

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

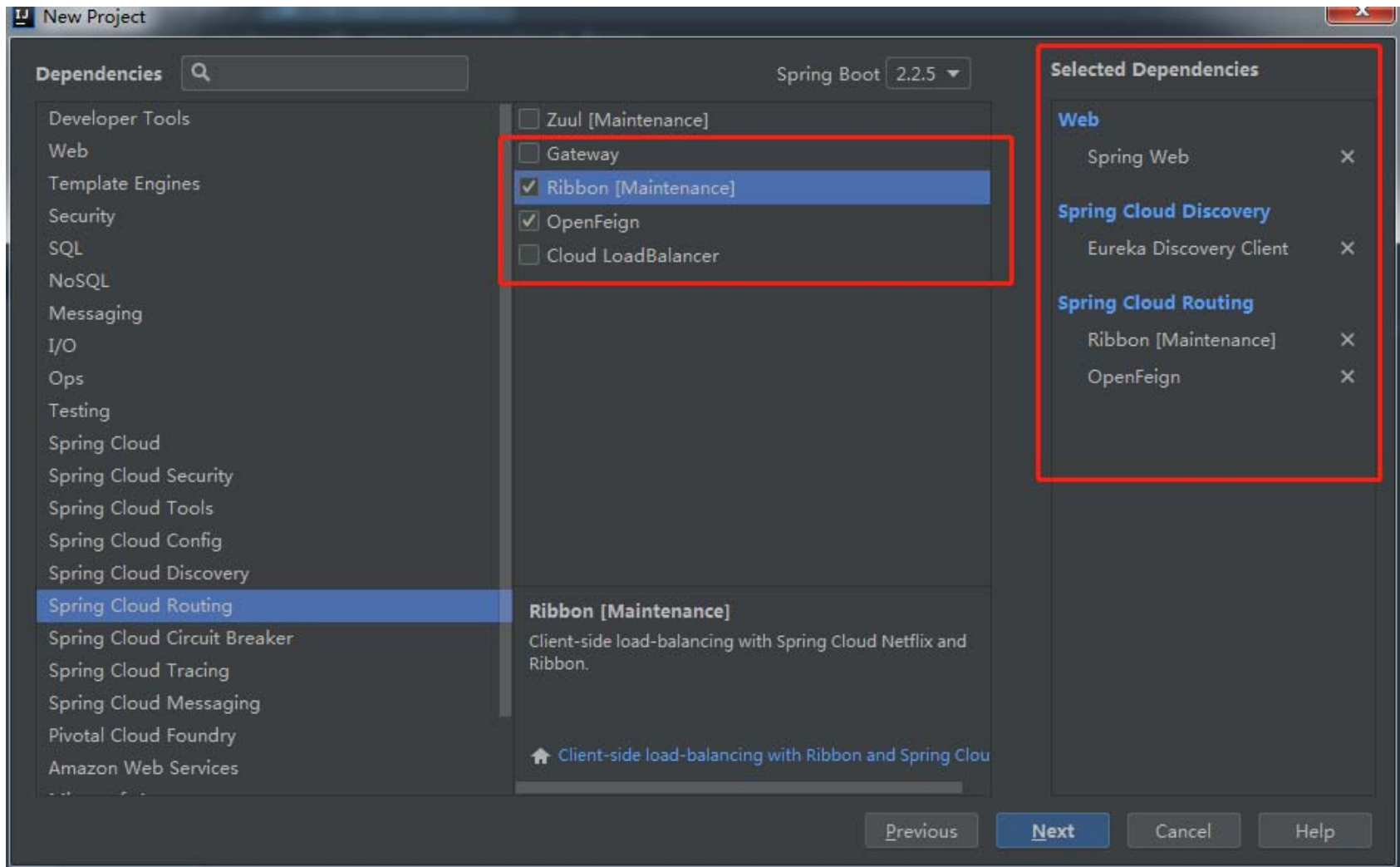
使用Feign实现负载均衡

使用Feign实现负载均衡

❖ 新建Feign工程

- ▶ 新建工程
- ▶ 修改全局配置文件
- ▶ 复用之前工程的bean及config包代码
- ▶ 在Service包下新建接口并新增注解@FeignClient
- ▶ 编写Controller
- ▶ 修改主启动类
- ▶ 测试

新建Feign工程



修改全局配置文件

❖ 修改application.yml

server:

port: 80

spring:

application:

name: feign

eureka:

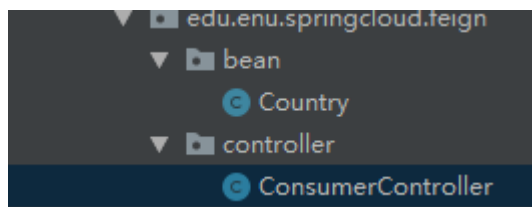
client:

service-url:

defaultZone: http://eureka.com:7001/eureka/

register-with-eureka: false

复用之前工程的bean及controller包代码



在Service包下新建接口

```
@Service
@FeignClient("provider")
public interface CountryService {

    @RequestMapping("/country/findAll")
    List<Country> findAll();

    @RequestMapping("/country/find/{code}")
    Country findByCode(@PathVariable String code);
}
```

编写Controller

```
@RestController
public class ConsumerController {
    @Autowired
    private CountryService countryService;

    @RequestMapping("/consumer/findAll")
    public List<Country> findAll() {
        return countryService.findAll();
    }

    @RequestMapping("/consumer/find/{code}")
    public Country find(@PathVariable String code) {
        return countryService.findByCode(code);
    }
}
```

修改主启动类

@SpringBootApplication

@EnableFeignClients // 允许调用其他微服务

@EnableEurekaClient

测试

❖ 步骤

- ▶ 启动3个Eureka集群（第三章Eureka工程）
- ▶ 启动3个生产者微服务（第三章Eureka工程）
- ▶ 启动Feign服务消费者
- ▶ 浏览器
 - ★ <http://localhost/consumer/find/CHN>
 - ★ 多次输入同一个网址，轮询访问不同的微服务

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，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

