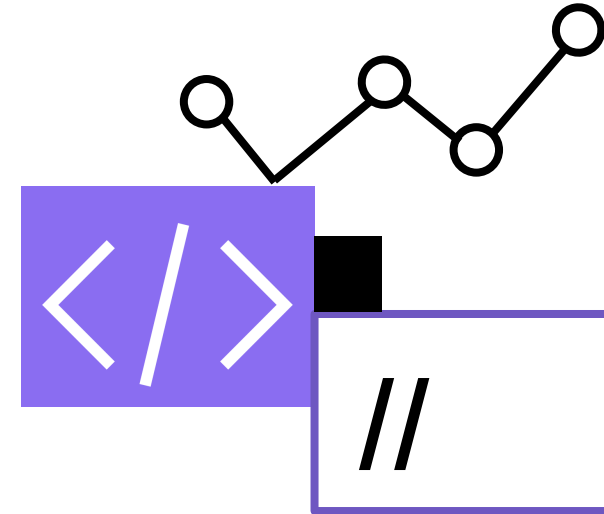


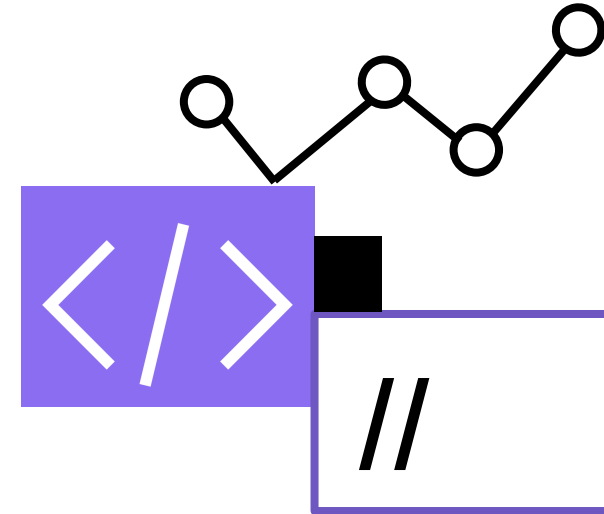
Balanceamento de carga com Eureka, Feign e Spring Cloud Load Balancer



Configuração de Spring Cloud Load Balancer

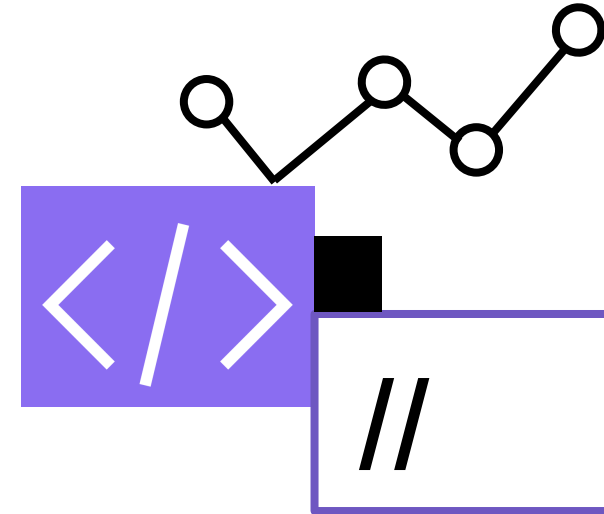
Para configurar o Spring Cloud Load Balancer, devemos adicionar a dependência `spring-cloud-starter-loadbalancer` ao nosso projeto. Com Maven, seria o seguinte:

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-loadbalancer</artifactId>
</dependency>
```



O balanceador obterá da Eureka as informações das instâncias em execução de cada microserviço. Para isso, devemos adicionar a anotação **@EnableDiscoveryClient** na classe principal do microserviço que irá realizar as requisições.

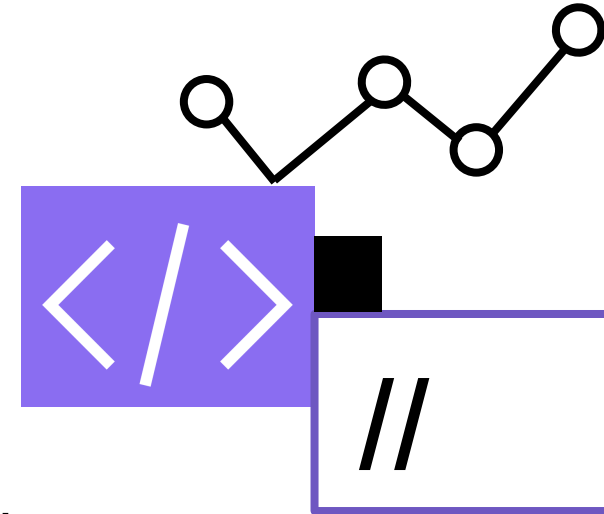
```
@SpringBootApplication
@EnableFeignClients
@EnableDiscoveryClient
public class Application {
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```



Uso de Spring Cloud Load Balancer com Feign

Ao adicionar as dependências da etapa anterior, o Feign usará o método **Round-robin** para balanceamento de carga por padrão, sem que seja necessário adicionar anotações ou configurações. Se quisermos configurar o balanceador de carga (por exemplo, alterar o algoritmo que seleciona as instâncias ou adicionar um novo), fazemos da seguinte forma:

```
public class CustomLoadBalancerConfiguration {  
  
    @Bean  
    ReactorLoadBalancer<ServiceInstance> randomLoadBalancer(Environment environment,  
        LoadBalancerClientFactory loadBalancerClientFactory) {  
        String name = environment.getProperty(LoadBalancerClientFactory.PROPERTY_NAME);  
        return new RandomLoadBalancer(loadBalancerClientFactory  
            .getLazyProvider(name, ServiceInstanceListSupplier.class),  
            name);  
    }  
}
```



Aqui estamos criando um método para usar o algoritmo de seleção aleatória. Uma vez que o método é criado e registrado com **@Bean**, temos que ir para a interface que criamos anteriormente e adicionar a anotação **@LoadBalancerClient** para fazer o algoritmo mudar.

```
@FeignClient(name = "product-service")
@LoadBalancerClient(name = "product-service",
configuration=CustomLoadBalancerConfiguration.class)
public interface ProductClient {
    @RequestMapping(method = RequestMethod.GET, value = "/products")
    List<ProductDTO> getProducts();
}
```

Muito obrigado!