

Hola a todos.

He subido la colección de Postman revisada, y con tests para comprobar que lo que devuelve el servicio está bien.

Algunos tests además sirven para analizar el performance, y puede que también para detectar si hay algún problema con la red. Me explico:

Ejemplo de test:

The screenshot shows the Postman interface for a REST client request. The request is a GET method to the URL `http://127.0.0.1:8090/api/v1/geocode/find?type=calle&portal=710&idcalle=8770`. The 'Tests' tab is active, displaying two test scripts:

```
1 pm.test("Body matches string", function () {
2   pm.expect(pm.response.text()).to.include("PLAZA INDEPENDENCIA 710");
3 });
4
5 pm.test("Response time is less than 200ms", function () {
6   pm.expect(pm.response.responseTime).to.be.below(200);
7 });
```

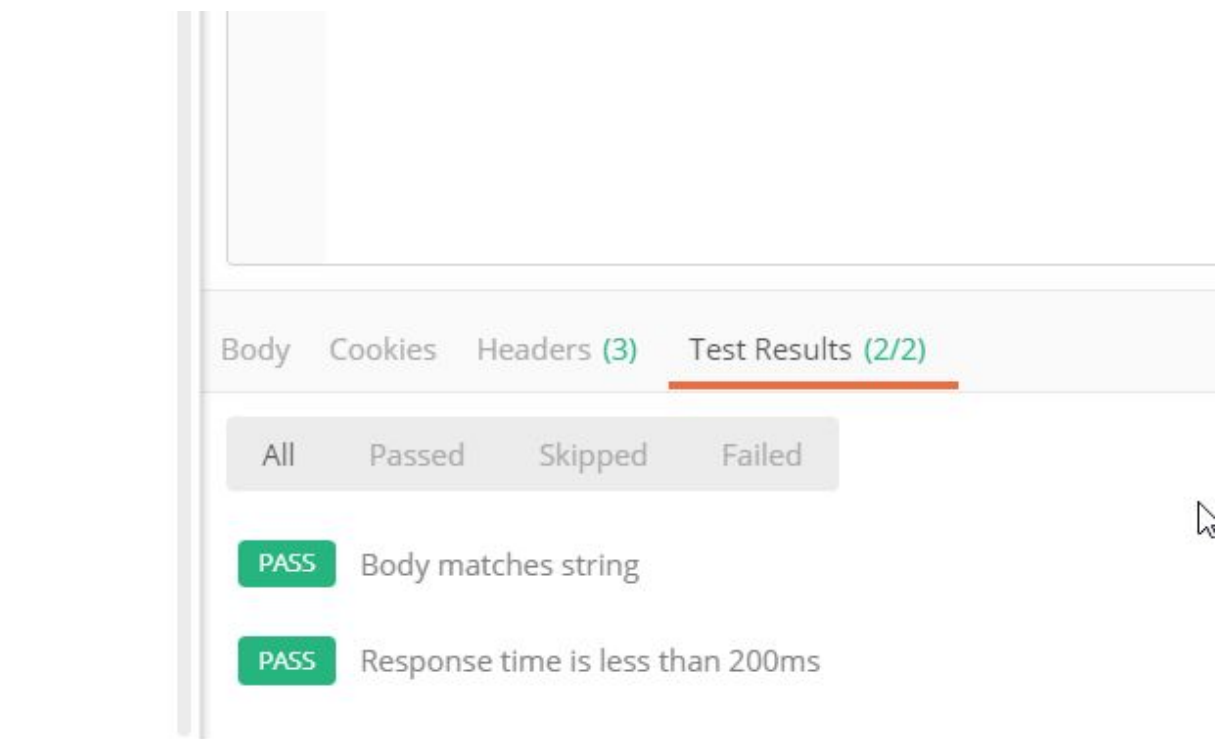
Below the test scripts, the 'Test Results (1/2)' section shows the following results:

- PASS** Body matches string
- FAIL** Response time is less than 200ms | AssertionError: expected 485 to be below 200

The status bar at the bottom indicates a status of 200 OK, a time of 485ms, and a size of 596 B.

Abajo a la izquierda podéis ver que no ha pasado el test de velocidad (lo he lanzado cuando el servicio estaba arrancando todavía).

Si lo lanzo en condiciones normales pasa los dos tests:



Hay distintos tipos de test, el que uso aquí comprueba que hay un texto en la respuesta (PLAZA INDEPENDENCIA 710) y que la petición ha tardado menos de 200 msecs:

```
pm.test("Body matches string", function () {  
  pm.expect(pm.response.text()).to.include("PLAZA INDEPENDENCIA 710");  
});
```

```
pm.test("Response time is less than 200ms", function () {  
  pm.expect(pm.response.responseTime).to.be.below(200);  
});
```

Esto servirá para hacer las pruebas de manera manual, os permite a vosotros completar con todos los tests que necesitéis y además indica al técnico de Openshift que no hay problemas de red que ralentizan las peticiones entre el servicio y la base de datos.

Estoy mirando también cómo podemos automatizarlo (integrarlo con el pipeline de Jenkins - OpenShift), y seguramente podremos, pero si no tenemos tiempo, al menos los tests se pueden pasar de manera semi automática (y creo que también pueden servir para las pruebas de carga y performance).

Además, he subido la librería Java para conectar a los servicios de geocodificación, e incluye un ejemplo de uso para desarrolladores (aunque tal y como dije, quizás se pueda utilizar también para las pruebas, porque procesa ficheros en el formato del que me pasasteis para probar)

@Santiago => El repositorio lo he tenido que crear aquí:.

<http://gitlab.paas.agesic.red.uy/francisco.penarrubia/direccionesuy-client.git>

Por favor, cuando puedas, a lo mejor habría que crearlo en SCD (yo no tengo permiso).

Espero que sirva.

Saludos!

Fran.

PS: Lo de OpenShift lo he estado pensando, y [mañana](#) probaré a montar el entrono ci/cd y todas las pipelines. Seguramente necesitaré ayuda, usaré el grupo de Skype.

--

Francisco José Peñarrubia

SCOLAB Software Colaborativo

<http://www.scolab.es>