

东软睿道内部公开

文件编号: D000-

Spring Cloud微服务架构

版本: 1.0.0

第5章 Feign负载均衡

东软睿道教育信息技术有限公司
(版权所有, 翻版必究)

Copyright © Neusoft Educational Information Technology Co., Ltd
All Rights Reserved



本章教学目标

- ✓ 了解Feign简介；
- ✓ 了解Feign用途；
- ✓ 掌握Feign接口开发；
- ✓ 掌握Feign服务消费者编写；



本章教学内容

| 节 | 知识点 | 掌握程度 | 难易程度 | 教学形式 | 对应在线微课 |
|--------------------|--------------|------|------|------|--------|
| Feign简介 | Feign简介 | 了解 | | 线下 | |
| | Feign用途 | 了解 | | 线下 | |
| 使用Feign实现声明式REST调用 | 开发Feign接口 | 掌握 | 难 | 线下 | |
| | 编写Feign服务消费者 | 掌握 | 难 | 线下 | |
| | 测试 | 掌握 | | 线下 | |

CONTENTS

目录

01

Feign简介

02

使用Feign实现负载均衡

Feign是什么

- ❖ 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。
- ❖ 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。
- ❖ 总之，Feign是基于Ribbon实现的，Feign使得接口更加规范化，更有阅读性并且更加简单化。
- ❖ Feign项目在github上托管：<https://github.com/OpenFeign/feign>

Feign用途

- ❖ Feign旨在使编写Java Http客户端变得更容易。
- ❖ 前面在使用Ribbon+RestTemplate时，利用RestTemplate对http请求的封装处理，形成了一套模版化的调用方法。但是在实际开发中，由于对服务依赖的调用可能不止一处，往往一个接口会被多处调用，所以通常都会针对每个微服务自行封装一些客户端类来包装这些依赖服务的调用。
- ❖ 所以，Feign在RestTemplate的基础上对其封装，由它来帮助我们定义和实现依赖服务接口的定义。
- ❖ 简化了使用Spring cloud Ribbon时，自动封装服务调用客户端的开发量。
- ❖ Feign集成了Ribbon
 - ▶ 利用Ribbon维护了mscloud-user的服务列表信息，并且通过轮询实现了客户端的负载均衡。
 - ▶ 而与Ribbon不同的是，通过feign只需要定义服务绑定接口且以声明式的方法，优雅而简单的实现了服务调用。

CONTENTS

目录

01

Feign简介

02

使用Feign实现负载均衡

开发Feign接口

❖ 修改mscloud-api工程

- ▶ 修改POM
- ▶ 新建UserClientService接口并新增注解@FeignClient
- ▶ mvn clean install

- ▶ 全部代码参见：[ch05-01-feign负载均衡/mscloud-api](#)

开发Feign接口

- ❖ 修改POM
 - ▶ 添加

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

开发Feign接口

- ❖ 使用Feign与Ribbon不同的是需要调用前实现规范化接口，使用@FeignClient和SpringMVC注解来绑定服务提供方的REST
- ❖ 新建UserClientService接口并新增注解@FeignClient

```
@FeignClient("eureka-client-user")
public interface UserClientService
{
    @RequestMapping(value = "/user/findAll", method = RequestMethod.GET)
    public List<User> findAll();

    @RequestMapping(value = "/user/findUserId/{id}", method = RequestMethod.GET)
    public User findUserId(@PathVariable("id") int id);
}
```

