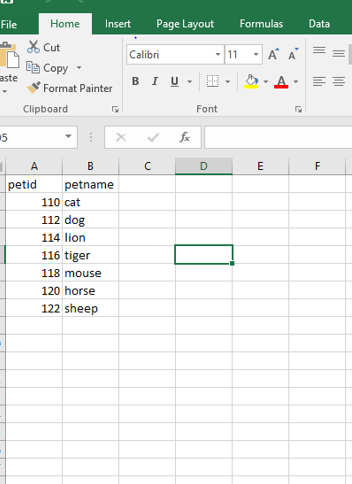
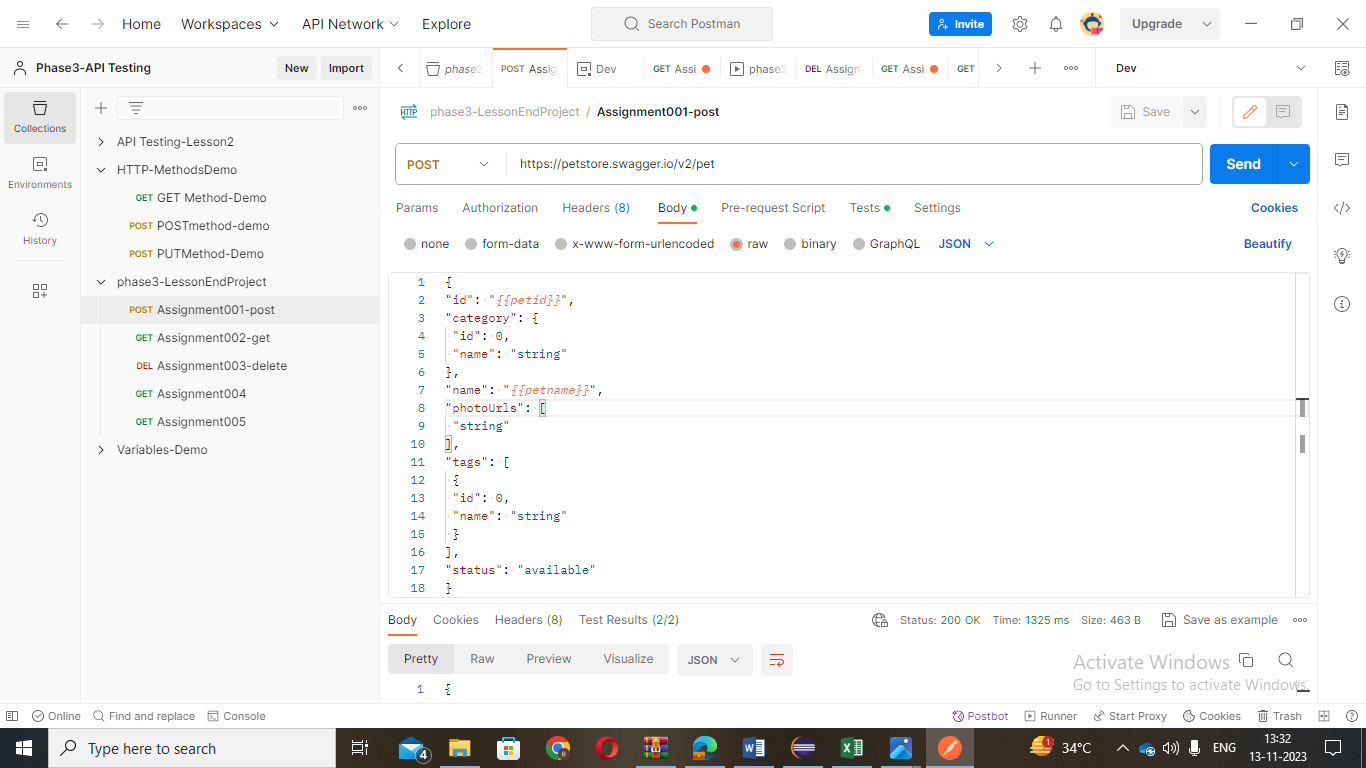
**Phase End Project**

