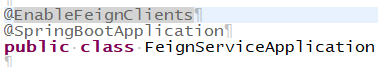
# Spring Cloud Feign使用笔记

Spring Cloud各组件基于HTTP协议进行交互。

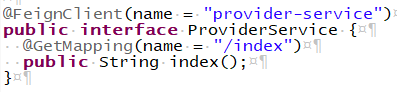
结合RestTemplate、Ribbon、Hystrix可以实现接口的调用，即Feign。

1. 主应用类添加@EnableFeignClients，开启Feign。

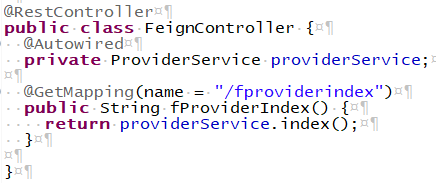


1. 假设提供者服务名“provider-service”，有GET接口index。

创建接口ProviderService，使用@FeignClient指定对应的Service，通过Spring MVC的注解配置对应的接口。

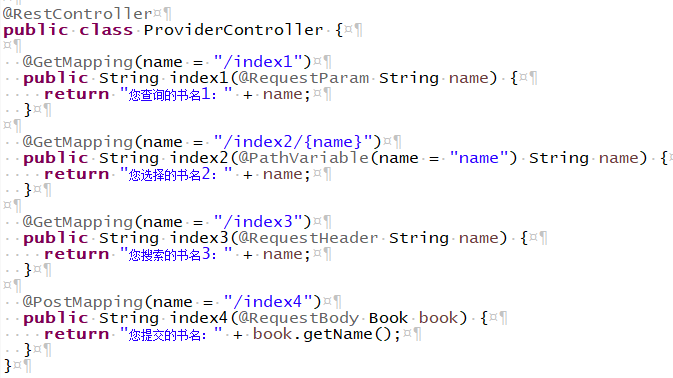


1. 可以像以前使用Service的方法一样，使用ProviderService。具体的接口调用，由Feign处理。本例使用Controller调用。



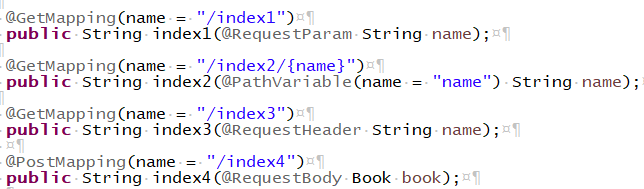
根据地址<http://127.0.0.1:8080/fproviderindex>，即可调用provider-service的index接口。

1. 多种传参方式，改造Provider的接口如下：



基本包含了常用的传参形式。

Feign的Service做对应的修改：

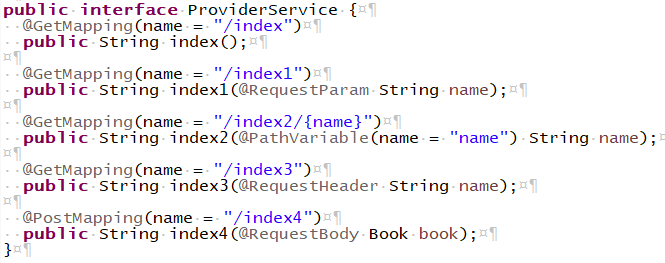


通过以上的代码，我们可以看出，其实Feign的Service接口内的方法，其实就是提供者Controller里的方法，包括用来传递参数的实体类。

如果在初期，Provider的方法不多，写Service的时候，完全可以把提供者的Controller复制下来，去掉实现的代码，再把用到的实体类复制到Feign工程里。

但是一旦Provider的实体类、Controller、方法比较多，这么做起来就比较苦手了，同样的代码，Provider开发人员写一遍，Feign开发人员复制一遍（或者反过来）。

