

Reto Rest Assured

RESPONSABLE: JULIO ELIAS VASQUEZ ROJAS

Introducción

Este documento se describe la solución para la ejecución de pruebas automatizadas de API REST utilizando las bibliotecas Cucumber, Rest-Assured y JUnit utilizando el lenguaje Java. Se han elaborado pruebas sobre dos páginas como recurso:

- reqres.in.
- jsonplaceholder.typicode.com

Reqres

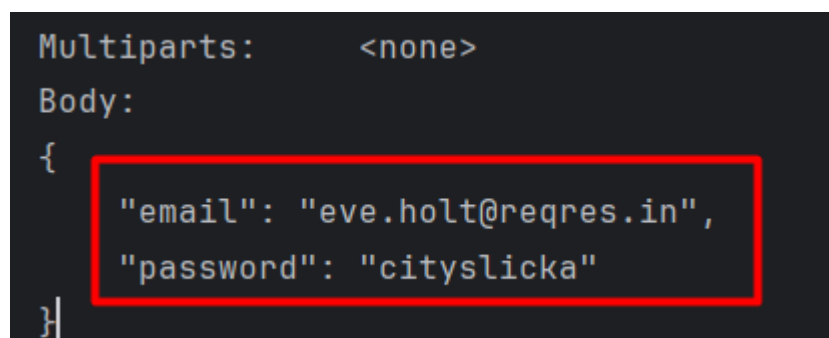
POST: LOGIN – SUCCESSFUL

1. Inicio de sesión de usuario en la página Reqres, usando el servicio POST

- Página Reqres



- Cuerpo de la solicitud



- Token de respuesta

```
CF-RAY: 87fbd5c418be0a2a-MIA

{
  "token": "QpwL5tke4Pnpja7X4"
}

{"token":"QpwL5tke4Pnpja7X4"}
```

- Escenario de prueba


```
loguearUsuario.feature
1 Feature: loguear usuario en Regres
2   yo como usuario registrado de Regres
3   quiero loguearme en el sistema
4   para poder acceder a las funcionalidades
5   #Autor: Julio Vasquez
6
7   Background:
8     Given el usuario se encuentra en la pagina inicio de sesion
9
10  @task1
11  Scenario Outline: Inicio de sesion exitoso
12    When ingresa su email <email> con la contraseña <contrasena>
13    Then deberia recibir una respuesta con el codigo de estado 200
14    And deberia recibir un token
15    Examples:
16      | email | contrasena |
17      | "eve.holt@reqres.in" | "1548" |
18      | "eve.holt@reqres.in" | "hola2" |
```

- Assert


```
log4j:WARN No appenders could be found for logger (org.apache.http.impl.conn.BasicClientConnectionManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Cuerpo de la solicitud: {"email": "eve.holt@reqres.in", "password": "1548"}
codigo de respuesta: 200
HTTP/1.1 200 OK
Date: Tue, 07 May 2024 22:36:04 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 29
Connection: keep-alive
Report-To: {"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku
```

■ Cucumber


100% passed
2 executed


Windows 11

3 minutes ago
Last Run


Java HotSpot(TM) 64-Bit Server VM
17.0.10+11-LTS-240

3.3 seconds
Duration


cucumber-jvm 6.11.0

Delete Report

file:///C:/Users/vasqu/OneDrive/SofkaU/SEMANA%2010/2024-C1-QA-AP-T02/ApiRestAssured/src/test/resources/features/loguearUsuario.feature

Report Edit

Feature: loguear usuario en Reqres

yo como usuario registrado de Reqres quiero loguearme en el sistema para poder acceder a las funcionalidades

Background:

Given el usuario se encuentra en la pagina inicio de sesion

task1

Scenario Outline: Inicio de sesion exitoso

When ingresa su email <email> con la contraseña <contrasena>

Then deberia recibir una respuesta con el codigo de estado 200

And deberia recibir un token

Examples:

	email	contrasena
✓	"eve.holt@reqres.in"	"1548"
✓	"eve.holt@reqres.in"	"hola2"

Resultado: Éxito, los criterios de validación se han cumplido. Confirmando los resultados obtenidos. La prueba se realizó en el Date: Tue, 07 May 2024 22:36:05 GMT

