1.Write a program to set up or configure REST assured.

* Create a Maven project.
* Add following dependencies in pom.xml

<properties>

<serenity.version>1.5.8</serenity.version>

<serenity.maven.version>1.5.8</serenity.maven.version>

<slf4j.version>1.6.1</slf4j.version>

<maven.failsafe.plugin.version>2.18</maven.failsafe.plugin.version>

<maven.compiler.plugin.version>3.2</maven.compiler.plugin.version>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>net.serenity-bdd</groupId>

<artifactId>serenity-core</artifactId>

<version>${serenity.version}</version>

</dependency>

<dependency>

<groupId>net.serenity-bdd</groupId>

<artifactId>serenity-rest-assured</artifactId>

<version>${serenity.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>${slf4j.version}</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.1.0</version>

<scope>test</scope>

</dependency>

<!-- API, java.xml.bind module -->

<dependency>

<groupId>jakarta.xml.bind</groupId>

<artifactId>jakarta.xml.bind-api</artifactId>

<version>2.3.2</version>

</dependency>

<!-- Runtime, com.sun.xml.bind module -->

<dependency>

<groupId>org.glassfish.jaxb</groupId>

<artifactId>jaxb-runtime</artifactId>

<version>2.3.2</version>

</dependency>

</dependencies>

2. Write a program to demonstrate automation of GET request and response.

**package** in.amazon.PhaseEndProject;

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

**import** io.restassured.RestAssured;

**public** **class** GETDemo {

@Test

**public** **void** getServerResponse() {

RestAssured

.*when*()

.get("https://reqres.in/api/unknown")

.then()

.assertThat()

.statusCode(200)

.log().all();

}

}

3. Write a program to demonstrate automation of POST request and response.

**package** in.amazon.PhaseEndProject;

**import** java.util.HashMap;

**import** org.testng.annotations.BeforeMethod;

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

**import** io.restassured.RestAssured;

**public** **class** PostDemo {

HashMap<String, String> map = **new** HashMap<String, String>();

@BeforeMethod

**public** **void** createPayLoad() {

map.put("name", "morpheus");

map.put("job", "leader");

RestAssured.*baseURI* = "https://reqres.in/";

RestAssured.*basePath* = "/api/users";

}

@Test

**public** **void** createResource() {

RestAssured

.*given*()

.contentType("application/json")

.body(map)

.when()

.post()

.then()

.statusCode(201)

.log().all();

}

}

4.Write a program to send XML payload in REST assured, parse XML response in REST assured, parse JSON response in REST assured, and explore native logging of REST assured.

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<CapitalCity xmlns="http://www.oorsprong.org/websamples.countryinfo">

<sCountryISOCode>IN</sCountryISOCode>

</CapitalCity>

</soap:Body>

</soap:Envelope>

**package** in.amazon.RestAssureddemo;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** org.apache.commons.io.IOUtils;

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

**import** io.restassured.RestAssured;

**public** **class** SOAPDemo {

@Test

**public** **void** getCapital() **throws** IOException {

FileInputStream fis = **new** FileInputStream(**new** File("capitals.xml"));

RestAssured.*baseURI* = "http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL";

RestAssured

.*given*()

.header("Content-Type", "text/xml")

.body(IOUtils.*toString*(fis, "UTF-8"))

.when()

.post()

.then()

.statusCode(200)

.log().all();

}

}

Write a program to demonstrate OAuth authorization in REST API using REST assured.

7.Write a program to demonstrate SSL authentication.

package in.amazon.RestAssureddemo;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class APIKeyDemo {

@Test

public void getweatherInfo() {

RestAssured

.given()

.param("q", "chennai")

.param("appid","6d1fd8b68b71f018058d1e583b0f2445" )

.when()

.get("https://api.openweathermap.org/data/2.5/weather")

.then()

.statusCode(200)

.log().all();

}

}

**package** in.amazon.RestAssureddemo;

**import** java.util.HashMap;

**import** java.util.UUID;

**import** org.testng.annotations.BeforeMethod;

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

**import** io.restassured.RestAssured;

**public** **class** BearerTokenDemo {

HashMap<String, String> map = **new** HashMap<String, String>();

UUID uuid = UUID.*randomUUID*();

@BeforeMethod

**public** **void** createPayLoad(){

map.put("name", "Superman");

map.put("email", uuid+"@gmail.com");

map.put("gender", "male");

map.put("status", "active");

RestAssured.*baseURI* = "https://gorest.co.in/";

RestAssured.*basePath* = "public/v2/users";

}

@Test

**public** **void** createResource() {

RestAssured

.*given*()

.contentType("application/json")

.body(map)

.header("Authorization","Bearer 753223d6e7d725acba0ea3fc4bd4963bacbc4142828d235881ed04dc213db839")

.when()

.post()

.then()

.statusCode(201)

.log().all();

}

}

package in.amazon.RestAssureddemo;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class GETDEMOBearerToken {

@Test

public void verifyResource() {

RestAssured

.given()

.contentType("application/json")

.header("Authorization","Bearer 753223d6e7d725acba0ea3fc4bd4963bacbc4142828d235881ed04dc213db839")

