

# feign详解

更多学习资料，一点课堂：[www.yidiankt.com](http://www.yidiankt.com)

一点课堂QQ群：984370849，QQ：2868289889

微信号：chengweixin9

免费公开课：<https://ke.qq.com/course/394307>

## Feign简介

Feign是Netflix开发的声明式、模板化的HTTP客户端，Feign可以帮助我们更快捷、优雅地调用HTTP API。

Feign是一个声明式的伪Http客户端，它使得写Http客户端变得更简单。使用Feign，只需要创建一个接口并注解。Feign默认集成了Ribbon，并和Eureka结合，默认实现了负载均衡的效果。

简而言之：

Feign 采用的是基于接口的注解

## RestTemplate和feign区别

使用 RestTemplate时，URL参数是以编程方式构造的，数据被发送到其他服务。

Feign是Spring Cloud Netflix库，用于在基于REST的服务调用上提供更高级别的抽象。Spring Cloud Feign在声明性原则上工作。使用Feign时，我们在客户端编写声明式REST服务接口，并使用这些接口来编写客户端程序。开发人员不用担心这个接口的实现。

再说，就是简化了编写，RestTemplate还需要写上服务器IP这些信息等等，而FeignClient则不用。

## Feign使用

github Feign：<https://github.com/OpenFeign/feign>

- 添加pom文件

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

- 在启动类上，添加开feign注解

```
@SpringBootApplication
@EnableEurekaClient
```

```

@EnableFeignClients
public class OrderApp {

    public static void main(String[] args) {

        SpringApplication.run(OrderApp.class, args);

    }

    @Bean
    @LoadBalanced
    RestTemplate restTemplate() {
        return new RestTemplate();
    }
}

```

- 定义接口

```

@FeignClient(value = "yidiankt-user")
public interface FeignService {

    @RequestMapping(value = "/user/{id}", method = RequestMethod.GET)
    String getUser(@PathVariable("id") int id);

}

```

- 使用feign

```

@RestController
public class OrderController {

    @Autowired
    FeignService feignService;

    @RequestMapping("/order2")
    public String addOrder2(String name, int id) {
        // 调用用户, 查询用户信息,
        String result = feignService.getUser(id);

        return "商品: " + name + ", 生成订单: " + result;
    }
}

```