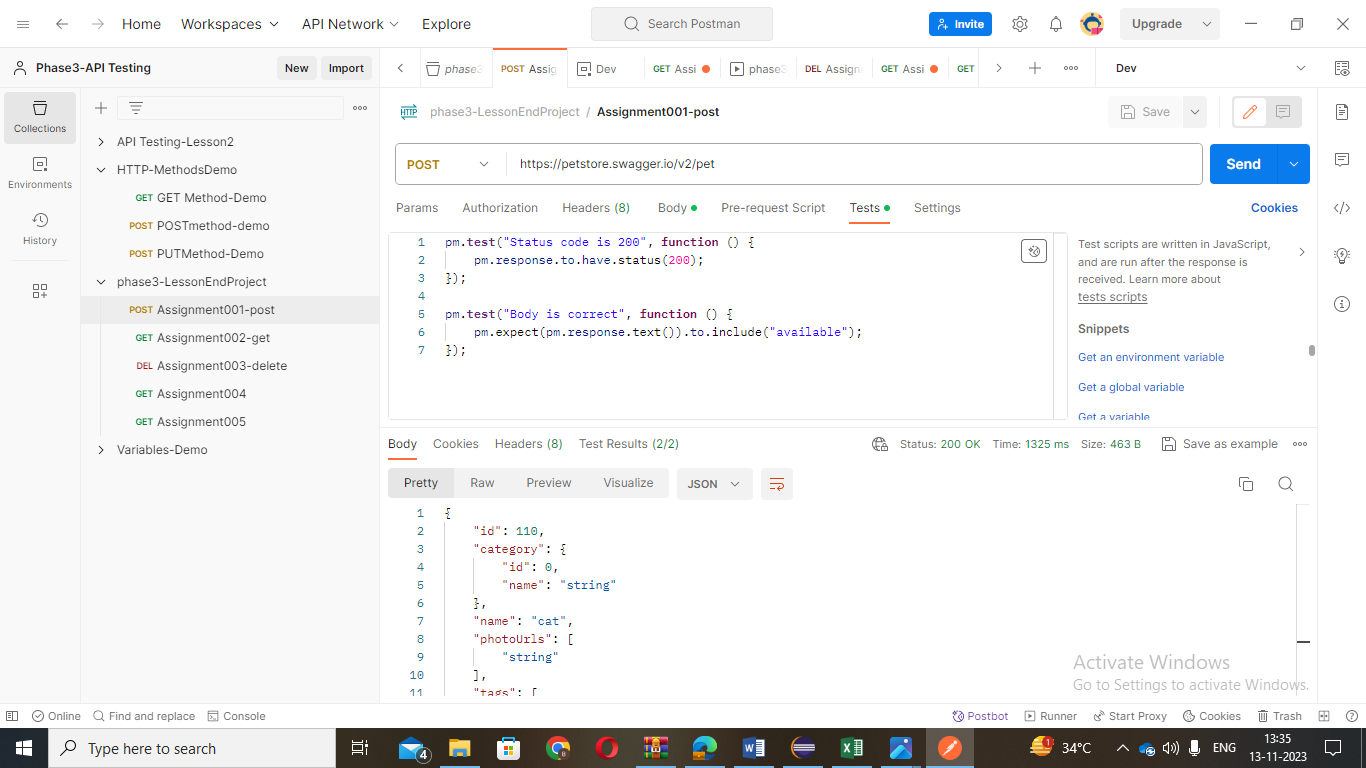
**Source code:**

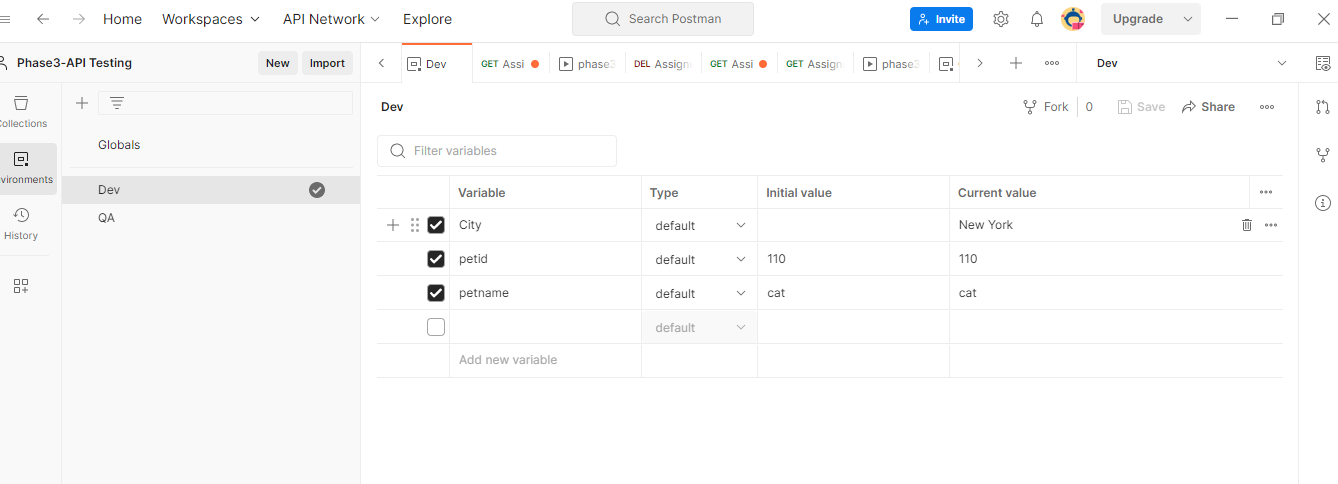
