import com.fasterxml.jackson.core.JsonProcessingException;

import com.google.gson.Gson;

import io.restassured.http.ContentType;

import io.restassured.response.Response;

import io.restassured.response.ResponseBody;

import org.testng.Assert;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import static io.restassured.RestAssured.given;

import static io.restassured.RestAssured.when;

public class ApiTutorialsTests {

@DataProvider(name = "cities")

public Object[][] dataProviderMethod() {

return new Object[][]{

{"London"},

{"Bishkek"},

{"Batken"},

{"Cholpon-Ata"},

{"Austin"}};

}

@Test

public void TestGetData() {

Response response = when().get("http://www.google.com");

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

Assert.assertNotEquals(null, response.asString());

}

@Test(dataProvider = "cities")

public void TestWeatherOfEachCityStatusCode(String city) {

Response response = when().get("http://api.openweathermap.org/data/2.5/weather?q=" + city + "&appid=2e0068a6768bc24abf94c41f07083571");

System.out.print(response.statusCode());

Assert.assertEquals(200, response.statusCode());

}

@Test(dataProvider = "cities")

public void TestWeatherOfEachCityNames(String city) {

Response response = when().get("http://api.openweathermap.org/data/2.5/weather?q=" + city + "&appid=2e0068a6768bc24abf94c41f07083571");

System.out.print(response.asString());

//Assert.assertEquals();

}

@DataProvider(name = "authors")

public Object[][] GetAuthors() {

return new Object[][]{

{"typicode", 1},

{"Nurbek Ibragim", 2},

{"Nurbek Ibragim", 3}

};

}

@Test(dataProvider = "authors")

public void TestGetPosts(String author, int id) throws JsonProcessingException {

Response response = null;

try {

response = given()

.contentType(ContentType.JSON)

.get("http://localhost:3000/posts/"+ id);

Assert.assertEquals(response.statusCode(), 200);

ResponseBody body = response.getBody();

//System.out.println(body.asString());

Gson g = new Gson();

Post post = g.fromJson(body.asString(), Post.class);

Assert.assertEquals(post.author, author);

}

catch (Exception e){

}

}

@Test

public void TestCreatePosts(){

String requestBody = "{\n" +

" \"id\": 5,\n" +

" \"title\": \"About DevX\",\n" +

" \"author\": \"Nurbek Ibragim\"\n" +

"}";

Response response = null;

try {

response = given()

.contentType(ContentType.JSON)

.body(requestBody)

.post("http://localhost:3000/posts");

Assert.assertEquals(response.statusCode(), 201);

}

catch (Exception e){

}

}

@Test

public void TestUpdatePost(){

String requestBody = "{\n" +

" \"id\": 2,\n" +

" \"title\": \"About DevX\",\n" +

" \"author\": \"Nurbek Azamat\"\n" +

"}";

Response response = null;

try {

response = given()

.contentType(ContentType.JSON)

.body(requestBody)

.put("http://localhost:3000/posts/2");

Assert.assertEquals(response.statusCode(), 200);

}

catch (Exception e){

}

}

@Test

public void TestDeletePost(){

Response response = null;

try {

response = given()

.contentType(ContentType.JSON)

.delete("http://localhost:3000/posts/2");

Assert.assertEquals(response.statusCode(), 200);

}

catch (Exception e){

}

}

}