

REST Assured Assignment: Step 1

Create the json file

```
{
  "id": 947,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "goat",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Step 2

Create the post method(**FOR ASSIGNMENT 1**)

```
package phaseEndproject;
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;
import testHTTPmethods.Log4J2Demo;

public class Assignment001Post {
```

```

        Logger logger =
LogManager.getLogger(Assignment001Post.class);

        @Test(priority='1')

        public void assignment001post()
        {
            logger.info(" pahse End project -
Assignment001- Post Request");

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

            int id = RestAssured.given()

.baseUrl("https://petstore.swagger.io/v2/pet")
            .contentType(ContentType.JSON)
            .body(file)
            .when().post()
            .then().statusCode(200)
            .log().all()
            .body("name",
Matchers.equalTo("goat")).extract().path("id");

            logger.debug(" The status code is checked");

            System.out.println(id);
            logger.info("Id has been captured and
printed.");
        }

        @Test(priority='2', dependsOnMethods=
{"assignment001post"})

        public void assignment001Get()
        {
            String url =
"https://petstore.swagger.io/v2/pet/947";

```

```

        RestAssured.given()
            .baseUrl(url)
            .when().get()
            .then().statusCode(200)
            .log().all()
            .body("status",
Matchers.equalTo("available"))
            .body("id", Matchers.equalTo(947));

    }

    @Test(priority='3', dependsOnMethods=
{"assignment001post"})

    public void assignment001Delete()
    {
        String url =
"https://petstore.swagger.io/v2/pet/947";
        RestAssured.given()
            .baseUrl(url)
            .when().delete()
            .then().statusCode(200)
            .log().all()
            .body("code", Matchers.equalTo(200))
            .body("message", Matchers.equalTo("947"));

    }

}

```

Assignment 2

```

package phaseEndproject;
import java.io.File;

```

```

import org.hamcrest.Matchers;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
import io.restassured.http.ContentType;

public class Assignment002 {

    @Test

    public void post_status_available()
    {
        File file = new File("resources/data.json");

        int id = RestAssured.given()

.baseUrl("https://petstore.swagger.io/v2/pet")
        .contentType(ContentType.JSON)
        .body(file)
        .when().post()
        .then().statusCode(200)
        .body("name",
Matchers.equalTo("goat")).extract().path("id");

        System.out.println(id);
    }

    @Test(priority='2')

    public void put_status_available_Dev()
    {
        File file = new
File("resources/data-dev.json");

        int id = RestAssured.given()

.baseUrl("https://petstore.swagger.io/v2/pet")
        .contentType(ContentType.JSON)

```

```

        .body(file)
        .when().put()
        .then().statusCode(200)
        .body("status",
Matchers.equalTo("available-Dev")).extract().path("id");

        System.out.println(id);
    }

    @Test(priority='3')

    public void put_status_available_QA()
    {
        File file = new
File("resources/data-qa.json");

        int id = RestAssured.given()

        .baseUrl("https://petstore.swagger.io/v2/pet")
        .contentType(ContentType.JSON)
        .body(file)
        .when().put()
        .then().statusCode(200)
        .body("status",
Matchers.equalTo("available-QA")).extract().path("id
");

        System.out.println(id);
    }

    @Test(priority='4')

    public void put_status_available_Prod()
    {
        File file = new
File("resources/data-prod.json");

```

```

        int id = RestAssured.given()

        .baseUrl("https://petstore.swagger.io/v2/pet")
          .contentType(ContentType.JSON)
          .body(file)
          .when().put()
          .then().statusCode(200)
          .body("status",
Matchers.equalTo("available-Prod")).extract().path("
id");

        System.out.println(id);
    }
}

```

Assignment 3&4&5&6

```

package phaseEndproject;

import org.hamcrest.Matchers;
import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class Assignement3456 {

    // @Test
    public void assignment003()
    {

        String url =
"https://petstore.swagger.io/v2/user/user123";
        RestAssured.given()
          .baseUrl(url)
          .when().get()
          .then().statusCode(200)

```

```

        .log().all()
        .body("username",
Matchers.equalTo("user123"))
        .body("email",
Matchers.equalTo("user@gmail.com"))
        .body("userStatus",Matchers.equalTo(0));

    }

```

```

//@Test
public void assignment004()
{

    String url =
"https://petstore.swagger.io/v2/user/login";
    RestAssured.given()
        .baseUrl(url)
        .auth().preemptive().basic("user123",
"user@123")
        .when().get()
        .then().statusCode(200)
        .log().all()
        .body("message", Matchers.anything());

}

```

```

//@Test
public void assignment005_available()
{

    String url =
"https://petstore.swagger.io/v2/pet/findByStatus";
    RestAssured.given()
        .baseUrl(url)
        .queryParams("status", "available")

```

```

        .when().get()
        .then().statusCode(200)
        .log().all();

