

# Ribbon

---

## 学习目标

1. ribbon能做什么
2. 客户端负载均衡
3. ribbon的使用方式

## Ribbon

ribbon是一个基于HTTP和TCP的客户端负载均衡工具，是基于Netflix Ribbon实现。通过Spring Cloud的封装，可以让我们轻松地将面向服务的REST模版请求自动转换成客户端负载均衡的服务调用。

Spring Cloud Ribbon虽然只是一个工具类框架，它不像**服务注册中心、配置中心、API网关**那样需要独立部署，但是它几乎存在于**每一个Spring Cloud构建的微服务和基础设施中。**因为微服务间的调用，API网关的请求转发等内容，实际上都是通过Ribbon来实现的，包括后续我们将要介绍的Feign，它也是基于Ribbon实现的工具。所以，对Spring Cloud Ribbon的理解和使用，对于我们使用Spring Cloud来构建微服务非常重要。

## 1.Ribbon能做什么

在服务与服务之中，我们可以使用Ribbon来进行远程服务的调用，通过使用面向服务的 **REST模板中** 中的

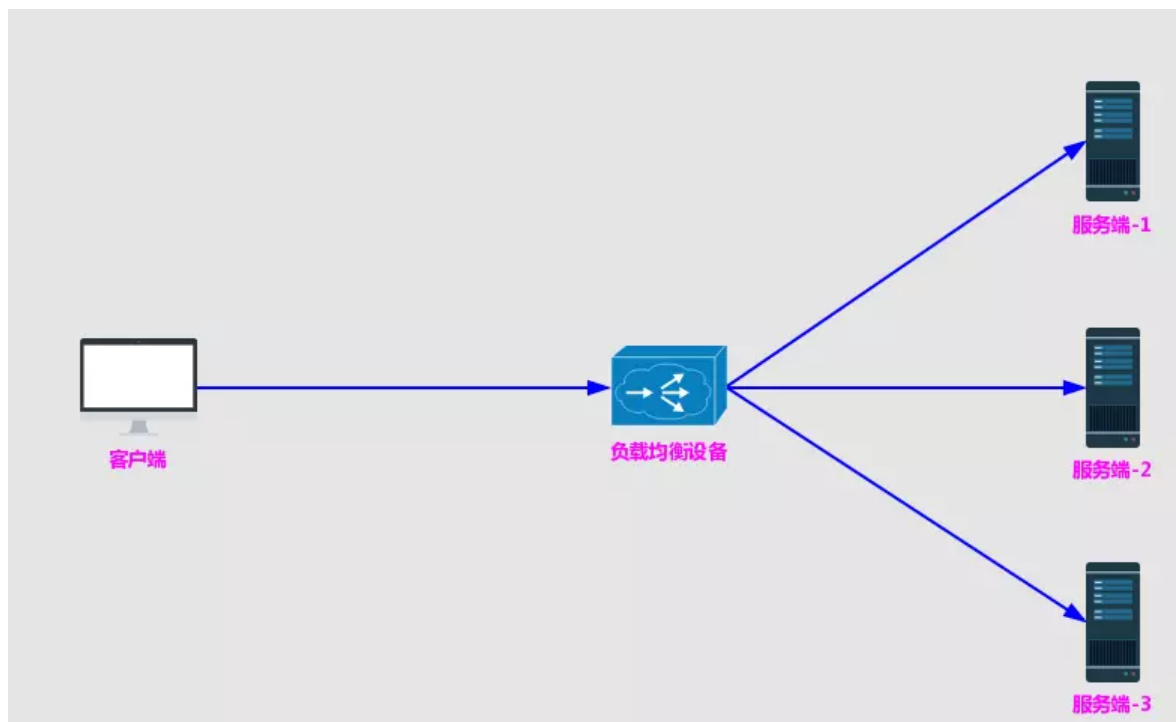
```
restTemplate.getForObject("http://PRODUCT-SERVER/api/findById?id=" + id, Product.class);
```

api来进行远程服务的调用，不过这样有一个缺点，就是端口号与返回值类型被写死了，所以后面我们会用**feign**的方式来进行远程服务的调用

## 2.客户端负载均衡

负载均衡在一个架构中，是一个**非常重要**并且不得不进行实施的内容，因为负载均衡是对系统的高可用，网络压力的缓解和处理能力扩容的重要手段之一，我们通常所说的负载均衡都是指，服务端的负载均衡，其中分为硬件负载均衡和软件负载均衡。硬件负载均衡主要通过服务器节点之间按照专门用于负载均衡的设备，比如F5等；而软件负载均衡则是通过在服务器上安装一些用于负载均衡功能或模块等软件来完成请求分发工作，比如Nginx等。

不论采用硬件负载均衡还是软件负载均衡，只要是服务端都能以类似下图的架构方式构建起来：



通过Spring Cloud Ribbon的封装，我们在微服务架构中使用客户端负载均衡调用非常简单，只需要如下两步：

- 服务提供者只需要启动多个服务实例并注册到一个注册中心或是多个相关联的服务注册中心。
- 服务消费者直接通过调用被 `@LoadBalanced` 注解修饰过的 `RestTemplate` 来实现面向服务的接口调用。

这样，我们就可以将服务提供者的高可用以及服务消费者的负载均衡调用一起实现了。

### SpringCloud中默认的Ribbon负载均衡选择器

默认SpringCloud中Ribbon负载均衡是通过**轮询**的方式来进行请求的转发，不过我们可以通过在客户端以配置文件的方式配置负责均衡的转发策略

```
1 #将负载均衡以随机的方式进行请求分配
2 PRODUCT-SERVER:
3   ribbon:
4     NFLoadBalancerRuleClassName: com.netflix.loadbalancer.RandomRule
```

在轮询中，如果遇到服务挂掉了，那么负载均衡将会尝试3次的请求分配，如果3次过后，服务端还是没有反应，就不会再访问那个了，但是过一段时间之后，会尝试的发送一次 请求，检测服务有没有挂掉

[详情](#)

[详情](#)