Phase3-EndProject SourceCode:

Github URL: Kirans12345/Phase3EndProjectSL (github.com)

PostMan SourceCode:

- **Postman exported file is uploaded in github.**
- **Sample:**

```
{
	"info": {
		"_postman_id": "a1e02a2b-fcce-4221-8b5b-45dc0e0b113c",
		"name": "PostmanFinalProject",
		"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
		"_exporter_id": "31715051"
	},
	"item": [
		{
			"name": "CreatePetIDass001",
			"event": [
				{
					"listen": "test",
					"script": {
						"exec": [
							"pm.test(\"Status code is 200\", function () {\r",
							"    pm.response.to.have.status(200);\r",
							"});\r",
							"\r",
							"pm.test(\"Body matches string\", function () {\r",
							"    pm.expect(pm.response.text()).to.include(\"available\");\r",
							"});"
						],
						"type": "text/javascript"
					}
				},
				{
					"listen": "prerequest",
					"script": {
						"exec": [
							""
						],
						"type": "text/javascript"
					}
				}
```

```
                        ],
                        "request": {
                                "method": "POST",
                                "header": [],
                                "body": {
                                        "mode": "raw",
                                        "raw": "{\r\n\"id\": {{petID}},\r\n\"category\": {\r\n
\"id\": 0,\r\n \"name\": \"string\"\r\n},\r\n\"name\":{{petName}},\r\n\"photoUrls\": [\r\n
\"string\"\r\n],\r\n\"tags\": [\r\n {\r\n \"id\": 0,\r\n \"name\": \"string\"\r\n }\r\n],\r\n\"status\":
\"available\"\r\n}",
                                        "options": {
                                                "raw": {
                                                        "language": "json"
                                                }
                                        }
                                },
                                "url": {
                                        "raw": "https://petstore.swagger.io/v2/pet",
                                        "protocol": "https",
                                        "host": [
                                                "petstore",
                                                "swagger",
                                                "io"
                                        ],
                                        "path": [
                                                "v2",
                                                "pet"
                                        ]
                                }
                        },
                        "response": []
                },
```

…

………………………Sample…………………………………………………

RestAssured:

All Other classes are uploaded in GitHub.

Sample:

```
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 RestAssured001 {


    Logger logger = LogManager.getLogger(RestAssured001.class);


    @Test(priority='1')
    public void Assignment001Post() {



        logger.info("Course End Project - Assignment001 - Post Request");
        File file = new File("J:\\MPHASIS\\Phase3\\RestAPI Assured\\Phase3_RestAssured\\src\\main\\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/344")
                .when().get()
                .then().statusCode(200)
                .log().all()
                .body("status", Matchers.equalTo("available")).extract().path("category.id");
        System.out.println(id);
        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/344")
                .when().delete()
                .then().statusCode(200)
                .log().all()
                .body("code", Matchers.equalTo(200))
                .body("type", Matchers.equalTo("unknown"))
                .body("message", Matchers.equalTo("344"));

    }
}
```

JMETER Code:

Sample:

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.6.2">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="JMeterSL-FinalProject" enabled="true">
      <boolProp name="TestPlan.functional_mode">false</boolProp>
      <boolProp name="TestPlan.tearDown_on_shutdown">false</boolProp>
      <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
      <elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
    </TestPlan>
    <hashTree>
      <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Thread Group" enabled="true">
        <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
        <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel" testclass="LoopController" testname="Loop Controller" enabled="true">
          <stringProp name="LoopController.loops">1</stringProp>
          <boolProp name="LoopController.continue_forever">false</boolProp>
        </elementProp>
        <stringProp name="ThreadGroup.num_threads">1</stringProp>
        <stringProp name="ThreadGroup.ramp_time">1</stringProp>
        <boolProp name="ThreadGroup.delayedStart">false</boolProp>
        <boolProp name="ThreadGroup.scheduler">false</boolProp>
        <stringProp name="ThreadGroup.duration"></stringProp>
        <stringProp name="ThreadGroup.delay"></stringProp>
        <boolProp name="ThreadGroup.same_user_on_next_iteration">true</boolProp>
      </ThreadGroup>
      <hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="HTTP Request-Authentication" enabled="true">
          <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">
            <collectionProp name="Arguments.arguments">
              <elementProp name="Username" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">Username</stringProp>
                <stringProp name="Argument.value">user</stringProp>
              </elementProp>
              <elementProp name="Password" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.metadata">=</stringProp>
                <boolProp name="HTTPArgument.use_equals">true</boolProp>
                <stringProp name="Argument.name">Password</stringProp>
                <stringProp name="Argument.value">passwd</stringProp>
              </elementProp>
            </collectionProp>
          </elementProp>
          <stringProp name="HTTPSampler.domain">httpbin.org</stringProp>
          <stringProp name="HTTPSampler.protocol">https</stringProp>
          <stringProp name="HTTPSampler.path">/basic-auth/user/passwd</stringProp>
          <stringProp name="HTTPSampler.method">GET</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
```

```
<boolProp name="HTTPSampler.use_keepalive">true</boolProp>

<boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>

<boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>

<boolProp name="HTTPSampler.image_parser">false</boolProp>
```

……..**Uploaded in GitHub**……