    }

    //@Test
    public void assignment005_pending()
    {
        String url =
"https://petstore.swagger.io/v2/pet/findByStatus";
        RestAssured.given()
            .baseUrl(url)
            .queryParams("status", "pending")
            .when().get()
            .then().statusCode(200)
            .log().all();

    }

    //@Test
    public void assignment005_sold()
    {
        String url =
"https://petstore.swagger.io/v2/pet/findByStatus";
        RestAssured.given()
            .baseUrl(url)
            .queryParams("status", "sold")
            .when().get()
            .then().statusCode(200)
            .log().all();

    }

```



```

@Test
public void assignment006()
{
    String url =
"https://petstore.swagger.io/v2/user/logout";
    RestAssured.given()
        .baseUrl(url)
        .when().get()
        .then().statusCode(200)
        .log().all();

}

}

```

Jmeter assignment

Step 1

Use blaze meter application to record the actions in the website .

After recording the action from the website ,save the file as jmeter file

Step 2

Open the jmeter application

Create the name for the test as lesson end project

After that upload the jmeter file as previously recorded jmeter file

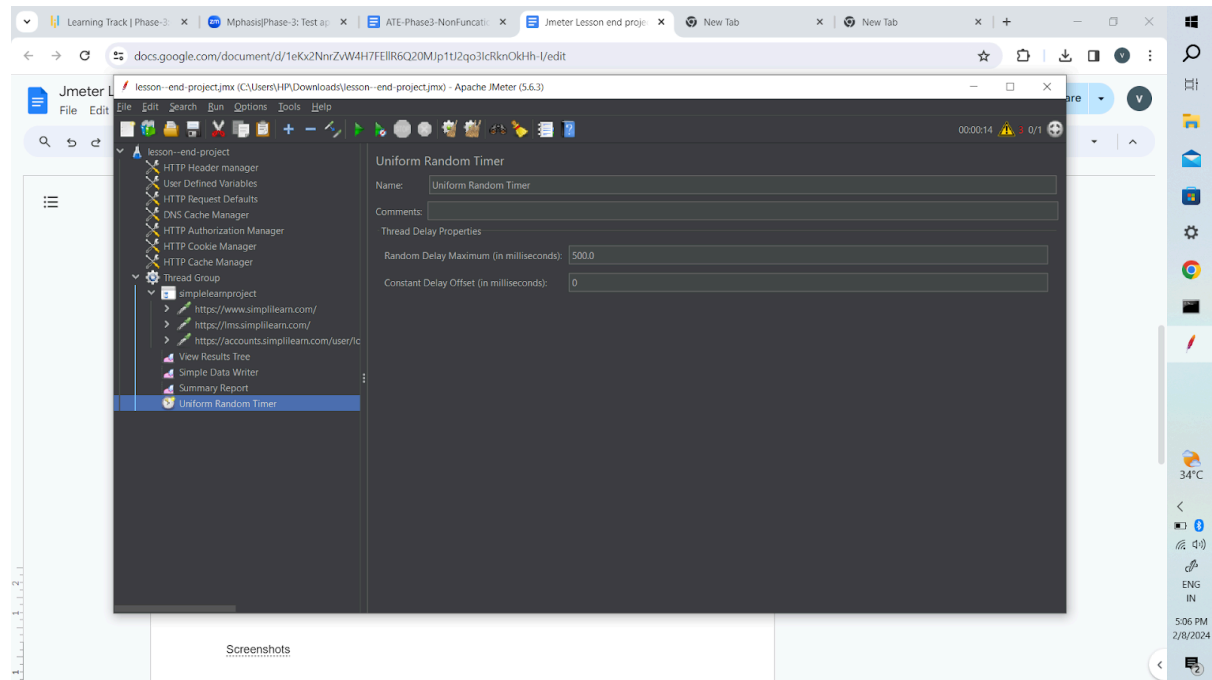
Step 3

Create the **view result tree** using thread group → add → listeners → **view result tree**

Create the **simple data writer** using thread group → add → listeners → **simple data writer**

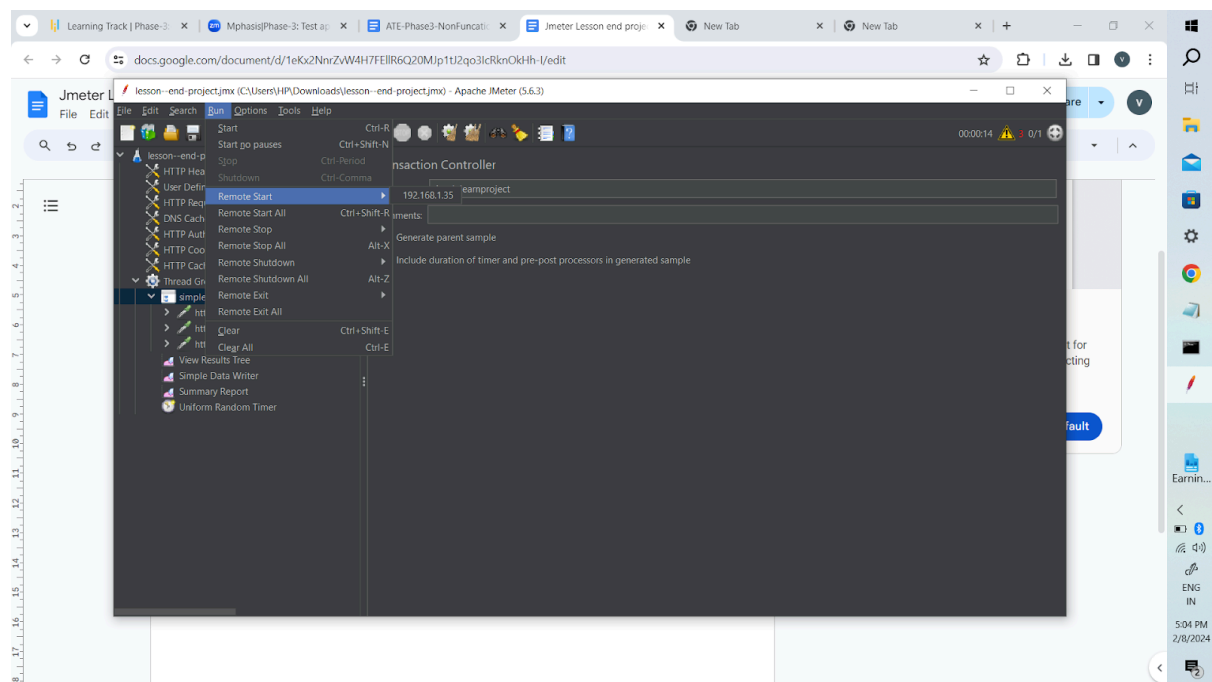
Create the **summary report** using thread group → add → listeners → **summary report**