GET: SINGLE USER

2. Obtener un usuario de la página Reqres, usando el servicio GET

■ Página Reqres

Request

/api/users/2

Response

200

GET

LIST USERS

GET

SINGLE USER

GET

SINGLE USER NOT FOUND

GET

LIST <RESOURCE>

GET

SINGLE <RESOURCE>

GET

SINGLE <RESOURCE> NOT FOUND

{

"data": {

"id": 2,

"email": "janet.weaver@reqres.in",

"first_name": "Janet",

"last_name": "Weaver",

"avatar": "https://reqres.in/img/faces/2-image-placeholder.png"

},

"support": {

"url": "https://reqres.in/#support-headlines",

"text": "To keep ReqRes free, contributions towards server costs are appreciated."

}

}

- Cuerpo de la solicitud y respuesta

```
Server: cloudflare
CF-RAY: 8804d3168d258e02-MIA
Content-Encoding: gzip

{
  "data": {
    "id": 1,
    "email": "george.bluth@reqres.in",
    "first_name": "George",
    "last_name": "Bluth",
    "avatar": "https://reqres.in/img/faces/1-image.jpg"
  },
  "support": {
    "url": "https://reqres.in/#support-heading",
    "text": "To keep ReqRes free, contributions towards server costs are appreciated!"
  }
}




Enlace de la solicitud: /users/12
codigo de respuesta: 200
id: 12
HTTP/1.1 200 OK
Date: Tue, 07 May 2024 23:00:10 GMT
Content-Type: application/json; charset=utf-8
```

- Escenario

```
obtenerUsuario.feature x ObtenerUserRunner.java ObtenerUsuarioStepDefinition.java
1 Feature: obtener usuario por id en Regres
2   yo como usuario registrado de Regres
3   quiero obtener la informacion de un usuario en el sistema
4   para poder acceder a su informacion
5   #Autor: Julio Vasquez
6
7   Background:
8     Given el usuario se encuentra en la pagina de perfil
9   @task2
10  Scenario Outline: Obtener la informacion de un usuario
11    When se pide al servicio la informacion por su id <id>
12    Then deberia recibir una respuesta con el codigo 200
13    And deberia recibir la informacion del usuario <id>
14    Examples:
15      | id |
16      | "1" |
17      | "12" |
```

■ Cucumber

2 PASSED

100% passed 2 executed	3 minutes ago Last Run	2.96 seconds Duration
 Windows 11	 Java HotSpot(TM) 64-Bit Server VM 17.0.10+11-LTS-240	 cucumber-jvm 6.11.0

Delete Report

✓ file:///C:/Users/vasqu/OneDrive/SofkaU/SEMANA%2010/2024-C1-QA-AP-T02/ApiRestAssured/src/test/resources/features/obtenerUsuario.feature

Report Edit

Feature: obtener usuario por id en Reqres

yo como usuario registrado de Reqres quiero obtener la informacion de un usuario en el sistema para poder acceder a su informacion

Background:

- ✓ **Given** el usuario se encuentra en la pagina de perfil

@task2

Scenario Outline: Obtener la informacion de un usuario

- ✓ **When** se pide al servicio la informacion por su id <id>
- ✓ **Then** deberia recibir una respuesta con el codigo 200
- ✓ **And** deberia recibir la informacion del usuario <id>

Examples:

	id
✓	"1"
✓	"12"

Resultado: Éxito, los criterios de validación se han cumplido. Confirmando los resultados obtenidos. La prueba se realizó en el Date: Tue, 07 May 2024 23:00:10 GMT

{JSON} Placeholder

POST: ROUTER POSTS

3. Crear una publicación en la página {JSON} Placeholder, usando el servicio POST

■ Página {JSON} Placeholder

Routes

All HTTP methods are supported. You can use http or https for your requests.

GET	/posts
GET	/posts/1
GET	/posts/1/comments
GET	/comments?postId=1
POST	/posts
PUT	/posts/1
PATCH	/posts/1
DELETE	/posts/1

Note: see [guide](#) for usage examples.

