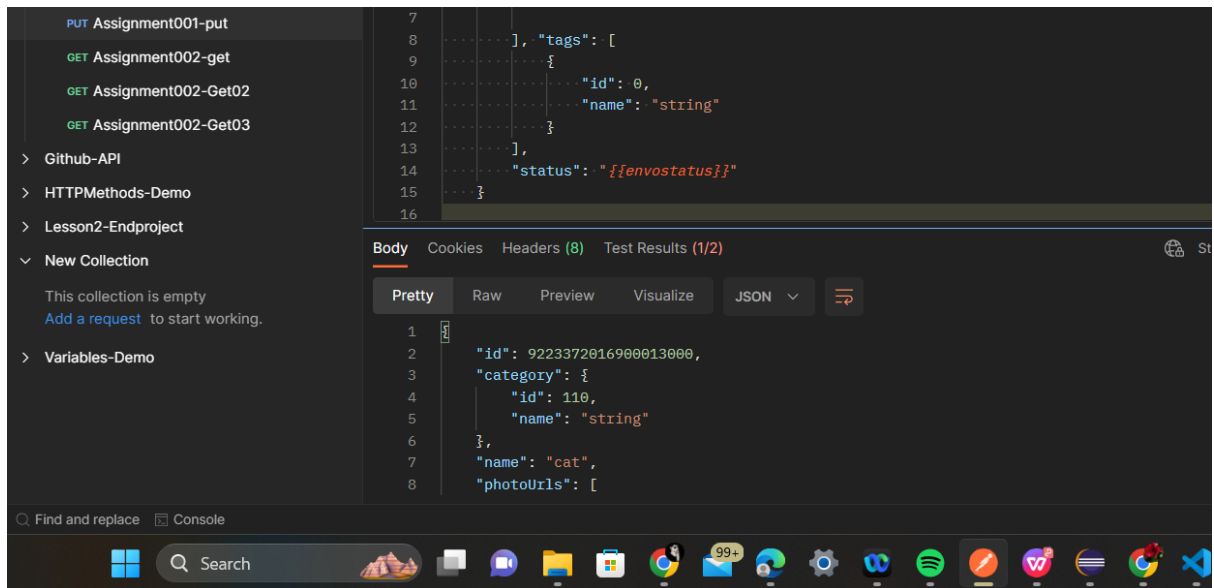
Next take get assignment002-get
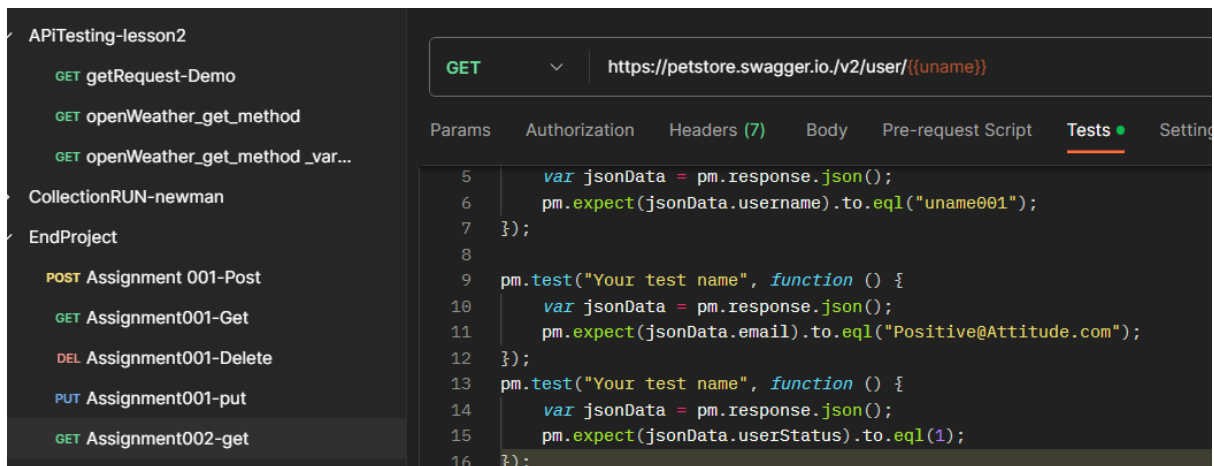
And select environment variables