Create the **uniform random timer** using thread group → add → timer → **uniform random timer** (Random delay seconds as 500 milliseconds)

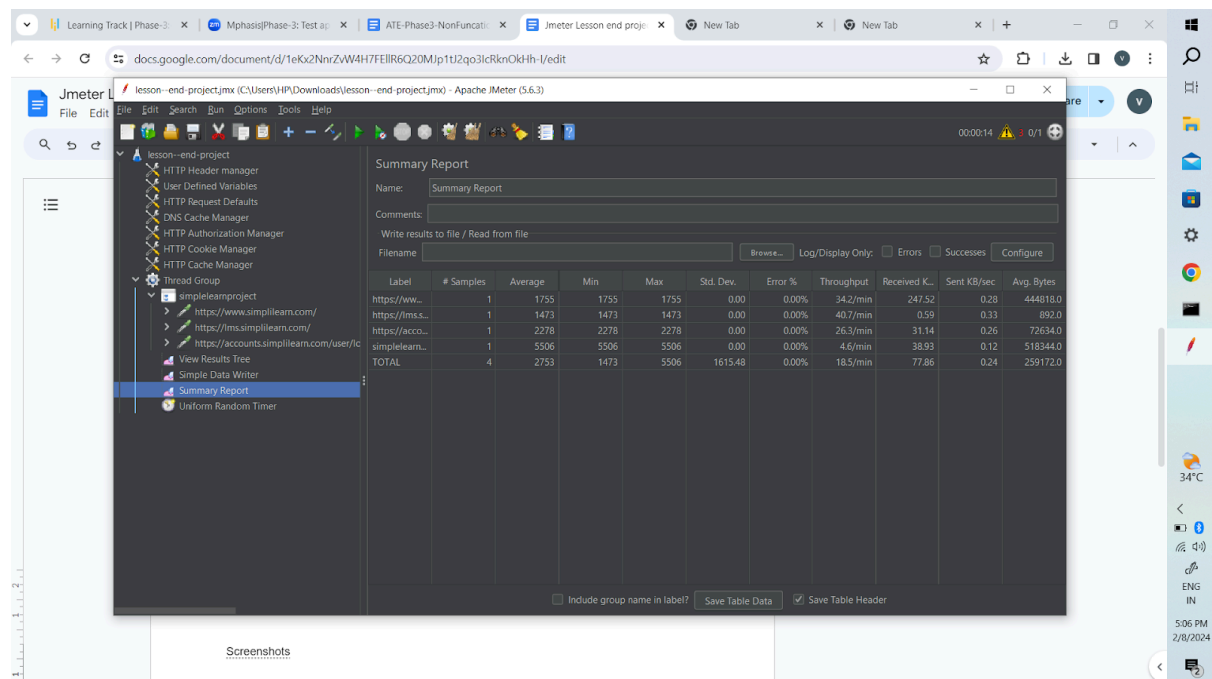
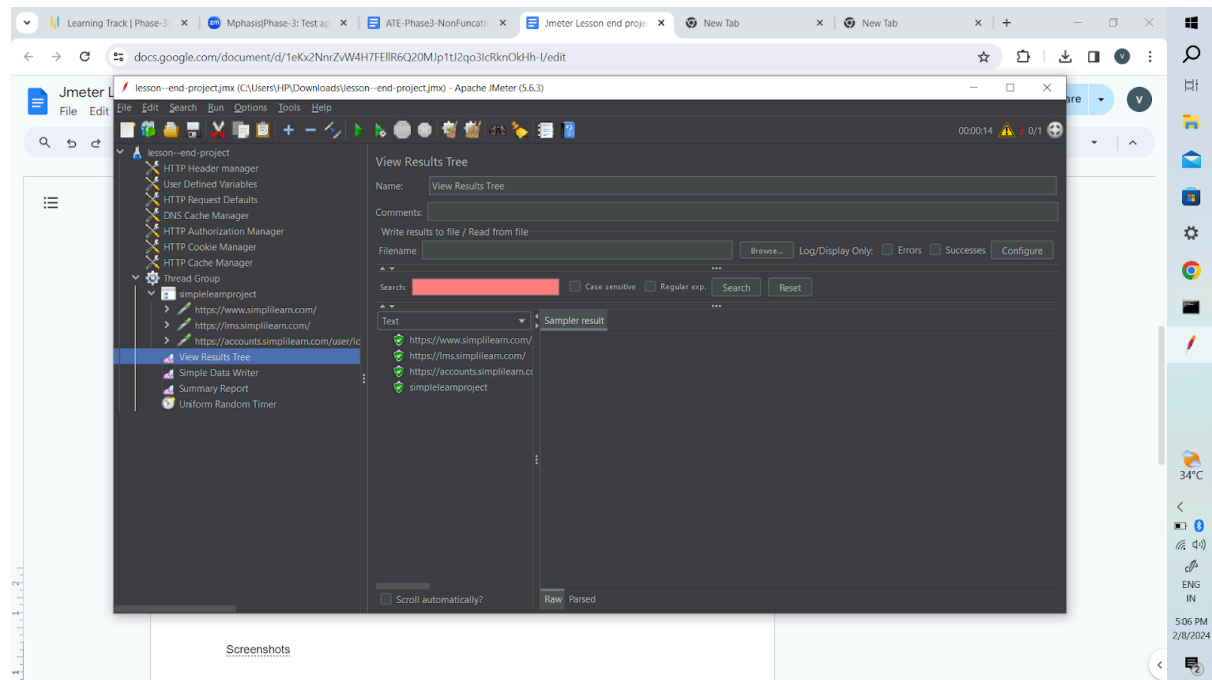


Step 4

Run as remote server(192.168.1.35) to run below jmeter



Screenshots

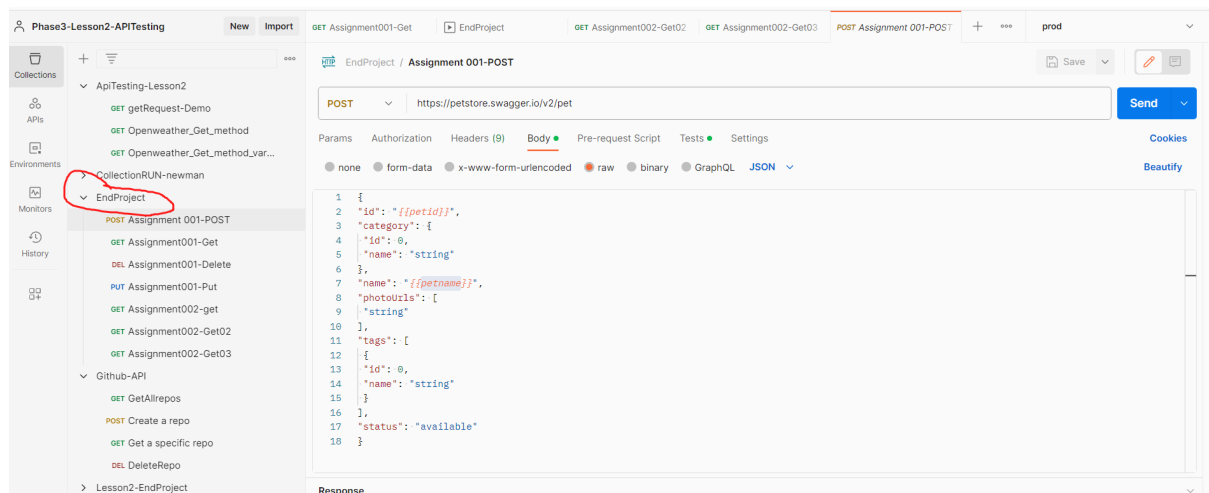


Postman Assignment_001:

Create the excel file with data and save it on your local machine

	A	B	C
1	petid	petname	
2	110	Cat	
3	120	Dog	
4	130	Tiger	
5	140	bird	
6	150	Lion	
7	160	Donkey	
8	170	Cow	
9	180	Doggie	
10	190	Jaguar	
11	200	hen	
12	210	goat	
13	220	horse	
14	230	bear	
15	240	sheep	
16	250	rat	
17	260	mice	
18	270	fish	
19	280	ant	
20	290	bee	

Open the post man use the POST



Use the below code

