Load balancing

load balancing between the multiple incidents of currency exchange from currency conversion. You might think there is a lot of complex code that is needed to be able to do that.

The thing is, it's very, very simple to implement.

**the Feign client to talk to Eureka and pick up the instances of currency exchange**

**and do load balancing between them.**

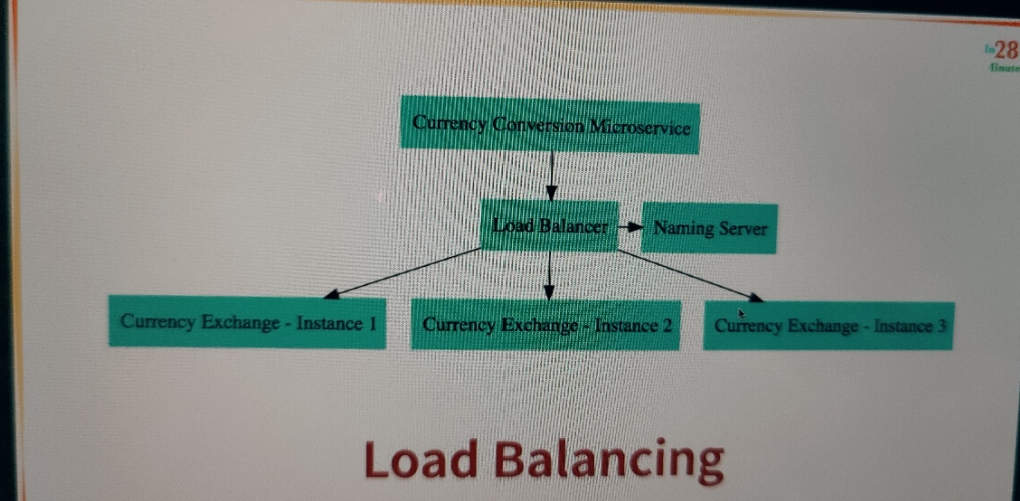
@FeignClient(name = "car\_service", url = "http://localhost:8000")

Just remove url TO

@FeignClient(name = "car\_service")

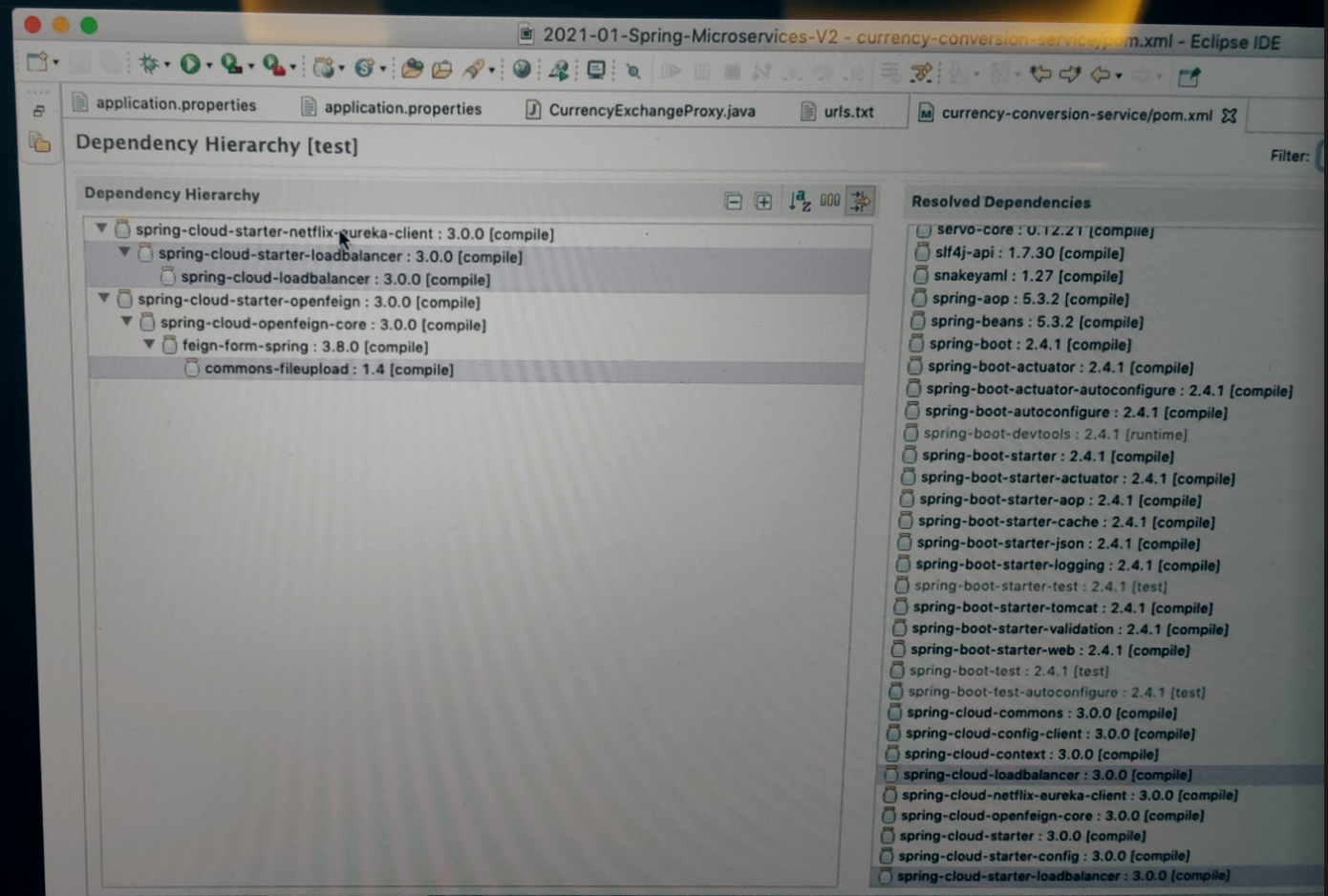
When two or more instance of same microservice start what would happen?

It would resist with Eureka. So it's registering with Eureka.



inside the currency conversion microservice there is load balancer component

which is talking to the naming server, finding the instances and doing automatic load balancing between them. And this is what is called client side load balancing. And this is happening through Feign.



spring cloud started load balancer which is brought into the class path by spring-Cloud-start-up-Netflix-Eureka-client. And this is the load balancer framework that is used by Feign to actually distribute the load among the multiple instances which are returned by Eureka.

In the earlier versions of Spring Cloud the load balancer which was used was **Ribbon**