And next select Get Assignment002-get02 method

APiTesting-lesson2
GET getRequest-Demo
GET openWeather_get_method
GET openWeather_get_method _var...
CollectionRUN-newman
EndProject
POST Assignment 001-Post
GET Assignment001-Get
DEL Assignment001-Delete
PUT Assignment001-put
GET Assignment002-get
GET Assignment002-Get02
GET Assignment002-Get03
Github-API
HTTPMethods-Demo
Lesson2-Endproject

GET https://petstore.swagger.io./v2/pet/findByStatus?status=available

Params •   Authorization   Headers (7)   Body   Pre-request Script   Tests   Settings

Query Params

	Key	Value
☑	status	available
	Key	Value

Body   Cookies   Headers (8)   Test Results   Status:

Pretty   Raw   Preview   Visualize   JSON ∨

Body   Cookies   Headers (8)   Test Results   Status:

Pretty   Raw   Preview   Visualize   JSON ∨

```
 4        "category": {
 5            "id": 0,
 6            "name": "string"
 7        },
 8        "name": "doggie",
 9        "photoUrls": [
10            "string"
11        ],
12        "tags": [
13            {
14                "id": 0,
15                "name": "string"
```

Next take Get Assignment002-Get 3 method
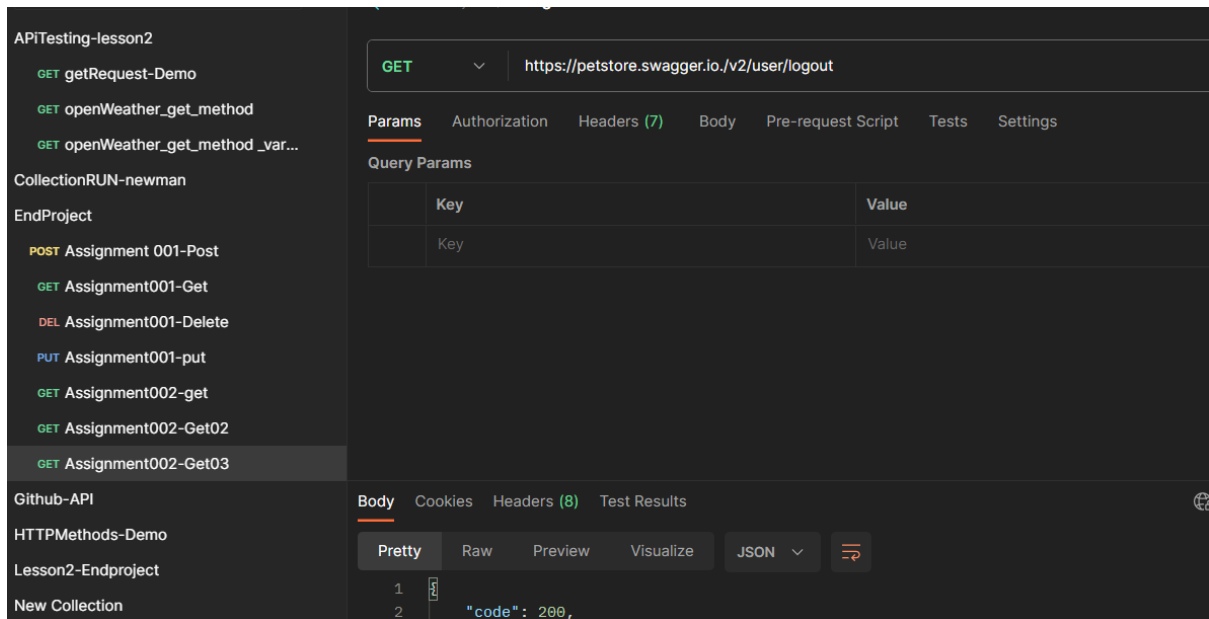
Next Take Rest Assured Assignments:

Go to eclipse part create a project resrassured scripts

Enter package name phase end project take class name Assignment001

package courseEndProject;

import java.io.File;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

import org.hamcrest.Matchers;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

import io.restassured.http.ContentType;

public class Assignment001Demo {

      Logger logger = LogManager.getLogger(Assignment001Demo.class);

