# 微服务之间相互调用

### 导入ribbon依赖

|  |
| --- |
| *<!--ribbon-->* ***<*dependency*>  <*groupId*>***org.springframework.cloud***</*groupId*>  <*artifactId*>***spring-cloud-starter-ribbon***</*artifactId*> </*dependency*>*** |

### 创建RestTemplate

|  |
| --- |
| @SpringBootApplication @EnableEurekaClient @MapperScan***(*"cn.shu.cart.mapper"*)* public class** StarterCart ***{* public static void** main***(***String***[]*** args***) {*** SpringApplication.*run****(***StarterCart.**class**,args***)***;  ***}*** @Bean  @LoadBalanced  **public** RestTemplate init***(){* return new** RestTemplate***()***;  ***} }*** |

### 注入RestTemplate

|  |
| --- |
| @Autowired **private** RestTemplate **restTemplate**=**null**; |

### 调用服务

|  |
| --- |
| String url=**"http://easymall-product/product/manage/item/"**+cart.getProductId***()***; Product forObject = **restTemplate**.getForObject***(***url, Product.**class*)***; |

# feign使用场景

之前微服务调用结构：

多个微服务调用同一服务使用ribbon，有重复代码。能否提取出来？

|  |
| --- |
|  |

在微服务框架中,对于多个微服务调用同一个微服务的功能,能否编写成更简单的形式,实现公用服务调用---feign

# feign介绍

也是一个服务调用的客户端,也可以实现负载均衡.和ribbon有关系.为了简化服务调用形式,springcloud封装了ribbon+restTemplate形成了一个新的组件feign. ribbon和feign完全可以相互代替使用.

组件底层:ribbon

公用服务调用:feign

# feign测试案例

### pom文件

#### 继承spring-boot

#### 依赖feign /eureka（抓取服务信息）

|  |
| --- |
| *<!--eureka 客户端-->* ***<*dependency*>  <*groupId*>***org.springframework.cloud***</*groupId*>  <*artifactId*>***spring-cloud-starter-eureka***</*artifactId*> </*dependency*>*** *<!--feign-->* ***<*dependency*>  <*groupId*>***org.springframework.cloud***</*groupId*>  <*artifactId*>***spring-cloud-starter-feign***</*artifactId*> </*dependency*>*** |

### 启动类

|  |
| --- |
| @SpringBootApplication @EnableEurekaClient @EnableFeignClients @MapperScan***(*"cn.shu.cart.mapper"*)* public class** StarterCart ***{* public static void** main***(***String***[]*** args***) {*** SpringApplication.*run****(***StarterCart.**class**,args***)***;  ***} }*** |

### 编写服务接口

该接口和easymall-product服务的controller接口一致，当调用该方法时，会自动调用easymall-product中的指定接口

|  |
| --- |
| *//调用easymall-product微服务* @FeignClient***(***name=**"easymall-product"*)* public interface** ProductService ***{*** *//单个商品查询* @RequestMapping***(*"/product/manage/item/{productId}"*)* public** Product queryOneProduct***(***@PathVariable String productId***)***; ***}*** |

easymall-product中的controller接口：

|  |
| --- |
| @RestController @RequestMapping***(*"/product/manage"*)* public class** ProductController ***{*** @Autowired  **private** ProductService **productService**;   *//单个商品查询* @RequestMapping***(*"/item/{productId}"*)* public** Product queryOneProduct***(***@PathVariable String productId***){* return productService**.queryOneProduct***(***productId***)***;  ***}   }*** |

### 使用

|  |
| --- |
| *//注入服务对象* @Autowired **private** ProductService **productService**=**null**;  *//使用*  Product forObject = **productService**.queryOneProduct***(***cart.getProductId***())***; |

# feign的实际应用

编写的服务接口和被调用者的controller一致，所以应该由被调用者来写这个接口，然后调用者添加依赖使用相关接口