- Cuerpo de la solicitud y respuesta

```
✓ Tests passed: 1 of 1 test - 2 sec 463 ms

"C:\Program Files\Java\jdk-17\bin\java.exe" ...
log4j:WARN No appenders could be found for logger (org.apache.http.impl.conn.BasicClientConnectionManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Cuerpo de la solicitud: {"title": "Prueba", "body": "Prueba del servicio", "userId": 1}
codigo de respuesta: 201
HTTP/1.1 201 Created
Date: Tue, 07 May 2024 23:09:05 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 84
Connection: keep-alive
```




- Escenario

```
crearPost.feature  CrearPostRunner.java  CrearPostStepDefinition.java

1  Feature: Gestión de los servicios de correos
2    yo como usuario del servicio JsonPlaceholder
3    quiero gestionar mis correos
4    para manejar mi contenido eficientemente
5    #Autor: Julio Vasquez
6
7  Background:
8    Given el usuario esta en la pagina web de jsonplaceholder
9
10
11 @task3
12 Scenario: crear un nuevo correo exitosamente
13   When escribe la informacion para el nuevo correo "Prueba" "Prueba del servicio"
14   And envia la solicitud
15   Then deberia obtener un codigo de respuesta 201
16   And el titulo debe ser "Prueba"
```

- Cucumber

1 PASSED

100% passed 1 executed	6 minutes ago Last Run	2.48 seconds Duration
 Windows 11	 Java HotSpot(TM) 64-Bit Server VM 17.0.10+11-LTS-240	 cucumber-jvm 6.11.0

Delete Report

✓ file:///C:/Users/vasqu/OneDrive/SofkaU/SEMANA%2010/2024-C1-QA-AP-T02/ApiRestAssured/src/test/resources/features/crearPost.feature

Report Edit

Feature: Gestión de los servicios de correos

yo como usuario del servicio JsonPlaceholder quiero gestionar mis correos para manejar mi contenido eficientemente

Background:

- ✓ Given el usuario esta en la pagina web de jsonplaceholder

@task3

Scenario: crear un nuevo correo exitosamente

- ✓ When escribe la informacion para el nuevo correo "Prueba" "Prueba del servicio"
- ✓ And envia la solicitud
- ✓ Then deberia obtener un codigo de respuesta 201
- ✓ And el titulo debe ser "Prueba"

Resultado: Éxito, los criterios de validación se han cumplido. Confirmando los resultados obtenidos. La prueba se realizó en el Date: Tue, 07 May 2024 23:09:05 GMT

GET: USER WITH ID

4. Obtener usuarios en la página {JSON} Placeholder, usando el servicio GET

- Página {JSON} Placeholder

Resources

JSONPlaceholder comes with a set of 6 common resources:

/posts	100 posts
/comments	500 comments
/albums	100 albums
/photos	5000 photos
/todos	200 todos
/users	10 users

Note: resources have relations. For example: posts have many comments, albums have many photos, ... see [guide](#) for the full list.

- Cuerpo de la solicitud y respuesta

```
log4j:WARN No appenders could be found for logger (org.apache.http.impl.conn.BasicClientConnectionManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
HTTP/1.1 200 OK
Date: Tue, 07 May 2024 23:18:16 GMT
Content-Type: application/json; charset=utf-8
```




```
[
  {
    "userId": 1,
    "id": 1,
    "title": "delectus aut autem",
    "completed": false
  },
  {
    "userId": 1,
    "id": 2,
    "title": "quis ut nam facilis et officia qui",
    "completed": false
  },
  {
    "userId": 1,
    "id": 3,
    "title": "fugiat veniam minus",
    "completed": false
  },
]
```

▪ Escenario

```
listarUsuario.feature x  ListarUsuarioRunner.java  ListarUsuarioStepDefinition.java
1  Feature: listar usuarios por userId
2      yo como usuario del servicio JsonPlaceholder
3      quiero revisar mi lista de usuarios
4      para alcanzar los objetivos
5      #Autor: Julio Vasquez
6
7      Background:
8          Given un usuario administrador en la pagina web de jsonplaceholder
9
10     @task4
11     Scenario Outline: listar todos los todos por Id
12         And consulta por userId <id>
13         When envía la solicitud todos los usuarios
14         Then la cantidad de todos debe ser 20
15         Examples:
16             | id |
17             | 1 |
```