      @Test(priority='1')

      public void assignment001Post()

```java
{
        logger.info("Course End project - Assignment001 - POST request");

        File file = new File("resources/data.json");

        int id = RestAssured.given()

        .baseUri("https://petstore.swagger.io/v2/pet")

        .contentType(ContentType.JSON)

        .body(file)

        .when().post()

        .then()

        .statusCode(200)

        .log().all()

        .body("name", Matchers.equalTo("duck")).extract().path("id");
logger.trace("The status code is checked");


System.out.println(id);


logger.trace("Id has been captured and validated");


}


@Test(priority='2',dependsOnMethods="assignment001Post")
public void assignment001Get()
{

        logger.info("Course End project - Assignment001 - GET request");
int id = RestAssured.given()
        .baseUri("https://petstore.swagger.io/v2/pet/987")
        .when()
        .get()
        .then()
        .statusCode(200)
```

```
                    .log().all()

                    .body("status", Matchers.equalTo("available")).extract().path("category.id");

        System.out.println(id); // 0

        logger.trace("Id and status has been captured and validated");

            }


            @Test(priority='3',dependsOnMethods="assignment001Get")

            public void assignment001Delete()

            {

                    logger.info("Course End project - Assignment001 - Delete request");

RestAssured.given()

                    .baseUri("https://petstore.swagger.io/v2/pet/987")

                    .when()

                    .delete()

                    .then()

                    .statusCode(200)

                    .log().all()

                    .body("code", Matchers.equalTo(200))

                    .body("type", Matchers.equalTo("unknown"))

                    .body("message", Matchers.equalTo("987"));


            }
```

Output part:

```
[RemoteTestNG] detected TestNG version 7.7.1
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
[INFO ] 2023-11-10 20:36:35.517 [main] Assignment001Demo - Course End project - Assignment001 - POST request
HTTP/1.1 200 OK
Date: Fri, 10 Nov 2023 15:06:41 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": 987,
    "category": {
        "id": 0,
        "name": "string"
    },
    "name": "duck",
    "photoUrls": [
        "string"
    ],
    "tags": [
        {
            "id": 0,
            "name": "string"
        }
    ],
```

Next step is Assignment002 post.java

package courseEndProject;

import java.io.File;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

import org.hamcrest.Matchers;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

import io.restassured.http.ContentType;

public class Assignment002Post {

        Logger logger = LogManager.getLogger(Assignment002Post.class);

        @Test(priority='1')

        public void assignment002Post()

        {

                logger.info("Course End project - Assignment002 - POST request");

```java
        File file = new File("resources/data.json");

        int id = RestAssured.given()

        .baseUri("https://petstore.swagger.io/v2/pet")

        .contentType(ContentType.JSON)

        .body(file)

        .when().post()

        .then()

        .statusCode(200)

        .log().all()

        .body("name", Matchers.equalTo("duck")).extract().path("id");
logger.trace("The status code is checked");


System.out.println(id);


logger.trace("Id has been captured and validated");


}



@Test(priority='2',dependsOnMethods = "assignment002Post")
public void assignment002Put()
{
        File file = new File("resources/dataput.json");
int id = RestAssured.given()
        .baseUri("https://petstore.swagger.io/v2/pet/")

        .contentType(ContentType.JSON)

        .body(file)

        .when()

        .post()

        .then()

        .statusCode(200)
```

```
.log().all()

.body("status", Matchers.equalTo("available_QA")).extract().path("id");


System.out.println(id);



    }
}
```

Output:

        Next Assignment003 Demo.java

package courseEndProject;

import org.hamcrest.Matchers;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class Assignment003Demo {

```java
@Test(priority='1')

public void assignment003User()

{

        RestAssured.given()

        .baseUri("https://petstore.swagger.io/v2/user/Uname001")

        .when()

        .get()

        .then()

        .statusCode(200)

        .log().all()

        .body("username", Matchers.equalTo("Uname001"))

        .body("email", Matchers.equalTo("Positive@Attitude.com"))

        .body("userStatus", Matchers.equalTo(1))

        ;




}

@Test(priority='2')

public void assignment004login()

{

        RestAssured.given()

        .baseUri("https://petstore.swagger.io/v2/user/login")

        .auth().preemptive().basic("Uname001", "@tt!tude")

        .when()

        .get()

        .then()

        .statusCode(200)

        .log().all()

        .body("message", Matchers.anything());
```

Next Assignment004

public void assignment004login()

```java
{
        RestAssured.given()

        .baseUri("https://petstore.swagger.io/v2/user/login")

        .auth().preemptive().basic("Uname001", "@tt!tude")

        .when()

        .get()

        .then()

        .statusCode(200)

        .log().all()

        .body("message", Matchers.anything());
```

