**API Testing for CURD operation using Rest Assured**

**reassured.java**

package com.phase3\_project;

import org.json.simple.JSONObject;

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import io.restassured.http.Headers;

import io.restassured.http.Method;

import io.restassured.RestAssured;

import io.restassured.response.Response;

import io.restassured.specification.RequestSpecification;

public class reassured {

@BeforeClass

public static void setUpBeforeClass() throws Exception {

}

@AfterClass

public static void tearDownAfterClass() throws Exception {

}

private io.restassured.http.Method Method;

@Before

public void setUp() throws Exception {

}

@After

public void tearDown() throws Exception {

}

@Test

public void test() {

RestAssured.baseURI= "https://reqres.in/api/users";

RequestSpecification given = RestAssured.given();

JSONObject jsonObject = new JSONObject();

jsonObject.put("name", "Varsha");

jsonObject.put("job", "Software Test Engineer");

given.header("content-type", "application/json");

given.body(jsonObject.toString());

Response response = given.request(Method.POST, "/");

System.out.println(response.getStatusCode());

System.out.println(response.getBody().asString());

}

@Test

public void testGetUser() {

RestAssured.baseURI= "https://reqres.in/api/users";

RequestSpecification given = RestAssured.given();

Response response = given.request(Method.GET, "/2");

// System.out.println(response.getBody().asString());

System.out.println(response.getStatusCode());

System.out.println(response.getHeader("Content-Type"));

Headers headers = response.getHeaders();

}

@Test

public void testUpdateUser() {

RestAssured.baseURI= "https://reqres.in/api/users";

RequestSpecification given = RestAssured.given();

JSONObject jsonObject = new JSONObject();

jsonObject.put("name", "Varsha");

jsonObject.put("job", "Software Test Engineer");

given.header("content-type", "application/json");

given.body(jsonObject.toString());

Response response = given.request(Method.PUT, "/");

System.out.println(response.getStatusCode());

System.out.println(response.getBody().asString());

}

@Test

public void testDeleteUser() {

RestAssured.baseURI= "https://reqres.in/api/users";

RequestSpecification given = RestAssured.given();

Response response = given.request(Method.DELETE, "/2");

System.out.println(response.getStatusCode());

}

}

**API Testing for CURD operation using Rest Assured and Log4j**

**InitLogger**

import org.apache.log4j.Level;

import org.apache.log4j.LogManager;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

public class InitLogger {

Logger logger;

public InitLogger(Class<TestRestAssured> class1) {

logger=Logger.getLogger(class1);

PropertyConfigurator.configure("log4j.properties");

logger.setLevel(Level.DEBUG);

}

public InitLogger(String name) {

logger = LogManager.getLogger(name);

PropertyConfigurator.configure("log4j.properties");

logger.setLevel(Level.DEBUG);

}

public void LogInfo(String message) {

logger.info(message);

}

public void LogDebug(String message) {

logger.debug(message);

System.out.println("Added to debug");

}

public void LogWarn(String message) {

logger.warn(message);

}

}

**Logtest.java**

package com.log4j.test;

import org.apache.log4j.BasicConfigurator;

import org.apache.log4j.Logger;

public class Logtest {

static Logger logger=Logger.getLogger(Logtest.class);

public static void main(String[] args) {

BasicConfigurator.configure();

logger.debug("Debugging you know..");

}

}

**TestRestAssured.java**

import static org.junit.Assert.\*;

import org.json.simple.JSONObject;

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import io.restassured.RestAssured;

import io.restassured.http.Method;

import io.restassured.response.Response;

import io.restassured.specification.RequestSpecification;

public class TestRestAssured {

RequestSpecification given;

InitLogger logger;

@BeforeClass

public static void setUpBeforeClass() throws Exception {

}

@AfterClass

public static void tearDownAfterClass() throws Exception {

}

@Before

public void setUp() throws Exception {

logger = new InitLogger(TestRestAssured.class.getName());