▪ Cucumber

1 PASSED

100% passed 1 executed	7 minutes ago Last Run	2.73 seconds Duration
 Windows 11	 Java HotSpot(TM) 64-Bit Server VM 17.0.10+11-LTS-240	 cucumber-jvm 6.11.0

Delete Report

file:///C:/Users/vasqu/OneDrive/SofkaU/SEMANA%2010/2024-C1-QA-AP-T02/ApiRestAssured/src/test/resources/features/listarUsuario.feature

Report Edit

Feature: listar usuarios por userId

yo como usuario del servicio JsonPlaceholder quiero revisar mi lista de usuarios para alcanzar los objetivos

Background:

- Given un usuario administrador en la pagina web de jsonplaceholder

@task4

Scenario Outline: listar todos los todos por Id

- And consulta por userId <id>
- When envía la solicitud todos los usuarios
- Then la cantidad de todos debe ser 20

Examples:

id
1

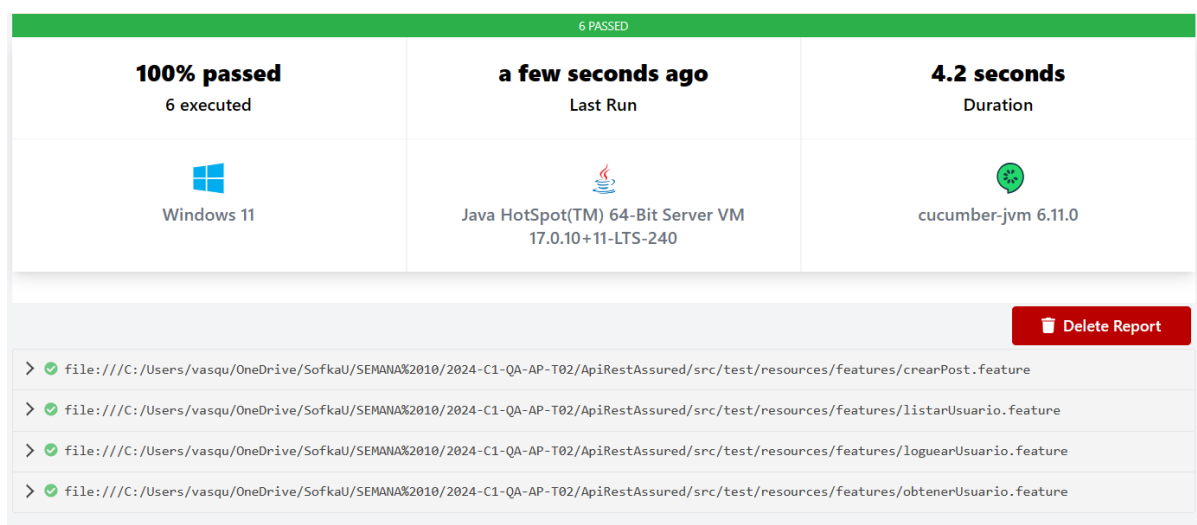
Resultado: Éxito, los criterios de validación se han cumplido. Confirmando los resultados obtenidos. La prueba se realizó en el Date: Tue, 07 May 2024 23:09:05 GMT

Configuración del Proyecto

El proyecto está configurado utilizando Gradle con Groovy, con las siguientes dependencias principales:

- Cucumber para escribir y ejecutar pruebas de aceptación utilizando lenguaje natural, con el modelo Gherkins.
- JUnit Jupiter para las pruebas unitarias.
- Rest-Assured para realizar pruebas de API REST.

Ejecución de todas las pruebas



Resultado Exitoso. El tiempo total de ejecución de todas las pruebas es de 4.2 segundos.

En la ejecución simultánea de las pruebas, se observa que el 100% de ellos pasa exitosamente, cumpliendo con los criterios de validación y verificando los asserts de manera satisfactoria. Se observa concordancia en los códigos de estado HTTP esperado y recibido, lo que resulta en un éxito en la prueba y su aprobación.

Conclusión

Mediante esta solución, es posible mantener la calidad del software de manera efectiva a lo largo del ciclo de vida del desarrollo. La implementación proporciona un conjunto de pruebas automatizadas para los recursos de API REST. Se emplea una combinación de pruebas unitarias de servicios y de aceptación para asegurar la funcionalidad y la integridad de los servicios web.