**Postman Assignment\_001**

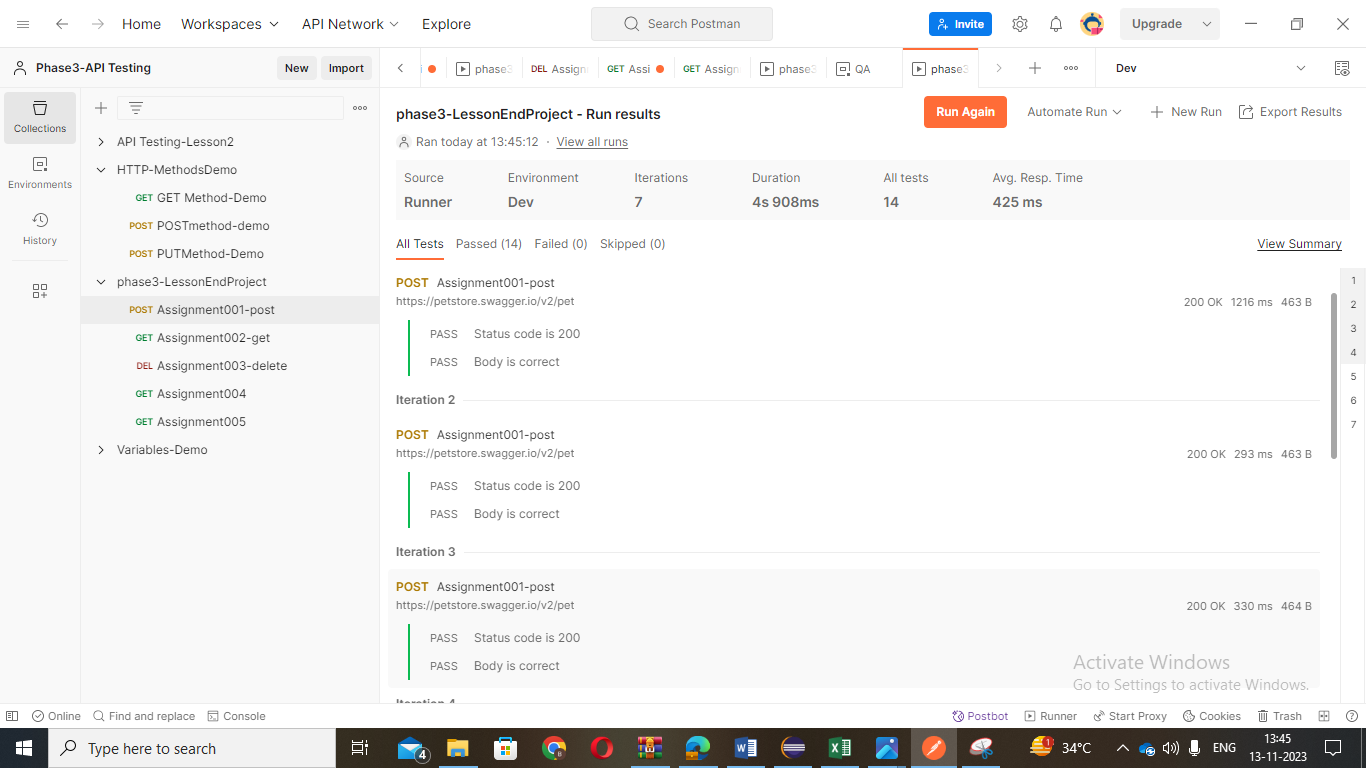