```

{
  "id": "{{petid}}",
  "category": {
    "id": 0,
    "name": "string"
  }
}

```

```

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

```

After that run as collection

EndProject - Run results

Run Again Automate Run + New Run Export Results

Ran today at 12:37:38 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	dev	19	8s 854ms	38	340 ms

All Tests Passed (38) Failed (0) Skipped (0)

View Summary

PASS Status code is 200

PASS Body matches string

Iteration 12

POST Assignment 001-POST

https://petstore.swagger.io/v2/pet

200 OK 281 ms 465 B

PASS Status code is 200

PASS Body matches string

Iteration 13

POST Assignment 001-POST

https://petstore.swagger.io/v2/pet

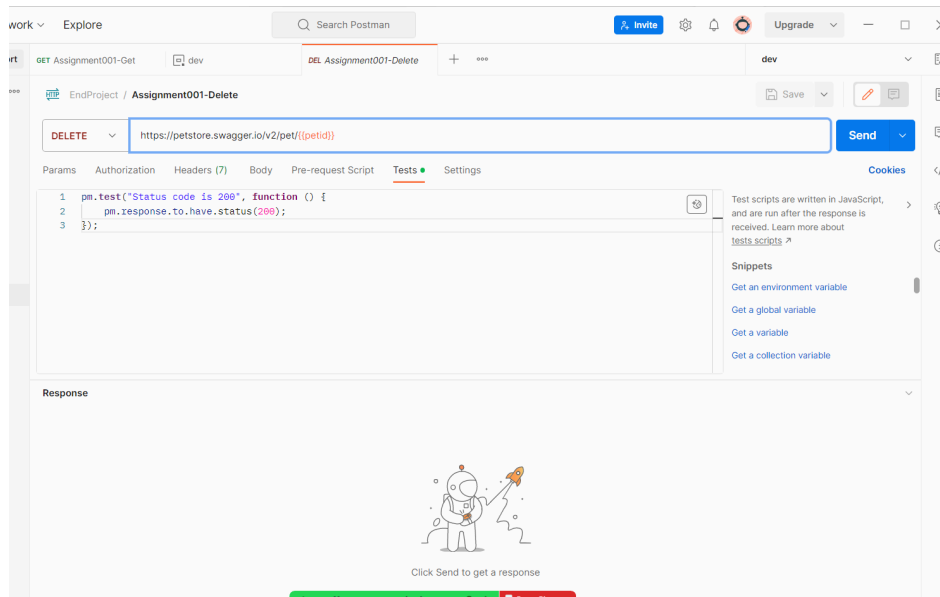
200 OK 307 ms 464 B

PASS Status code is 200

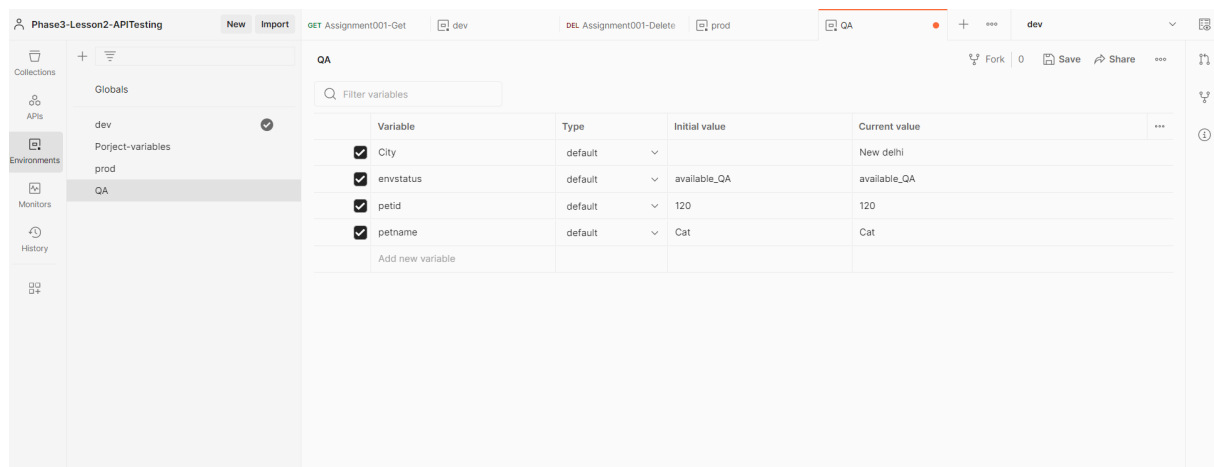
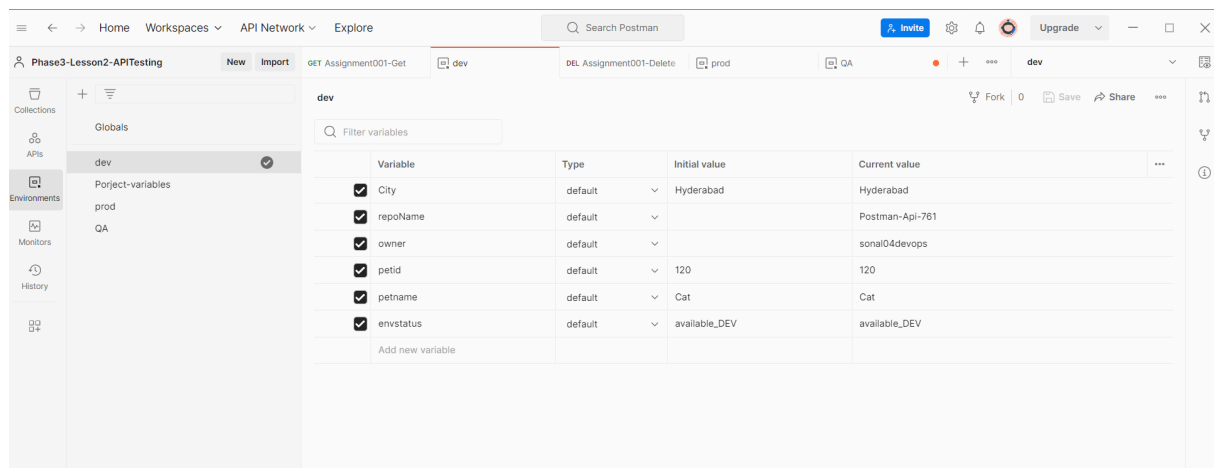
PASS Body matches string

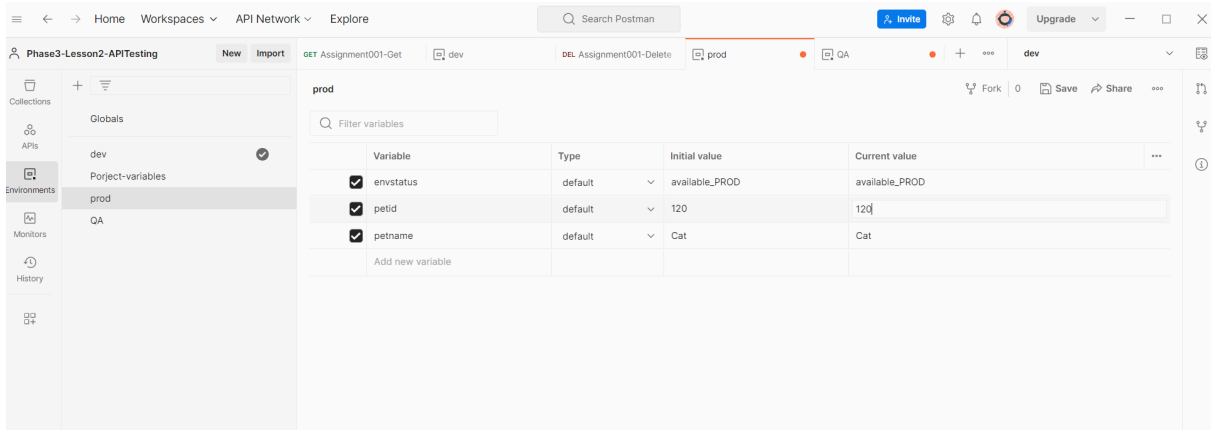
Iteration 14

Request 3

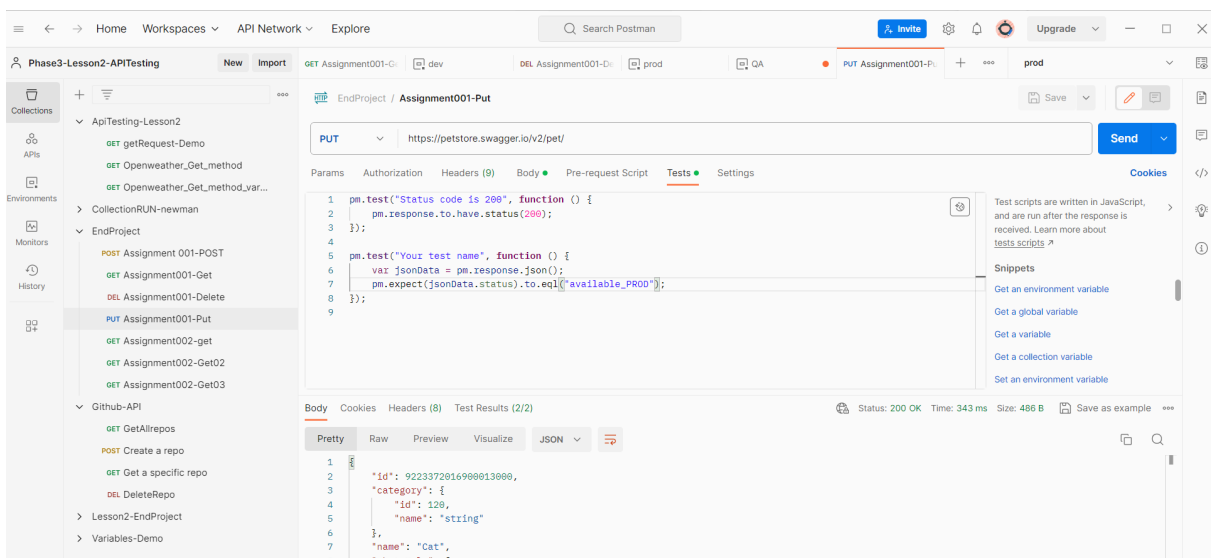
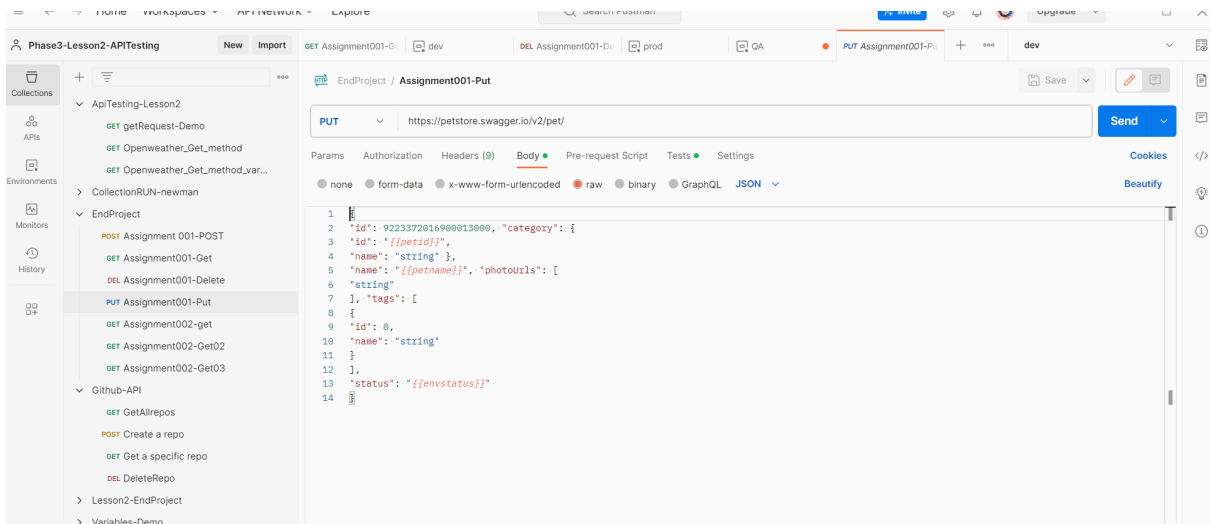