.when()

.get("https://gorest.co.in/public/v2/users/4503398")

.then()

.statusCode(200)

.log().all();

}

}

package in.amazon.RestAssureddemo;

import java.util.HashMap;

import org.testng.Assert;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

import io.restassured.path.json.JsonPath;

import io.restassured.response.Response;

public class PUTDemoBearerToken {

HashMap<String, String> map = new HashMap<>();

@BeforeMethod

public void createPayLoad() {

map.put("name", "Superman");

map.put("email", "superman57544766@gmail.com");

map.put("gender", "male");

map.put("status", "active");

RestAssured.baseURI = "https://gorest.co.in/";

RestAssured.basePath = "public/v2/users/4503398";

}

@Test

public void updateResource() {

Response response = RestAssured

.given()

.contentType("application/json")

.header("Authorization","Bearer 753223d6e7d725acba0ea3fc4bd4963bacbc4142828d235881ed04dc213db839")

.body(map)

.when()

.put()

.then()

.extract().response();

JsonPath jsonPath = response.jsonPath();

//System.out.println(jsonPath.get("name").toString());

Assert.assertTrue(jsonPath.get("name").toString().equals("Superman"));

}

}

Write a program to configure Log4j in Eclipse Maven project.

Write a program to generate logs on Eclipse Console using Log4j.

**package** in.amazon.RestAssureddemo;

**import** java.util.HashMap;

**import** java.util.UUID;

**import** org.junit.Assert;

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

**import** io.restassured.RestAssured;

**import** io.restassured.path.json.JsonPath;

**public** **class** create\_Update\_Delete\_Resource **extends** TestBase {

UUID uuid = UUID.*randomUUID*();

**int** id;

JsonPath jsonPath;

HashMap<String, String> map = **new** HashMap<String, String>();

@Test(priority = 0)

**public** **void** createPayLoad() {

map.put("name", "Captain Planet");

map.put("email", uuid+"@gmail.com");

map.put("gender", "male");

map.put("status", "Active");

RestAssured.*baseURI* = "https://gorest.co.in/";

RestAssured.*basePath* = "public/v2/users";

logger.info("Payload Created");

}

@Test(priority = 1)

**public** **void** createResource() {

*response* = RestAssured

.*given*()

.contentType("application/json")

.body(map)

.header("Authorization", "Bearer 753223d6e7d725acba0ea3fc4bd4963bacbc4142828d235881ed04dc213db839")

.when()

.post()

.then()

.extract().response();

logger.info("Resource Created Successfully");

jsonPath = *response*.jsonPath();

id = jsonPath.get("id");

System.***out***.println(id);

}

@Test(priority = 2)

**public** **void** verifyResource() {

System.***out***.println("Id = " +id);

RestAssured

.*given*()

.contentType("application/json")

.header("Authorization", "Bearer 753223d6e7d725acba0ea3fc4bd4963bacbc4142828d235881ed04dc213db839")

.when()

.get("https://gorest.co.in/public/v2/users/"+id)

.then()

.statusCode(200);

logger.info("Resource Verified");

Assert.*assertTrue*(jsonPath.get("name").equals("Captain Planet"));

}

@Test(priority = 3)

**public** **void** updateResource() {

map.put("name", "Captain Planett");

map.put("email", uuid+"@gmail.com");

map.put("gender", "male");

map.put("status", "Active");

RestAssured.*baseURI* = "https://gorest.co.in/";

RestAssured.*basePath* = "public/v2/users/"+id;

RestAssured

.*given*()

.contentType("application/json")

.header("Authorization", "Bearer 753223d6e7d725acba0ea3fc4bd4963bacbc4142828d235881ed04dc213db839")

.body(map)

.when()

.put()

.then()

.statusCode(200);

// .assertThat()

// .body("name",is("Captain planett"));

logger.info("Resource updated");

}

@Test(priority =4)

**public** **void** deleteResource() {

RestAssured.*baseURI* = "https://gorest.co.in/";

RestAssured.*basePath* = "public/v2/users/"+id;

RestAssured

.*given*()

.contentType("application/json")

.header("Authorization", "Bearer 753223d6e7d725acba0ea3fc4bd4963bacbc4142828d235881ed04dc213db839")

.when()

.delete()

.then()

.statusCode(204);

logger.info("Resource Deleted");

}

}

package in.amazon.RestAssureddemo;

import org.testng.annotations.BeforeClass;

import org.apache.log4j.Level;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

import io.restassured.response.Response;

import io.restassured.specification.RequestSpecification;

public class TestBase {

public static RequestSpecification httpRequest;

public static Response response;

public Logger logger;

@BeforeClass

public void setUp() {

logger = Logger.getLogger("LogDemo");

PropertyConfigurator.configure("log4j.properties");

logger.setLevel(Level.DEBUG);

}

}

Write a program to write log file using Log4j.

log4j.rootLogger=ERROR,stdout

log4j.logger.com.endeca=INFO

log4j.logger.LogDemo=DEBUG, dest1

# Logger for crawl metrics

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

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

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

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