**史上最简单的SpringCloud教程 | 第三篇: 服务消费者（Feign）**

2017年04月09日 11:53:43

阅读数：305743

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/69808079>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp" \t "_blank)

* 最新版本：
* [史上最简单的SpringCloud教程 | 第三篇: 服务消费者（Feign）(Finchley版本)](http://blog.csdn.net/forezp/article/details/81040965)

上一篇文章，讲述了如何通过RestTemplate+Ribbon去消费服务，这篇文章主要讲述如何通过Feign去消费服务。

**一、Feign简介**

Feign是一个声明式的伪Http客户端，它使得写Http客户端变得更简单。使用Feign，只需要创建一个接口并注解。它具有可插拔的注解特性，可使用Feign 注解和JAX-RS注解。Feign支持可插拔的编码器和解码器。Feign默认集成了Ribbon，并和Eureka结合，默认实现了负载均衡的效果。

简而言之：

* Feign 采用的是基于接口的注解
* Feign 整合了ribbon

**二、准备工作**

继续用上一节的工程， 启动eureka-server，端口为8761; 启动service-hi 两次，端口分别为8762 、8773.

**三、创建一个feign的服务**

新建一个spring-boot工程，取名为serice-feign，在它的pom文件引入Feign的起步依赖spring-cloud-starter-feign、Eureka的起步依赖spring-cloud-starter-eureka、Web的起步依赖spring-boot-starter-web，代码如下：

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.forezp</groupId>

<artifactId>service-feign</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>service-feign</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.5.2.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-eureka</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-feign</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>Dalston.RC1</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

<repositories>

<repository>

<id>spring-milestones</id>

<name>Spring Milestones</name>

<url>https://repo.spring.io/milestone</url>

<snapshots>

<enabled>false</enabled>

</snapshots>

</repository>

</repositories>

</project>

在工程的配置文件application.yml文件，指定程序名为service-feign，端口号为8765，服务注册地址为[http://localhost:8761/eureka/](http://localhost:8761/eureka/" \t "_blank) ，代码如下：

eureka:

client:

serviceUrl:

defaultZone: http://localhost:8761/eureka/

server:

port: 8765

spring:

application:

name: service-feign

在程序的启动类ServiceFeignApplication ，加上@EnableFeignClients注解开启Feign的功能：

@SpringBootApplication

@EnableDiscoveryClient

@EnableFeignClients

public class ServiceFeignApplication {

public static void main(String[] args) {

SpringApplication.run(ServiceFeignApplication.class, args);

}

}

定义一个feign接口，通过@ FeignClient（“服务名”），来指定调用哪个服务。比如在代码中调用了service-hi服务的“/hi”接口，代码如下：

/\*\*

\* Created by fangzhipeng on 2017/4/6.

\*/

@FeignClient(value = "service-hi")

public interface SchedualServiceHi {

@RequestMapping(value = "/hi",method = RequestMethod.GET)

String sayHiFromClientOne(@RequestParam(value = "name") String name);

}

在Web层的controller层，对外暴露一个”/hi”的API接口，通过上面定义的Feign客户端SchedualServiceHi 来消费服务。代码如下：

@RestController

public class HiController {

@Autowired

SchedualServiceHi schedualServiceHi;

@RequestMapping(value = "/hi",method = RequestMethod.GET)

public String sayHi(@RequestParam String name){

return schedualServiceHi.sayHiFromClientOne(name);

}

}

启动程序，多次访问[http://localhost:8765/hi?name=forezp](http://localhost:8765/hi?name=forezp" \t "_blank),浏览器交替显示：

hi forezp,i am from port:8762

hi forezp,i am from port:8763

Feign源码解析：<http://blog.csdn.net/forezp/article/details/73480304>

本文源码下载：   
<https://github.com/forezp/SpringCloudLearning/tree/master/chapter3>