**Post Request**



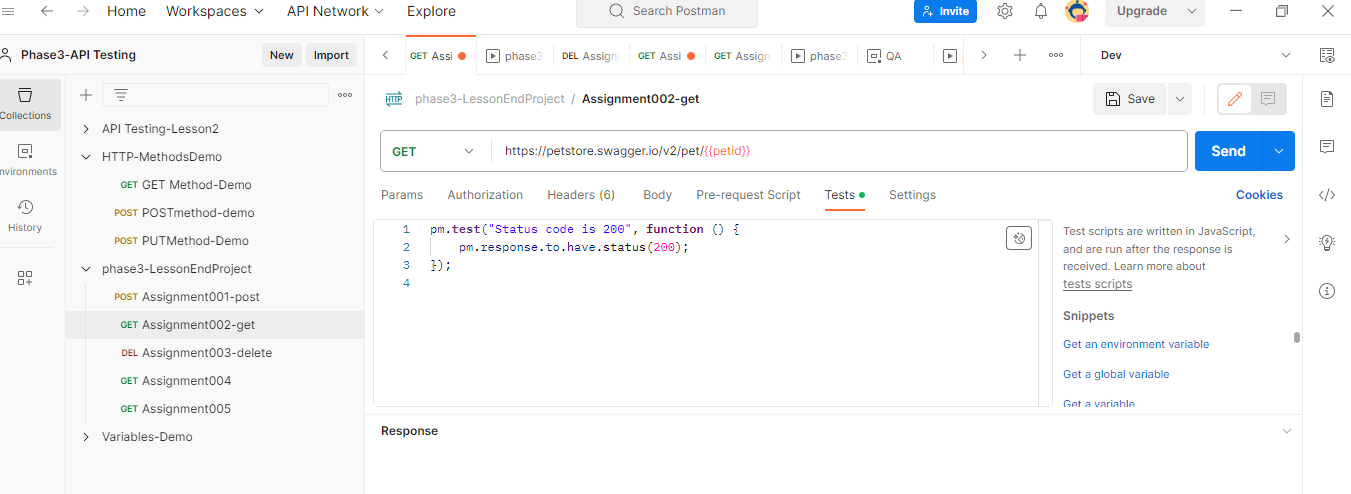
****

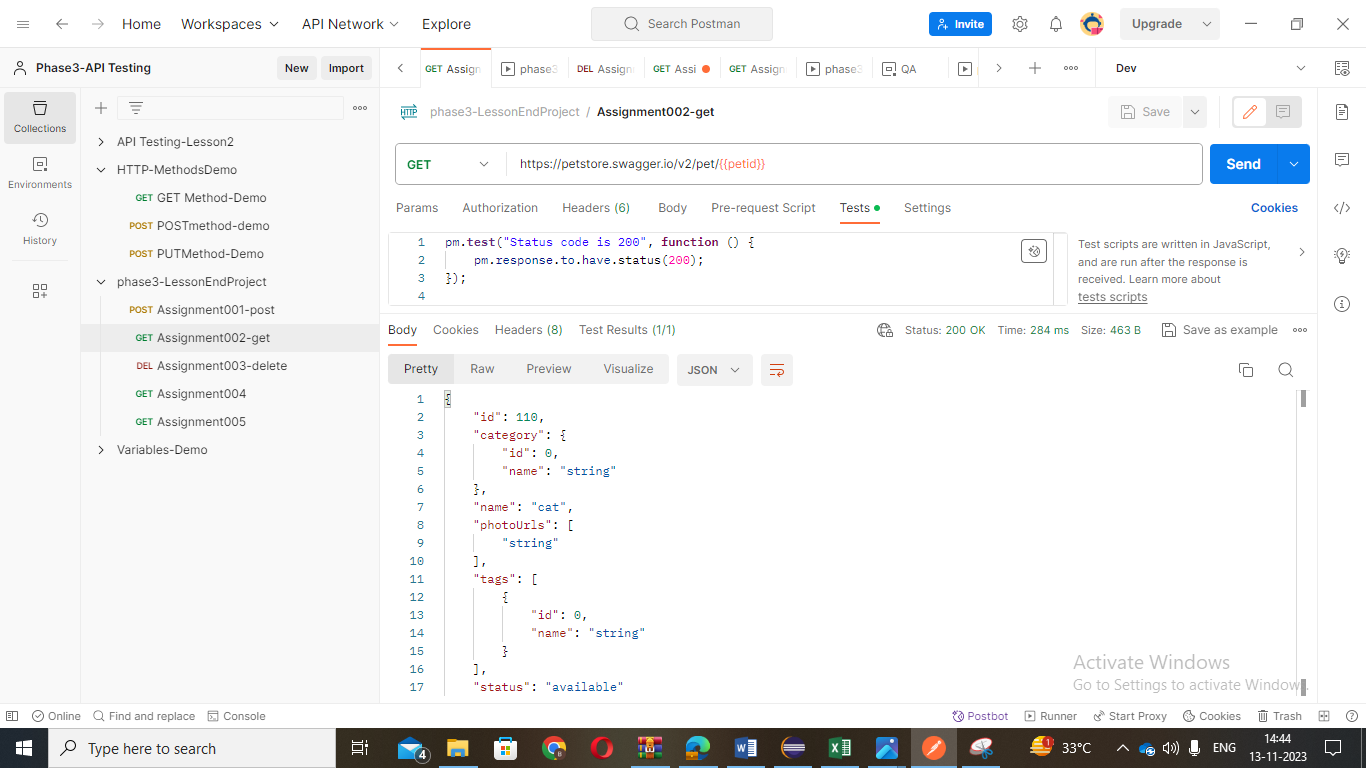
