Source Code phase end

package ReqRes.demo;

import static org.hamcrest.Matchers.is;

import java.util.HashMap;

import java.util.UUID;

import org.junit.Assert;

import org.testng.annotations.Test;

import com.Restassured.RestAssured.TestBase;

import io.restassured.RestAssured;

import io.restassured.path.json.JsonPath;

import io.restassured.response.Response;

public class Create\_Update\_Delete\_Reqres extends TestBase {

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

UUID uuid = UUID.randomUUID();

int id;

Response response;

JsonPath jsonPath;

@Test(priority=0)

public void creataPayload() {

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

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

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

RestAssured.basePath = "/api/users";

logger.info("Payload created");

}

@Test(priority=1)

public void createResource() {

response = RestAssured

.given()

.contentType("application/json")

.body(map)

.when()

.post()

.then()

.extract().response();

logger.info("Resource Created Successfully");

jsonPath = response.jsonPath();

id = jsonPath.getInt("id");

System.out.println(id);

}

@Test(priority=2)

public void verifyResource() {

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

RestAssured

.given()

.contentType("application/json")

.when()

.get("https://reqres.in/api/users?page=2"+id)

.then()

.statusCode(200);

logger.info("Resource Verified");

Assert.assertTrue(jsonPath.get("name").equals("mallikarjun"));

}

@Test(priority=3)

public void updateResource() {

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

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

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

RestAssured.basePath = "/api/users/"+ id;

RestAssured

.given()

.contentType("application/json")

.body(map)

.when()

.put()

.then()

.statusCode(200)

.assertThat()

.body("name", is("bothamallikarjun"));

logger.info("Resource Updated");

}

@Test(priority=4)

public void deleteResource() {

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

RestAssured.basePath = "/api/users/"+ id;

RestAssured

.given()

.contentType("application/json")

.when()

.delete()

.then()

.statusCode(204);

logger.info("Resource Deleted");

}

}

----------------------------------------------------------------------------------

package ReqRes.demo;

import org.apache.log4j.Level;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

import org.testng.annotations.BeforeClass;

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);

}

}

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