Request 4:-After that Create 3 env variables

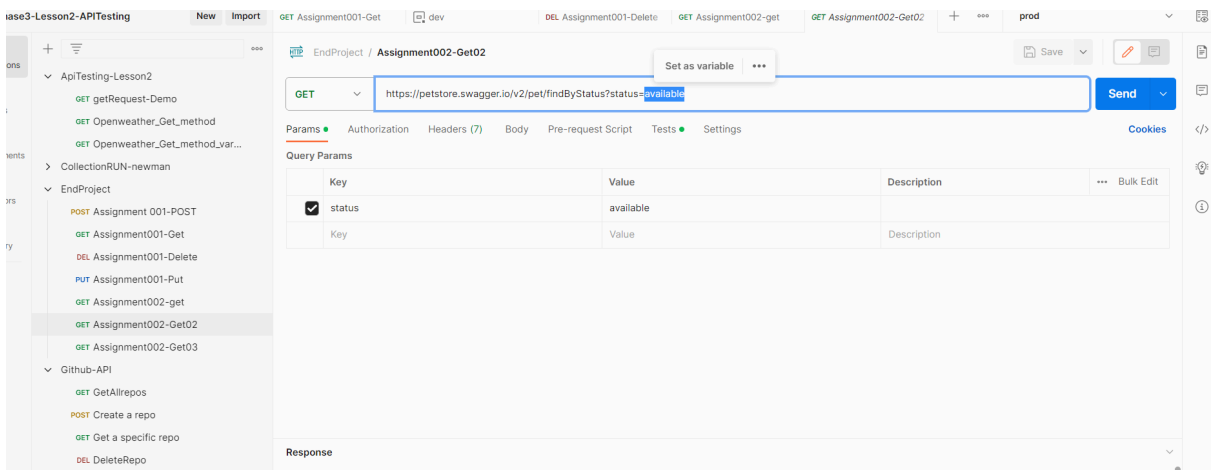
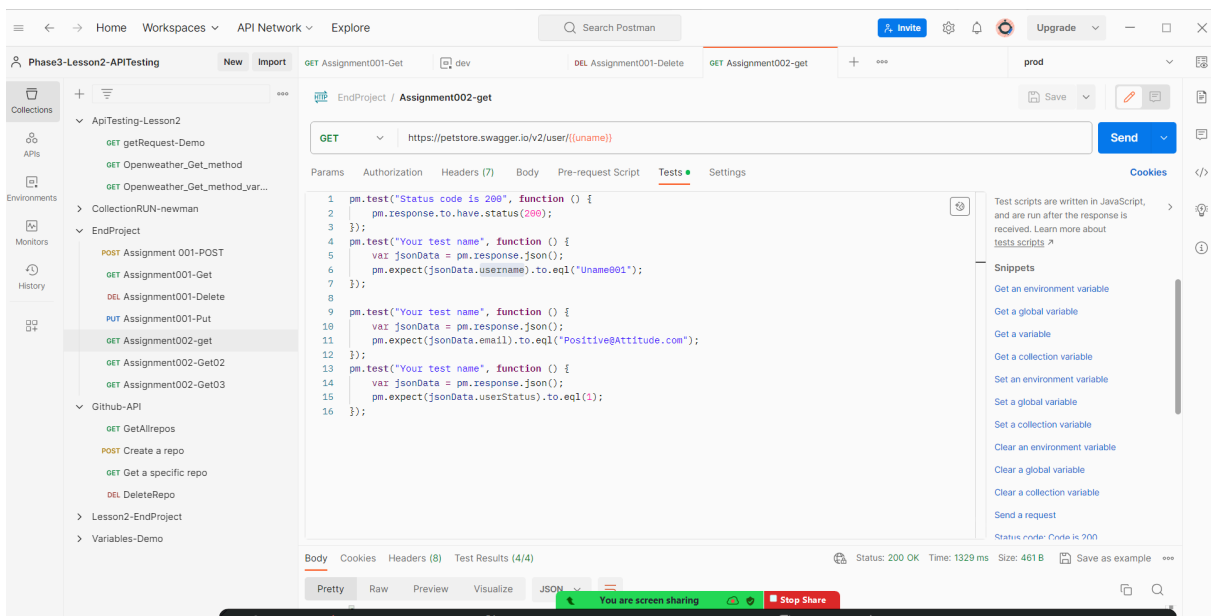
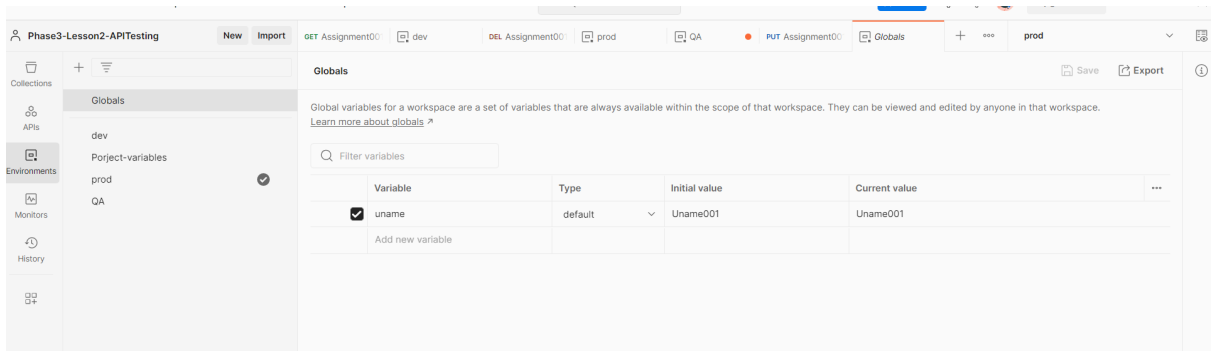




PUT request



Request 5 (GET request)



← → Home Workspaces API Network Explore

Search Postman

Invite Upgrade

Phase3-Lesson2-APITesting

New Import

GET Assignment001-Get dev DEL Assignment001-Delete GET Assignment002-get GET Assignment002-Get03 + prod

collections

APIs

environments

Monitors

History

ApiTesting-Lesson2

getRequest-Demo

Openweather_Get_method

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

GET GetAllrepos

POST Create a repo

GET Get a specific repo

DEL DeleteRepo

Lesson2-EndProject

Variables-Demo

EndProject / Assignment002-Get03

GET https://petstore.swagger.io/v2/user/logout Send

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

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 1256 ms Size: 370 B Save as example

Pretty Raw Preview Visualize JSON

```
1
2 {"code": 200,
3   "message": "unknown"}
```