****

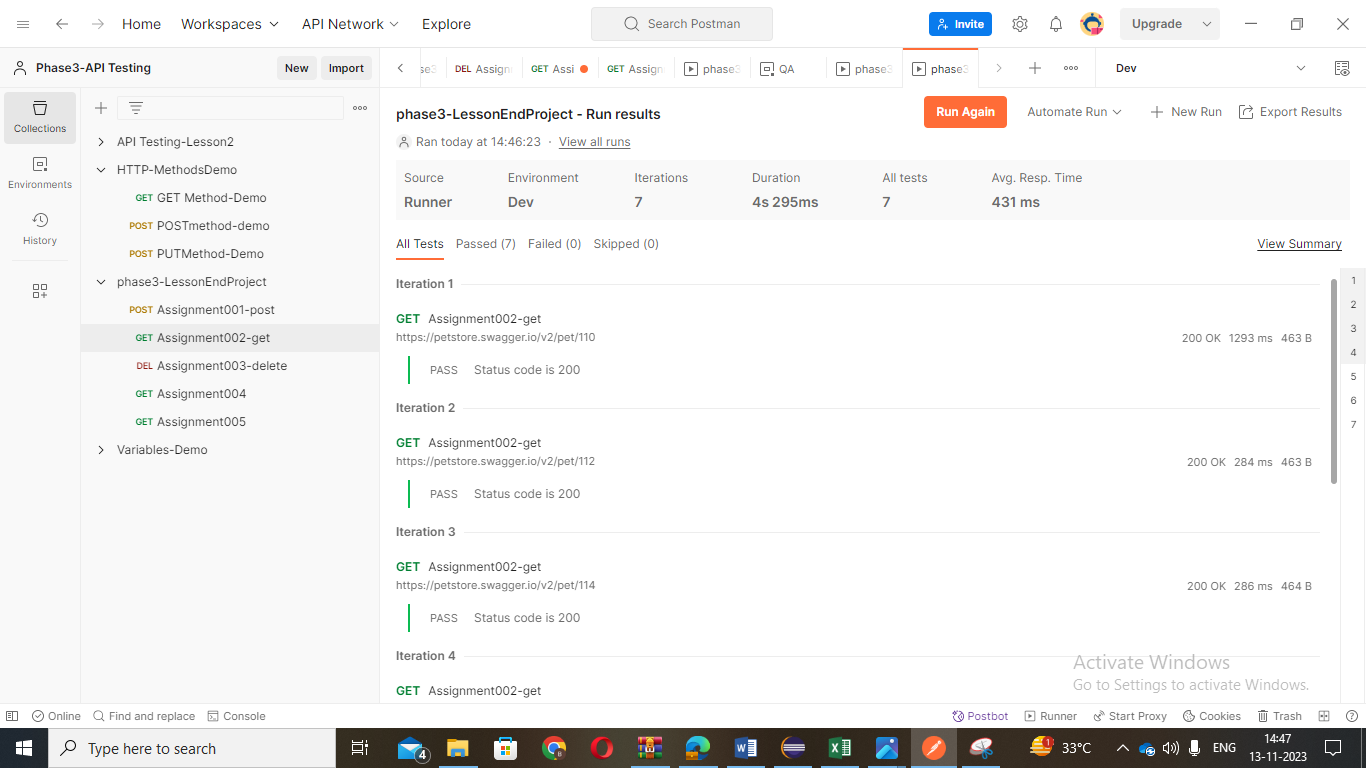


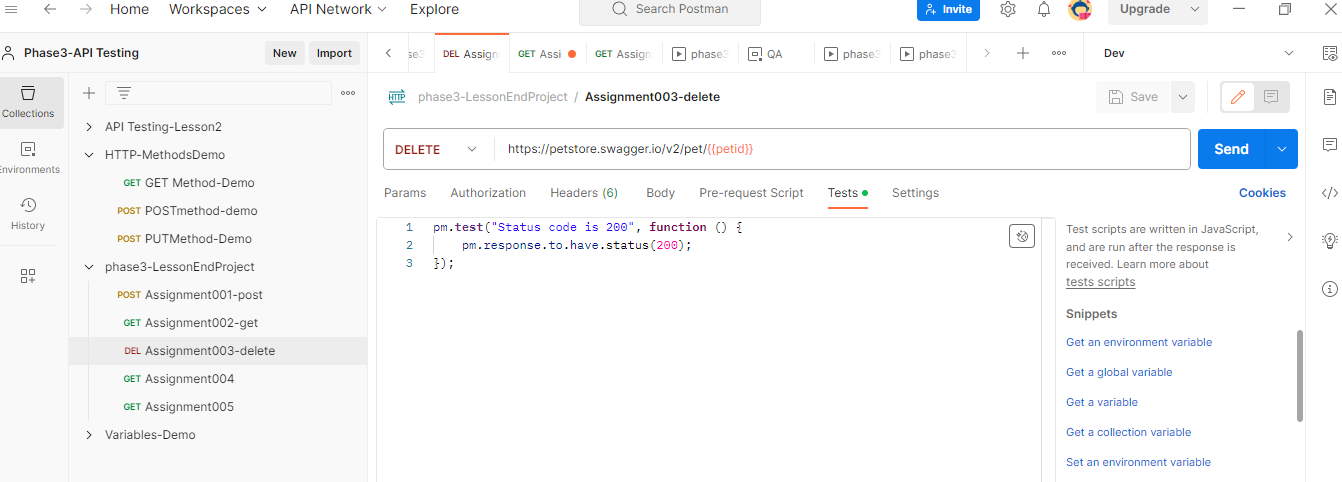