Next take Assignment 005 Demo

package courseEndProject;

import org.hamcrest.Matchers;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class Assignment005Demo {

```java
        @Test(priority='1')

        public void assignment005FindByStatus()

        {
                RestAssured.given()

                .baseUri("https://petstore.swagger.io/v2/pet/findByStatus")

                //.queryParam("status", "available")
```

```java
            //.queryParam("status", "pending")

            .queryParam("status", "sold")

            .when()

            .get()

            .then()

            .statusCode(200)

            .log().all()


            ;




    }




    @Test(priority='2')

    public void assignment006Logout()

    {

            RestAssured.given()

            .baseUri("https://petstore.swagger.io/v2/user/logout")

            .when()

            .get()

            .then()

            .statusCode(200)

            .log().all()

            .body("code", Matchers.equalTo(200))

            .body("type", Matchers.equalTo("unknown"))

            .body("message", Matchers.equalTo("ok"))

            ;
```

}

        }

TTP/1.1 200 OK
ate: Fri, 10 Nov 2023 14:01:33 GMT
ontent-Type: application/json
ransfer-Encoding: chunked
onnection: keep-alive
ccess-Control-Allow-Origin: *
ccess-Control-Allow-Methods: GET, POST, DELETE, PUT
ccess-Control-Allow-Headers: Content-Type, api_key, Authorization
erver: Jetty(9.2.9.v20150224)

    {
        "id": 381,
        "category": {
            "id": 0,
            "name": "Cat"
        },
        "name": "Hilma",
        "photoUrls": [
            "string"
        ],
        "tags": [
            {
                "id": 0,
                "name": "string"
            }
        ],
        "status": "sold"
    }

TTP/1.1 200 OK
ate: Fri, 10 Nov 2023 14:01:35 GMT
ontent-Type: application/json
ransfer-Encoding: chunked
onnection: keep-alive
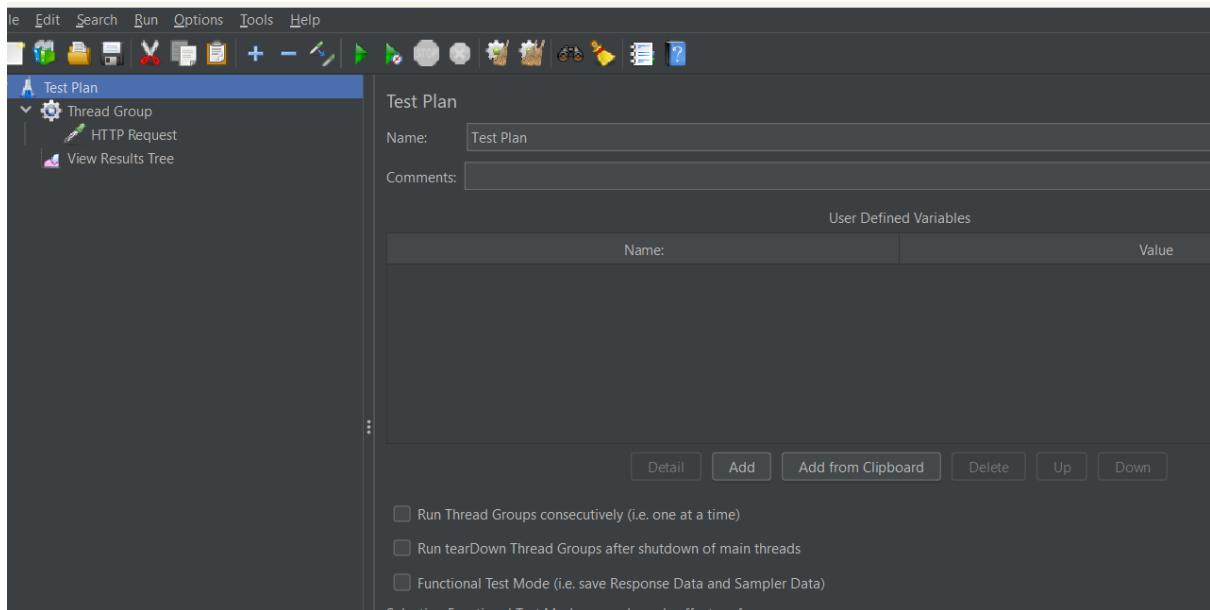
28°C
Mostly cloudy   Q Search

