**Non-Functional Testing Using Postman, REST Assured, and JMeter**

**(Source-code)**

**AssinmentDemo1.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 AssignmentDemo1 {

Logger logger = LogManager.getLogger(AssignmentDemo1.class);

@Test(priority='1')

public void Assignment001Post() {

logger.info("Course End Project - Assignment001 - Post Request");

File file = new File("C:\\Users\\farha\\eclipse-workspace\\Rest-API\\Phase-3-rest-API\\scr\\main\\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("Doggie")).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"));

}

}

**AssignmentDemo2.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 AssignmentDemo2 {

Logger logger = LogManager.getLogger(AssignmentDemo2.class);

@Test(priority = '1')

public void assignment002Post() {

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

File file = new File(

"C:\\Users\\farha\\eclipse-workspace\\Rest-API\\Phase-3-rest-API\\scr\\main\\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("Doggie"))

.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(

"C:\\Users\\farha\\eclipse-workspace\\Rest-API\\Phase-3-rest-API\\scr\\main\\resource\\dataput.json");

int id = RestAssured.given().baseUri("https://petstore.swagger.io/v2/pet/").contentType(ContentType.JSON)

.body(file).when().put().then().statusCode(200).log().all()

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

System.out.println(id);

}

}

**AssignmentDemoCombo.java**

package courseEndProject;

import org.hamcrest.Matchers;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class AssignmentDemoCombo {

@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());

}

@Test(priority='3')

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='4')

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"))

;

}

}

**Log4jDemo.java**

package courseEndProject;

import org.apache.logging.log4j.LogManager;

import org.apache.logging.log4j.Logger;

import org.testng.annotations.Test;

public class Log4jDemo {

Logger logger = LogManager.getLogger(Log4jDemo.class);

@Test

public void method1() {

System.out.println("this is method1");

logger.trace(" This is trace method");

logger.info(" this is information message");

logger.error(" this is an error message");

logger.fatal(" this is a fatal messgae");

logger.warn(" this is a warning message");

System.out.println("this is end of method1");

}

}

**Pet\_ID\_Testing01.postman\_collection.json**

{

"info": {

"\_postman\_id": "48b62da6-48b4-401b-9d7c-1d3fc579ac90",

"name": "Pet\_ID\_Testing01",

"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",

"\_exporter\_id": "31715014"

},

"item": [

{

"name": "CreatePetID",

"request": {

"method": "POST",

"header": [],

"body": {

"mode": "raw",

"raw": "{\r\n\"id\": 344,\r\n\"category\": {\r\n \"id\": 0,\r\n \"name\": \"string\"\r\n},\r\n\"name\": \"Doggie\",\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": "{{Test\_Uri}}/pet",

"host": [

"{{Test\_Uri}}"

],

"path": [

"pet"

]

}

},

"response": []

},

{

"name": "GetPetID",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "{{Test\_Uri}}/pet/344",

"host": [

"{{Test\_Uri}}"

],

"path": [

"pet",

"344"

]

}

},

"response": []

},

{

"name": "DelPet",

"request": {

"method": "DELETE",

"header": [],

"url": {

"raw": "{{Test\_Uri}}/pet/344",

"host": [

"{{Test\_Uri}}"

],

"path": [

"pet",

"344"

]

}

},

"response": []

}

]

}

**Pet\_ID\_Testing02.postman\_collection.json**

{

"info": {

"\_postman\_id": "3062b97c-c595-4899-b3ac-a51d047588f1",

"name": "Pet\_ID\_Testing02",

"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",

"\_exporter\_id": "31715014"

},

"item": [

{

"name": "CreatePetID",

"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": []

},

{

"name": "GetPetID",

"event": [

{

"listen": "test",

"script": {

"exec": [

"pm.test(\"Status code is 200\", function () {\r",

" pm.response.to.have.status(200);\r",

"});\r",

"pm.test(\"Body matches string\", function () {\r",

" pm.expect(pm.response.text()).to.include(\"available\");\r",

"});"

],

"type": "text/javascript"

}

}

],

"request": {

"method": "GET",

"header": [

{

"key": "accept",

"value": "application/json"

},

{

"key": "api\_key",

"value": "12345"

}

],

"url": {

"raw": "https://petstore.swagger.io/v2/pet/{{petID}}",

"protocol": "https",

"host": [

"petstore",

"swagger",

"io"

],

"path": [

"v2",

"pet",

"{{petID}}"

]

}

},

"response": []

},

{

"name": "DeletePet",

"event": [

{

"listen": "test",

"script": {

"exec": [

"pm.test(\"Status code is 200\", function () {\r",

" pm.response.to.have.status(200);\r",

"});\r",

"pm.test(\"Body matches string\", function () {\r",

" pm.expect(pm.response.text()).to.include(\"unknown\");\r",

"});"

],

"type": "text/javascript"

}

}

],

"request": {

"method": "DELETE",

"header": [

{

"key": "accept",

"value": "application/json"

},

{

"key": "api\_key",

"value": "12345"

}

],

"url": {

"raw": "https://petstore.swagger.io/v2/pet/{{petID}}",

"protocol": "https",

"host": [

"petstore",

"swagger",

"io"

],

"path": [

"v2",

"pet",

"{{petID}}"

]

}

},

"response": []

}

]

}

**QA\_Environment01.postman\_environment.json**

{

"id": "8f56c37e-adf6-4715-bbec-6ada0b435ab2",

"name": "QA\_Environment01",

"values": [

{

"key": "Base\_URL",

"value": "https://api.openweathermap.org/data/2.5/weather",

"type": "default",

"enabled": true

},

{

"key": "apikey\_env",

"value": "ea66dd9dbb8ce34d4a2ad01539908ac9",

"type": "default",

"enabled": true

},

{

"key": "BaseURLDemo",

"value": "https://reqres.in/api/users",

"type": "default",

"enabled": true

},

{

"key": "City",

"value": "Hyderabad",

"type": "any",

"enabled": true

},

{

"key": "email",

"value": "",

"type": "any",

"enabled": true

},

{

"key": "password",

"value": "",

"type": "any",

"enabled": true

},

{

"key": "workspace\_id",

"value": "",

"type": "any",

"enabled": true

},

{

"key": "petID",

"value": "",

"type": "default",

"enabled": true

},

{

"key": "petName",

"value": "",

"type": "default",

"enabled": true

}

],

"\_postman\_variable\_scope": "environment",

"\_postman\_exported\_at": "2023-12-19T11:55:04.489Z",

"\_postman\_exported\_using": "Postman/10.21.7"

}

**data.json**

{

"id": 344,

"category": {

"id": 0,

"name": "string"

},

"name": "Doggie",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available"

}

**dataput.json**

{

"id": 987,

"category": {

"id": 0,

"name": "string"

},

"name": "Doggie",

"photoUrls": [

"string"

],

"tags": [

{

"id": 0,

"name": "string"

}

],

"status": "available\_QA"

}

**log4j2.properties**

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=courseEndProject

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