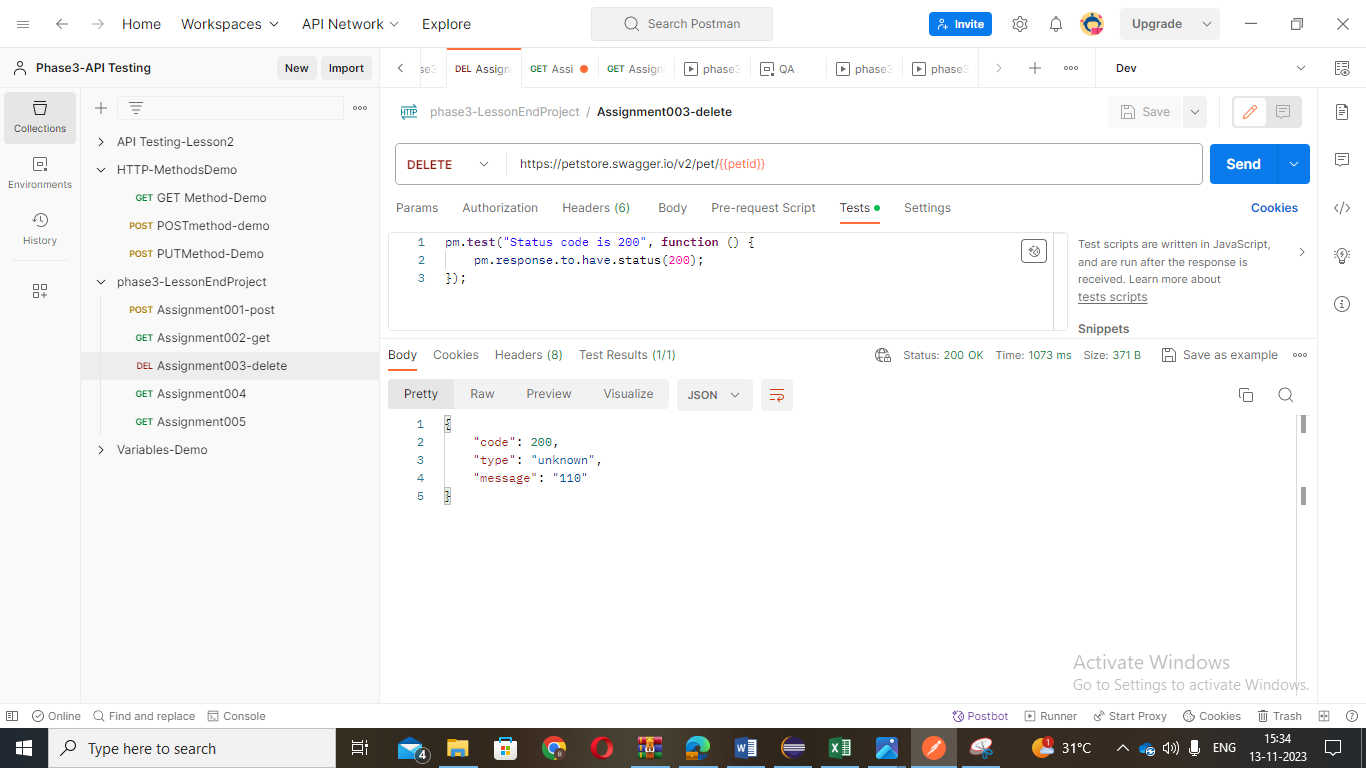
Postman Assignment\_002

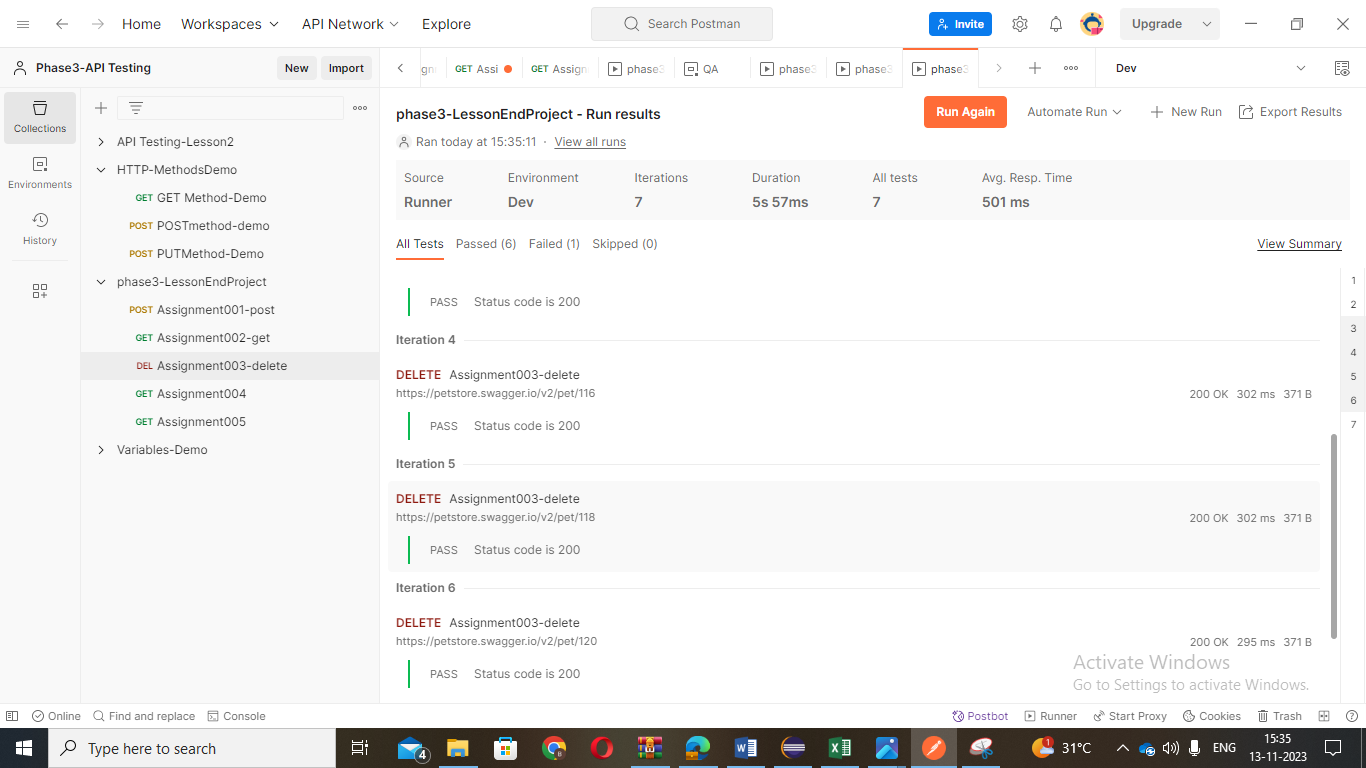