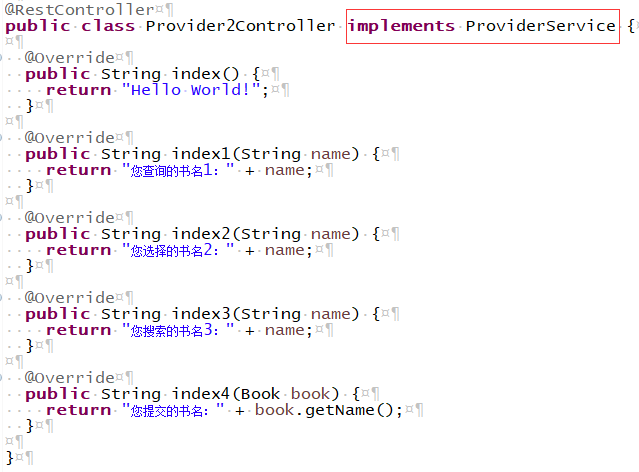
我们可以将Service接口、传参用的实体类全部提取出一个工程，比如provider-api，然后Feign和Provider分别引入本工程，继承或实现接口。



将前文缩写的ProviderService接口复制到provider-api工程中，原来注解的@FeignClient(name = "provider-service")去掉。相关实体类也复制到工程中。

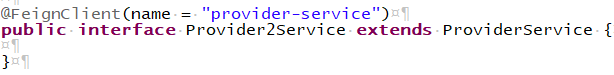
Provider、Feign分别引入provider-api工程，具体配置不再赘述。

改造Provider的Controller：



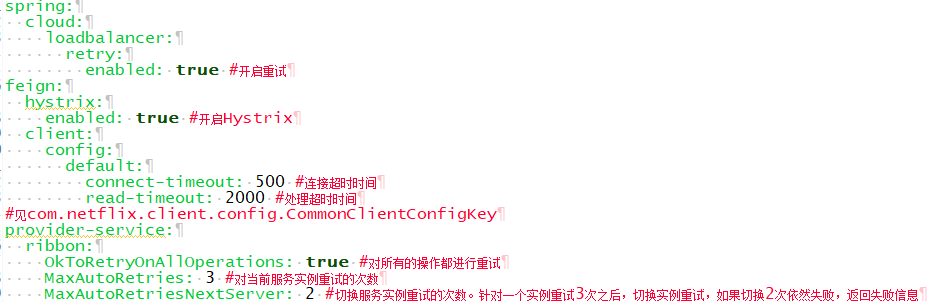
Controller实现了ProviderService接口，因为相关的注解都在Service接口里，所以Controller不用再配置。

改造Feign的Service：

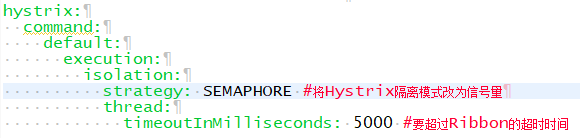


继承之后，用@FeignClient注解即可。

1. Ribbon的配置



1. Hystrix的配置

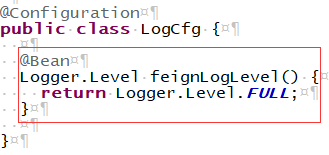


1. Log的配置



这里是指定ProviderService发起接口调用时，Log日志级别为Debug。

仅此还是无法输出日志的，因为Feign自己的Log级别是NONE，即“No logging”，要想看日志，还要做如下配置：

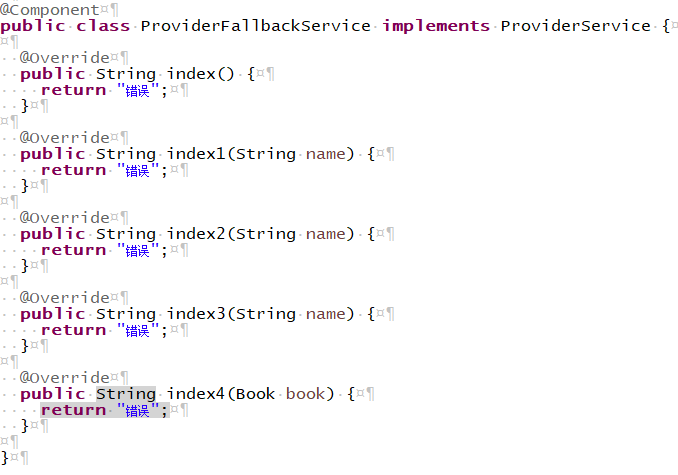




在指定的Feign接口上配置即可。

也可以将红框中的内容放在主应用类中，这就是全局的Log配置了。

1. 服务降级



新建实现Feign接口Service的具体类，重写方法实现降级。



在指定的Feign接口上配置即可。