Cucumber Rest Assured Integration

Cucumber is an [open source testing framework](https://www.pcloudy.com/blogs/the-best-java-testing-frameworks-to-focus-in-2021/) that supports Behavior Driven Development (BDD) and is a widely used framework that testers leverage to write test cases in simple English language called Gherkin.

To make the testing process and testing flow easily understandable by non-techies, many organizations want to integrate Cucumber framework with Selenium or Rest Assured for application automation. Since Cucumber allows writing tests and scenarios in simple English plain text, it becomes a primary advantage for many team members who are Business Analysts, Project Managers, Manual Testers and for any other non-technical persons to view and understand the testing workflow with much ease.

# Adding the Cucumber – TestNG – Rest Assured jar files

Add the below jar files:

* cucumber-java
* cucumber-core
* cucumber-jvm-deps
* cucumber-reporting
* gherkin
* testng
* cucumber-testng
* rest-assured
* json-simple

# Create Folder structure

1. Create Features folder in src/test/resources
2. Create a package for StepDefination and Test Runner in src/test/java

# Create Feature file, Step definition and runner classes

--------------------------Feature file-----------------------------

@tag

Feature: Verify different REST API

Verification of GET/POST/PUT/DELETE API methods

Scenario Outline: Reqres GET API test

Given I provide valid endpoint to fetch users

When I send request to the server with page number as <page>

Then I validate the response of first user record having email as <emailID>

Examples:

| page |emailID |

| 2 |michael.lawson@reqres.in|

| 1 |george.bluth@reqres.in |

---------------------Step definition--------------------------------

**public** **class** StepDefinations {

Response response;

@Given("I provide valid endpoint to fetch users")

**public** **void** i\_provide\_valid\_endpoint\_to\_fetch\_users() {

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

}

@When("^I send request to the server with page number as (.\*)$")

**public** **void** i\_sned\_request\_to\_the\_server\_with\_page\_number\_as(String pageNumber) {

response = RestAssured.*given*().header("Content-Type", "application/json").queryParam("page",pageNumber).request(Method.***GET***);

System.***out***.println(response.asPrettyString());

}

@Then("^I validate the response of first user record having email as (.\*)$")

**public** **void** i\_validate\_the\_response(String emailID) {

JsonPath path = **new** JsonPath(response.asString());

Assert.*assertEquals*(emailID, path.getString("data[0].email"));

}

}

--------------------------Test Runner----------------------------------

@CucumberOptions(tags = "", features = {"src/test/resources/Features"}, glue = {"stepDefinations"})

**public** **class** TestRunner **extends** AbstractTestNGCucumberTests {

}

# Performing POST

Scenario Outline: Reqres POST API test

Given I provide valid endpoint to "create" users

When I send request to the server with user name as <username> and job as <job>

Then I verify the status code is <statusCode>

Then I verify a new user is created with name as <username>

Examples:

| username | job | statusCode |

| john | HR | 201 |

----------------------------Step Def-----------------------

@When("^I send request to the server with user name as (.\*) and job as (.\*)$")

**public** **void** the\_request\_is\_send\_to\_the\_server(String name, String job) {

JSONObject requiredParam = **new** JSONObject();

requiredParam.put("name", name);

requiredParam.put("job", job);

response = RestAssured.*given*().header("Content-Type", "application/json").body(requiredParam.toJSONString()).when().post();

}

@Then("^I verify the status code is (.\*)$")

**public** **void** IVerifyStatusCode(**int** statusCode) {

Assert.*assertEquals*(statusCode, response.getStatusCode());

}

@Then("^I verify a new user is created with name as (.\*)$")

**public** **void** the\_new\_user\_must\_be\_created(String userName) {

JsonPath jPath = **new** JsonPath(response.asString());

Assert.*assertEquals*(userName, jPath.getString("name"));

}