0



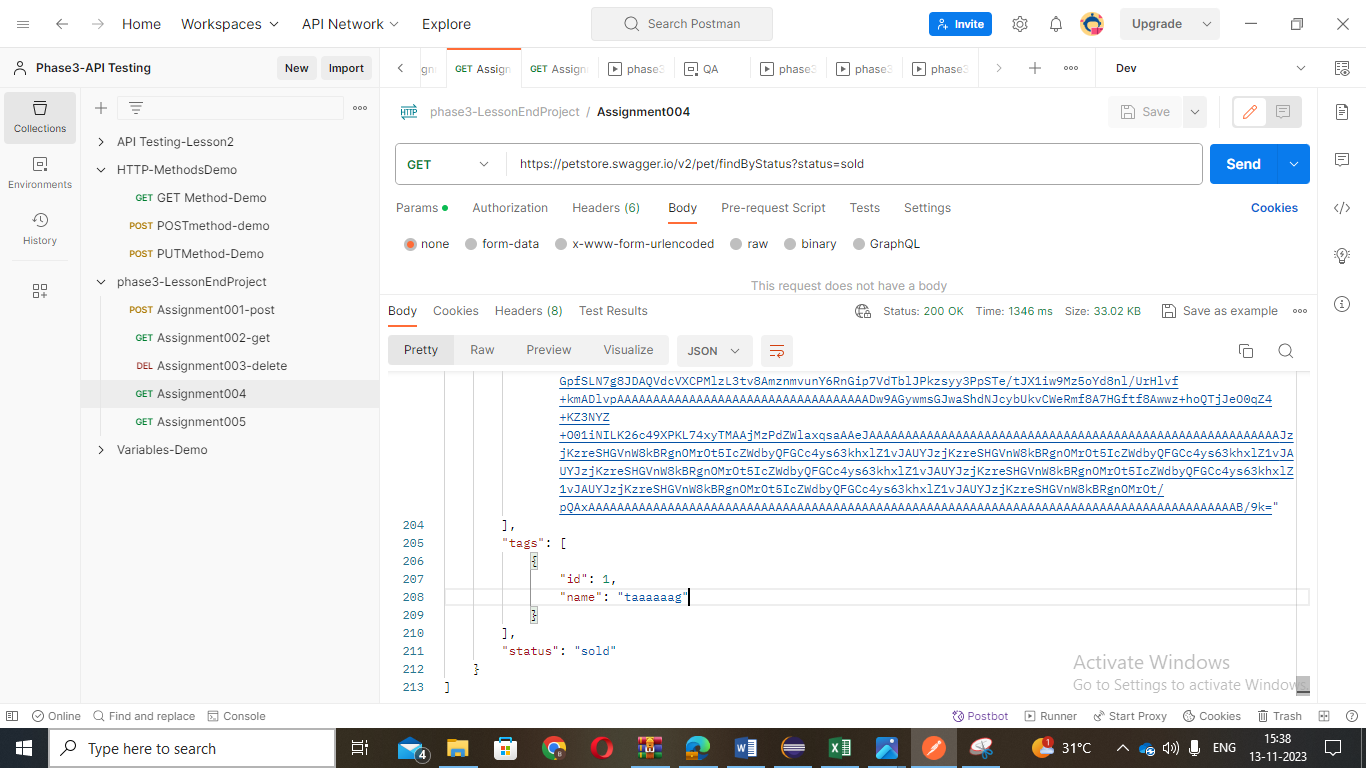


Postman Assignment\_003:

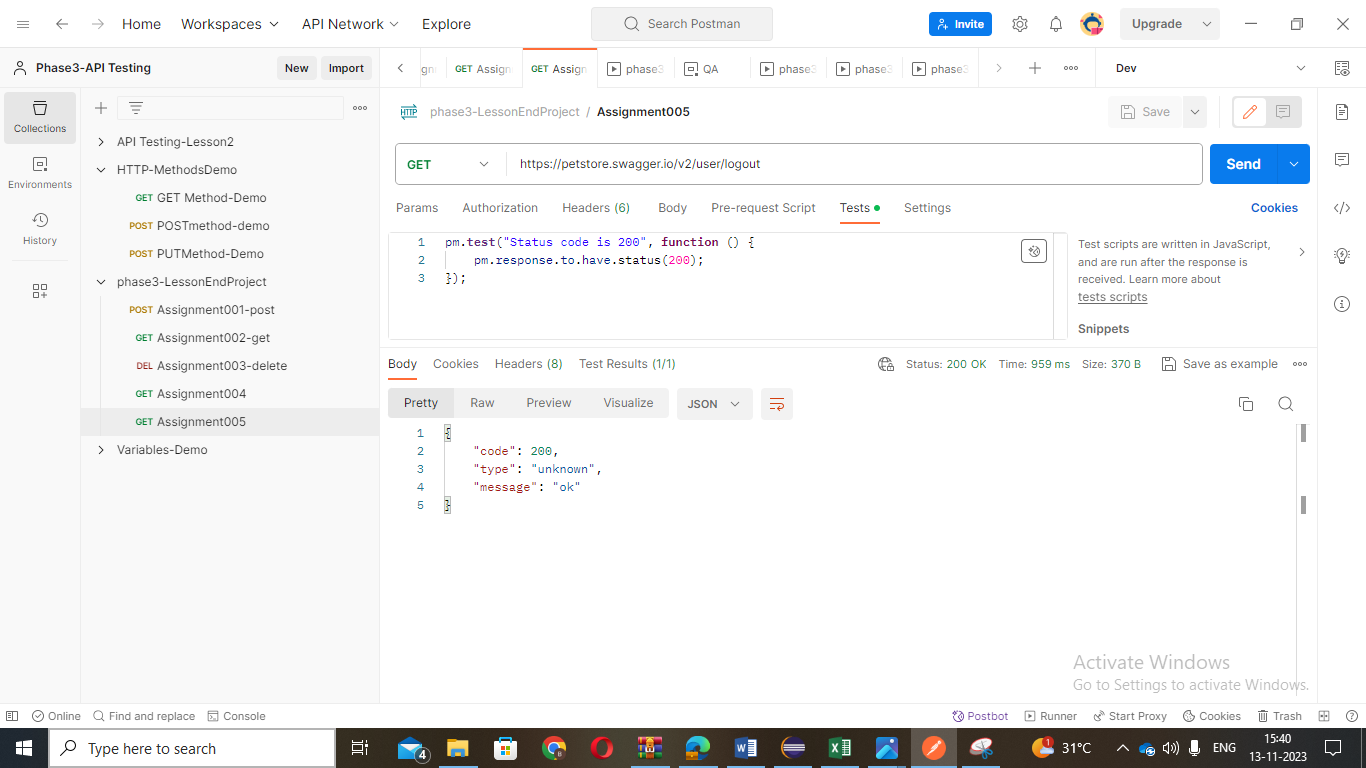
****

****

Postman Assignment\_004



Postman Assignment\_005:



Rest Assured Assignment\_001:

package restassurescripts;

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()

{

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("hen"))

.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/908")

.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/908")

.when()

.delete()

.then()

.statusCode(200)

.log()

.all()

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

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

.body("message", Matchers.equalTo("908"));

}

}

Json File For Post:

{

"id": 910,

"category": {

"id": 0,

"name": "string"

},

"name": "dog",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available"

}

**Rest Assured Assignment\_002**

**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 Assignment002Demo {**

**Logger logger = LogManager.getLogger(Assignment002Demo.class);**

**@Test(priority='1')**

**public void assignment002Post()**

**{**

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

**File file = new File("resource/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("dog")).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("resource/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);**

**}**

**}**

**Json file For Put:**

{

"id": 910,

"category": {

"id": 0,

"name": "string"

},

"name": "dog",

"photo Urls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available\_QA"

}

**Rest Assured Assignment\_003 and 004:**

**package courseEndProject;**

**import org.hamcrest.Matchers;**

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

**import io.restassured.RestAssured;**

**public class Assignment003Demo {**

**@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());**

**}**

**}**

**Rest Assured Assignment\_005 and 006:**

**package courseEndProject;**

**import org.hamcrest.Matchers;**

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

**import io.restassured.RestAssured;**

**public class Assignment005Demo {**

**@Test(priority='1')**

**public void assignment005FindByStatus()**

**{**

**RestAssured.given()**

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

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

**//.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))**

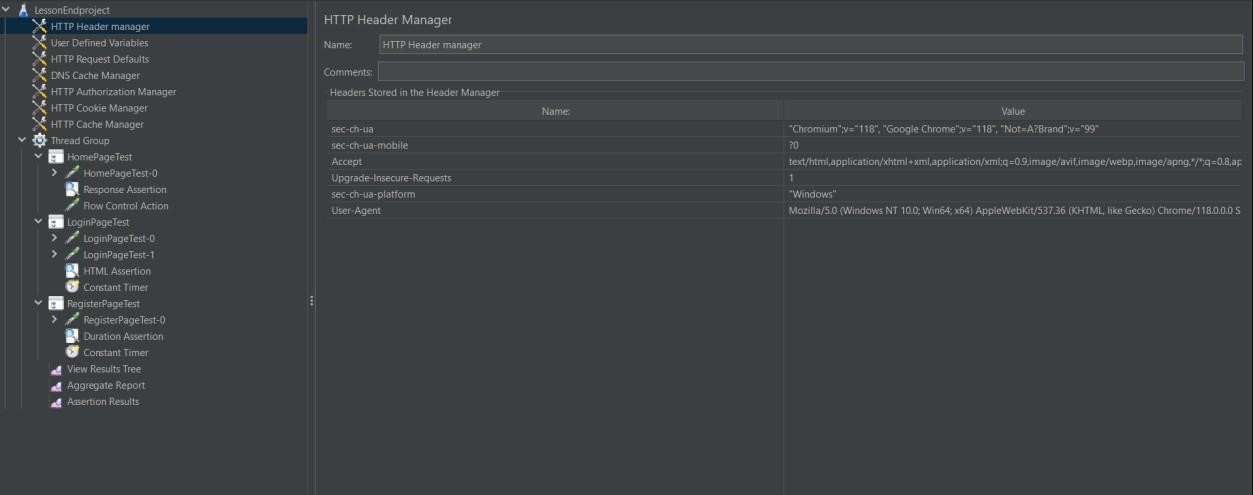
**.body("type", Matchers.equalTo("unknown"))**

**.body("message", Matchers.equalTo("ok"));}**

**}**

**JMeter Assignment:**

Record the Transaction of simple learn page by using Blazemeter and save the file and open JMeter and press file and click open Select the Transaction and open it in Jmeter



Add simple controller and Rename it