开发Feign接口

❖ mvn clean install

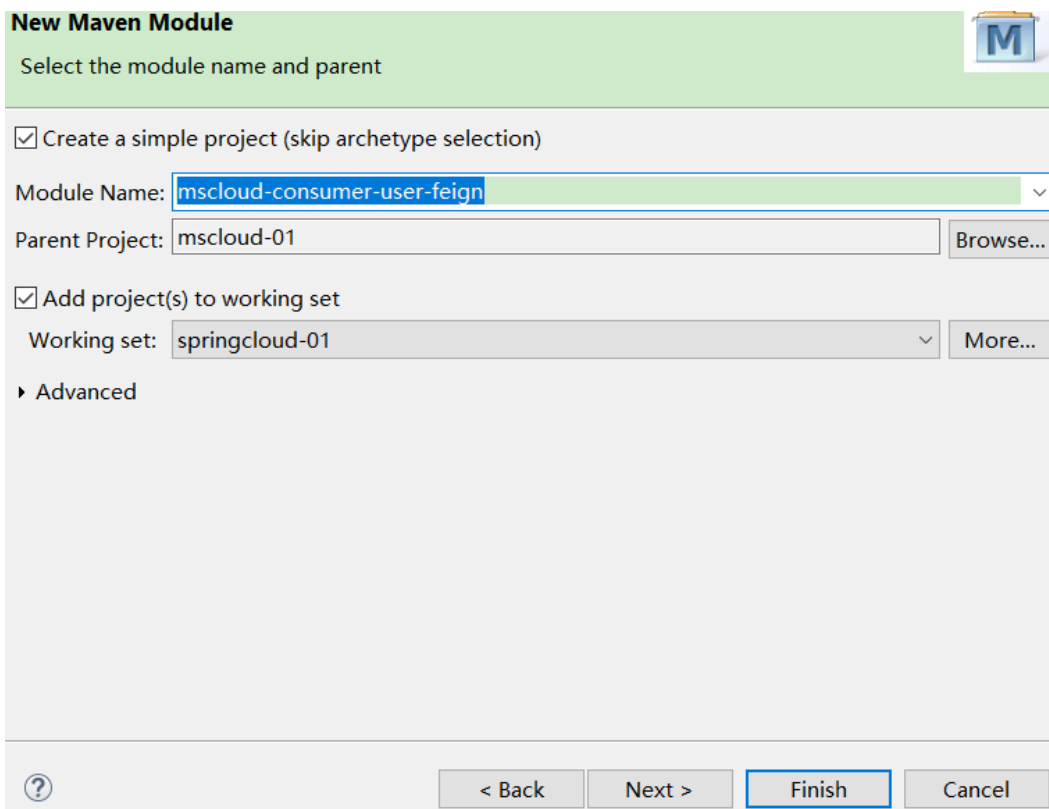
创建Feign服务消费者

❖ 实现步骤

- ▶ 新建Module模块mscloud-consumer-user-feign
 - ★ 参考mscloud-consumer-user-ribbon-80, 进行修改
 - ▶ 修改pom.xml
 - ▶ 修改Controller
 - ▶ 修改主启动类
-
- ▶ 全部代码参见: [ch05-01-feign负载均衡/ mscloud-consumer-user-feign](#)

创建Feign服务消费者

- ❖ 新建Module模块mscloud-consumer-user-feign
 - ▶ 参考mscloud-consumer-user-ribbon-80实现
 - ▶ 修改主启动类名



New Maven Module

Select the module name and parent

☒ Create a simple project (skip archetype selection)

Module Name:

Parent Project: [Browse...](#)

☒ Add project(s) to working set

Working set: [More...](#)

▶ Advanced

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

创建Feign服务消费者

- ❖ 修改pom.xml
 - ▶ 添加对feign的支持

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

创建Feign服务消费者

❖ 修改Controller

- ▶ 添加上一步新建的UserClientService接口

```
@RestController
public class UserController_Consumer {

    @Autowired
    private UserClientService service;

    @RequestMapping("/consumer/user/findAll")
    public List<User> findAll(){
        return service.findAll();
    }

    @RequestMapping("/consumer/user/findUserById/{id}")
    public User findUserById(@PathVariable("id") int id){

        return service.findUserById(id);
    }
}
```

创建Feign服务消费者

❖ 修改主启动类

▶ 添加@EnableFeignClients

```
@SpringBootApplication
@EnableEurekaClient
@EnableFeignClients
public class UserConsumer80_Feign_App {
    public static void main(String[] args)
    {
        SpringApplication.run(UserConsumer80_Feign_App.class, args);
    }
}
```


测试

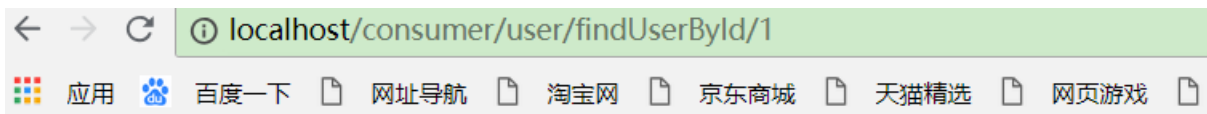
❖ 步骤

- ▶ 启动3个Eureka集群
- ▶ 启动3个用户微服务8001/8002/8003
- ▶ 启动Feign服务消费者
- ▶ 浏览器

★ <http://localhost/consumer/user/findUserById/1>

★ 多次输入同一个网址，随机访问不同的微服务

```
1 EurekaServer7001_App [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (2018年9月21日 下午5:15:22)
2 EurekaServer7002_App [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (2018年9月21日 下午5:15:38)
3 EurekaServer7003_App [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (2018年9月21日 下午5:15:55)
4 UserProvider8001_App [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (2018年9月21日 下午5:19:49)
5 UserProvider8002_App (1) [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (2018年9月21日 下午5:20:20)
6 UserProvider8003_App (2) [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (2018年9月21日 下午5:21:06)
7 UserConsumer80_Feign_App [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (2018年9月21日 下午5:22:17)
```



```
{"id":1,"loginName":"user111","username":"张三","password":"123456","dbSource":"mybatis2"}
```



```
{"id":1,"loginName":"user111","username":"张三","password":"123456","dbSource":"mybatis3"}
```

测试

❖ 小结

- ▶ Feign通过接口的方法调用Rest服务（之前是Ribbon+RestTemplate），
- ▶ 该请求发送给Eureka服务器（<http://eureka-client-user/user/findUserById/1>），通过Feign直接找到服务接口，由于在进行服务调用的时候融合了Ribbon技术，所以也支持负载均衡作用。

本章重点总结

- ❖ 了解Feign简介；
- ❖ 了解Feign用途；
- ❖ 掌握Feign接口开发；
- ❖ 掌握Feign服务消费者编写；

课后作业【必做任务】

❖ 独立完成课件中的示例



课后作业【线上任务】

❖ 线上任务

- ▶ 安排学员线上学习任务（安排学员到睿道实训平台进行复习和预习的任务，主要是进行微课的学习）