RestAssured.baseURI="https://reqres.in/api/users";

given = RestAssured.given();

}

@After

public void tearDown() throws Exception {

}

@Test

public void testGetUser() {

Response response = given.request(Method.GET,"/2");

System.out.println(response.getBody().asString());

String firstName = response.getBody().jsonPath().get("data.first\_name");

System.out.println(firstName);

logger.LogDebug("User got, Status Code--"+response.getStatusCode());

}

@Test

public void testCreateUser() {

RestAssured.baseURI= "https://reqres.in/api/users";

RequestSpecification given = RestAssured.given();

JSONObject jsonObject = new JSONObject();

jsonObject.put("name", "Varsha");

jsonObject.put("job", "Software Test Engineer");

given.header("content-type", "application/json");

given.body(jsonObject.toString());

Response response = given.request(Method.POST, "/");

System.out.println(response.getBody().asString());

logger.LogDebug("User Created, Status Code--"+response.getStatusCode());

}

@Test

public void testUpdateUser() {

RestAssured.baseURI= "https://reqres.in/api/users";

RequestSpecification given = RestAssured.given();

JSONObject jsonObject = new JSONObject();

jsonObject.put("name", "Varsha");

jsonObject.put("job", "Software Test Engineer");

given.header("content-type", "application/json");

given.body(jsonObject.toString());

Response response = given.request(Method.PUT, "/");

System.out.println(response.getStatusCode());

System.out.println(response.getBody().asString());

logger.LogDebug("User Updated, Status Code--"+response.getStatusCode());

}

@Test

public void testDeleteUser() {

RestAssured.baseURI= "https://reqres.in/api/users";

RequestSpecification given = RestAssured.given();

Response response = given.request(Method.DELETE, "/2");

// System.out.println(response.getBody().asString());

System.out.println(response.getStatusCode());

logger.LogDebug("User Deleted, Status Code--"+response.getStatusCode());

}

}

**crawl.log**

1. DEBUG 2022-03-28 11:53:56,947 0 com.log4j.test.TestRestAssured [main] User got, Status Code--200

2. DEBUG 2022-03-28 11:53:57,600 653 com.log4j.test.TestRestAssured [main] User Updated, Status Code--200

3. DEBUG 2022-03-28 11:53:58,091 1144 com.log4j.test.TestRestAssured [main] User Deleted, Status Code--204

4. DEBUG 2022-03-28 11:53:58,653 1706 com.log4j.test.TestRestAssured [main] User Created, Status Code—201

**Log4j.properties**

# initialize root logger with level ERROR for stdout and fout

log4j.rootLogger=ERROR,stdout,fout

# set the log level for these components

log4j.logger.com.endeca=INFO

log4j.logger.com.endeca.itl.web.metrics=INFO

# add a ConsoleAppender to the logger stdout to write to the console

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

# use a simple message format

log4j.appender.stdout.layout.ConversionPattern=%m%n

# add a FileAppender to the logger fout

log4j.appender.fout=org.apache.log4j.FileAppender

# create a log file

log4j.appender.fout.File=crawl.log

log4j.appender.fout.layout=org.apache.log4j.PatternLayout

# use a more detailed message pattern

log4j.appender.fout.layout.ConversionPattern=%p\t%d{ISO8601}\t%r\t%c\t[%t]\t%m%n

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.Phase2\_project.test</groupId>

<artifactId>RestAssured</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>rest-assured</artifactId>

<version>4.5.1</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assured/json-path -->

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>json-path</artifactId>

<version>4.5.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/io.rest-assured/json-schema-validator -->

<dependency>

<groupId>io.rest-assured</groupId>

<artifactId>json-schema-validator</artifactId>

<version>4.5.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.googlecode.json-simple/json-simple -->

<dependency>

<groupId>com.googlecode.json-simple</groupId>

<artifactId>json-simple</artifactId>

<version>1.1.1</version>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.17</version>

</dependency>

</dependencies>

</project>