

Pruebas de Api con Rest-Assured

Katherin Oquendo

Introducción

Este documento presenta una solución para la realización de pruebas automatizadas de API REST utilizando las bibliotecas Rest-Assured, JUnit, y Cucumber en un proyecto Java. Se han realizado pruebas sobre dos conjuntos de recursos: uno proporcionado por jsonplaceholder.typicode.com y otro por reqres.in.

Test	Descripción	Resultado Esperado	Resultado
testGetSinglePost	Obtiene detalles de un post existente	Código de estado 200, respuesta con los campos específicos	Pasó correctamente
testGetSingleUser	Obtiene detalles de un usuario existente	Código de estado 200, respuesta con los campos específicos	Pasó correctamente

```
Tests passed: 1 of 1 tests - Time: 351 ms
C:\Program Files\Java\jdk-11.0.10\java.exe ...
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "body": "quia et suscipit\nsuscipit recusanda consequuntur expedita et cum\nreprehenderit molestiae ut ut quas tota\n"
```

```
Tests passed: 2 of 2 tests - Time: 654 ms
C:\Program Files\Java\jdk-11.0.10\java.exe ...
{
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "body": "quia et suscipit\nsuscipit recusanda consequuntur expedita et cum\nreprehenderit molestiae ut ut quas tota\n"
```

Pruebas de Regres API

Test	Descripción	Resultado Esperado	Resultado
testListUsersApi	Lista todos los usuarios	Código de estado 200	Pasó correctamente
testGetSingleUserById	Obtiene detalles de un usuario existente	Código de estado 200	Pasó correctamente

```
✓ Tests passed: 1 of 1 test – 1sec 980 ms

"C:\Program Files\Java\jdk-11\bin\java.exe" ...

Process finished with exit code 0
```

```
Response Code: 200
Response Body: {"page":1,"per_page":6,"total":12,"data":[{"id":1,"email":"george.bluth@regres.io","first_name":"George","last_name":"Bluth","avatar":"https://randomuser.me/api/?avatar=120x120","id":1,"email":"emma.watson@regres.io","first_name":"Emma","last_name":"Watson","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":2,"email":"emma.watson@regres.io","first_name":"Emma","last_name":"Watson","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":3,"email":"emma.watson@regres.io","first_name":"Emma","last_name":"Watson","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":4,"email":"eve.holt@regres.io","first_name":"Eve","last_name":"Holt","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":5,"email":"eve.holt@regres.io","first_name":"Eve","last_name":"Holt","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":6,"email":"tracy.ramos@regres.io","first_name":"Tracy","last_name":"Ramos","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":7,"email":"tracy.ramos@regres.io","first_name":"Tracy","last_name":"Ramos","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":8,"email":"tracy.ramos@regres.io","first_name":"Tracy","last_name":"Ramos","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":9,"email":"tracy.ramos@regres.io","first_name":"Tracy","last_name":"Ramos","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":10,"email":"tracy.ramos@regres.io","first_name":"Tracy","last_name":"Ramos","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":11,"email":"tracy.ramos@regres.io","first_name":"Tracy","last_name":"Ramos","avatar":"https://randomuser.me/api/?avatar=120x120"}, {"id":12,"email":"tracy.ramos@regres.io","first_name":"Tracy","last_name":"Ramos","avatar":"https://randomuser.me/api/?avatar=120x120"}], "support": [{"id":1,"email":"support@regres.io","text":"To keep Hughes free, contributions towards server costs"}]
Total items: 12

Process finished with exit code 0
```

Escenarios de Cucumber

Feature	Descripción	Resultado Esperado	Resultado
Pruebas del Recurso de Posts	Escenarios para probar el recurso de posts	Se verifica estado de respuesta y contenido	Todos los escenarios pasaron correctamente
Pruebas del Recurso de Usuarios	Escenarios para probar el recurso de usuarios	Se verifica estado de respuesta y contenido	Todos los escenarios pasaron correctamente
API Listar Usuarios	Escenarios para probar el API de listar usuarios	Se verifica estado de respuesta y contenido	Todos los escenarios pasaron correctamente
API Usuario Individual	Escenarios para probar el API de usuario individual	Se verifica estado de respuesta y contenido	Todos los escenarios pasaron correctamente

Configuración del Proyecto

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

JUnit Jupiter para las pruebas unitarias.

Rest-Assured para realizar pruebas de API REST.

Apache HttpClient para realizar solicitudes HTTP.

Cucumber para escribir y ejecutar pruebas de aceptación utilizando lenguaje natural.

Conclusion

La solución proporciona un conjunto completo de pruebas automatizadas para los recursos de API REST de jsonplaceholder.typicode.com y reqres.in. Utiliza una combinación de pruebas unitarias y de aceptación para garantizar la funcionalidad y la integridad de los servicios web. Con esta solución, se puede mantener la calidad del software de manera efectiva a lo largo del ciclo de vida del desarrollo.