

Lab 10: API gateway, Zipkin, Sleuth, Hystrix

Exercise 1: API gateway.

Write an API gateway in a separate Spring boot project, and let this API gateway be a gateway between the webshop client and the other weshop services. Use the code from Lab 9, so that the API gateway works together with the service registry.

With the API Gateway, the webshop client should only know the URL of the gateway.

Write a pre and a post filter that writes every request and response to the console.

Exercise 2: Using Zipkin and Sleuth

Start Zipkin by double clicking the file **C:\architecturetraining\zipkin\startzipkin.bat**

Write 3 services A, B, and C. A should call B and B calls C.

Then add the Zipkin and Sleuth libraries and configuration to the 3 services.

Call service A with REST.

Then open the zipkin console on <http://localhost:9411/zipkin/index.html>

You can then see the traces between the services, and you can see the dependencies between the services.

Finally add the Zipkin and Sleuth libraries and configuration to the webshop microservices, including the API gateway of exercise 1.

Exercise 3:

In the 3 services A, B, and C of exercise 2, add the Hystrix circuit breaker to the remote calls and test its working.

Spring Boot Version: 2.0.3

Frequently Used:

- | | | |
|--|---|---------------------------------------|
| <input type="checkbox"/> Config Client | <input type="checkbox"/> Eureka Discovery | <input type="checkbox"/> HSQLDB |
| <input type="checkbox"/> JPA | <input type="checkbox"/> Kafka Streams | <input type="checkbox"/> MongoDB |
| <input type="checkbox"/> Reactive Web | <input checked="" type="checkbox"/> Web | <input type="checkbox"/> Web Services |

Available:

Type to search dependencies

- Azure
- Cloud AWS
- ▼ Cloud Circuit Breaker
 - ☒ Hystrix
 - ☐ Hystrix Dashboard
 - ☐ Turbine
 - ☐ Turbine Stream

Selected:

- X Hystrix
- X Web