## Spring Cloud Feign

Spring Cloud Feign是一套基于Netflix Feign实现的声明式服务调用客户端。它使得编写 Web服务客户端变得更加简单。我们只需要通过创建接口并用注解来配置它既可完成对Web服务接口的绑定。它具备可插拔的注解支持,包括Feign注解、JAX-RS注解。它也支持可插拔的编码器和解码器。Spring Cloud Feign还扩展了对Spring MVC注解的支持,同时还整合了 Ribbon和Eureka来提供均衡负载的HTTP客户端实现。

下面,我们通过一个例子来展现Feign如何方便的声明对eureka-client服务的定义和调用。 实战练习

下面的例子,我们将利用之前构建的eureka-server作为服务注册中心、eureka-client作为服务提供者作为基础。而基于Spring Cloud Ribbon实现的消费者,我们可以根据eureka-consumer实现的内容进行简单改在就能完成,具体步骤如下:

• 根据eureka-consumer复制一个服务消费者工程,命名为: eureka-consumerfeign。在pom.xml中增加下面的依赖:

• 修改应用主类。通过@EnableFeignClients注解开启扫描Spring Cloud Feign客户端的功能:

```
@EnableFeignClients
```

```
@EnableDiscoveryClient
@SpringBootApplication
public class EurekaConsumerFeignApplication {
    public static void main(String[] args) {
        new
SpringApplicationBuilder(EurekaConsumerFeignApplication.class).web(true).run(args);
    }
}
```

• 创建一个Feign的客户端接口定义。使用@FeignClient注解来指定这个接口所要调用的服务名称,接口中定义的各个函数使用Spring MVC的注解就可以来绑定服务提供方的REST接口,比如下面就是绑定eureka-client服务的/dc接口的例子:

package lizzy. springcloud. service;

```
import org. springframework. cloud. netflix. feign. FeignClient;
import org. springframework. web. bind. annotation. GetMapping;
@FeignClient("eureka-client")
public interface DcClient {
    @GetMapping("/dc")
```

通过Spring Cloud Feign来实现服务调用的方式更加简单了,通过@FeignClient定义的接口来统一的生命我们需要依赖的微服务接口。而在具体使用的时候就跟调用本地方法一点的进行调用即可。由于Feign是基于Ribbon实现的,所以它自带了客户端负载均衡功能,也可以通过Ribbon的IRule进行策略扩展。另外,Feign还整合的Hystrix来实现服务的容错保护,在Dalston版本中,Feign的Hystrix默认是关闭的。待后文介绍Hystrix带领大家入门之后,我们再结合介绍Feign中的Hystrix以及配置方式。

在完成了上面你的代码编写之后,读者可以将eureka-server、eureka-client、eureka-consumer-feign都启动起来,然后访问http://localhost:2103/consumer ,来跟踪观察 eureka-consumer-feign服务是如何消费eureka-client服务的/dc接口的,并且也可以通过启动多个eureka-client服务来观察其负载均衡的效果。

## 具体工程说明如下:

- eureka的服务注册中心: eureka-server
- eureka的服务提供方: eureka-client
- eureka的服务消费者: eureka-consumer-feign