J meter part :
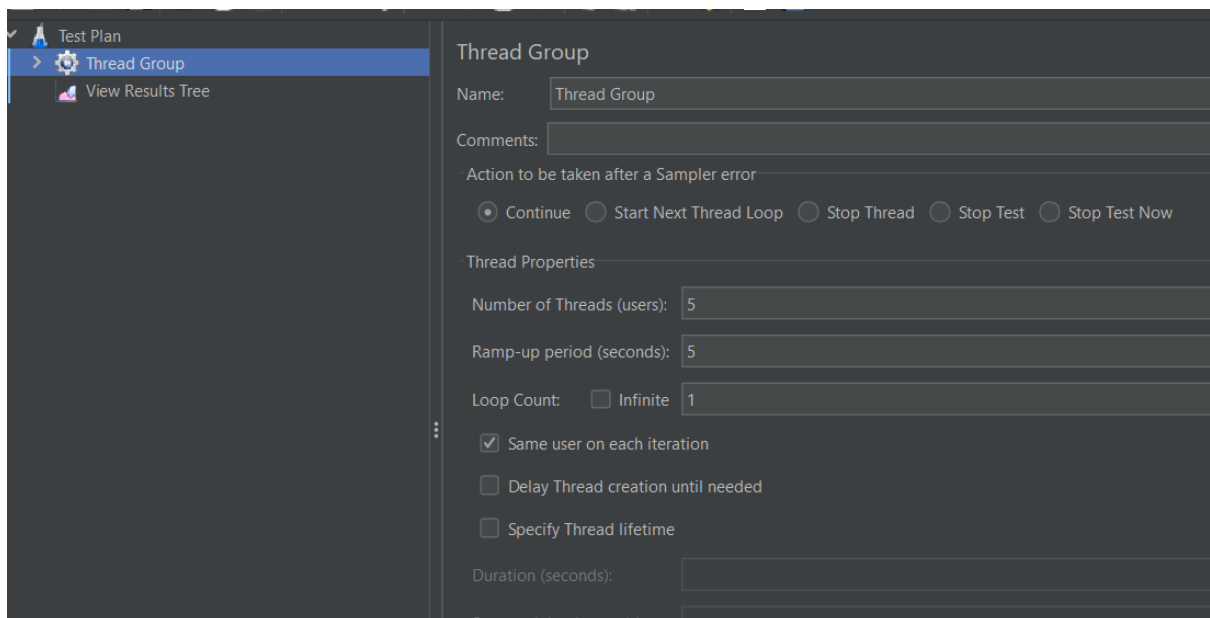
J meter part take simplilearn website

First go to j meter add test plan

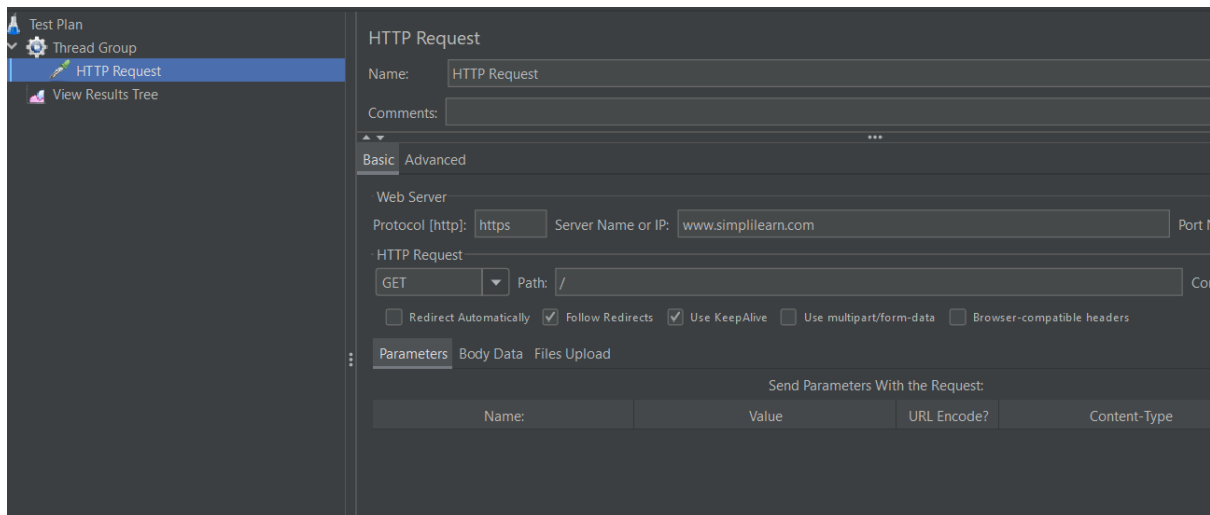In test plan add thread groups(user thread)

Add sampler request HTTP request

Add thread group

Create http request

View results tree