Spring-Cloud 系列第 3 篇：spring-cloud-eureka-consumer

作者：xjtushilei
原文链接：https://ld246.com/article/1497579337048
来源网站：链滴

<blockquote>
自学 spring-cloud 系列，越来越感觉 spring-cloud 很强大！
主要分为以下几篇：

spring-cloud-config: 分布式配置管理
spring-cloud-eureka: 服务注册与发现
spring-cloud-eureka-consumer: 远程服务调用和及其负载均衡
spring-cloud-Hystrix: 熔断器保证服务高可用
spring-cloud-config-eureka-ribbon: 分布式配置管理的高可用
pring-cloud-bus: 配置信息的实时更新
<a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.xjtushilei.com%2F201%2F06%2F14%2Fspring-cloud-series-spring-cloud-zuul" target="_blank" rel="nofollow ugc"spring-cloud-zuul: 网关基础服务

</blockquote>
<h2 id="介绍">介绍</h2>
spring-cloud-eureka，更加具体的内容，这里将会介绍远程服务调用和及其负载均衡。
我们将我们的服务注册在我们的服务中心里，那么如何去调用这些服务呢？我们可以用使用远程务调用来解决，顺带还有方便的负载均衡功能。
<h2 id="如何使用">如何使用</h2>

创建服务中心
注册几个被调用服务
注册一个 consumer
测试 consumer 与负载均衡

<h2 id="1-创建服务中心">1. 创建服务中心</h2>
上一篇文章，我们已经学会了使用单机或者集群的方式来创建服务中心，这里我们使用简单的单方式来创建！
在 <code>spring-cloud-eureka-server</code> 里启动采用这个 profile 文件：
<pre><code class="highlight-chroma">server:
  port: 8761

spring:
  application:
    name: eureka-s rver

eureka:
  instance:
    lease-expiration

duration-in-seconds: 6
  lease-renewal-interval-in-seconds: 2
  client:
  service-url:
  defaultZone: http://localhost:8761/eureka/

</code></pre>
这个是单机版的。并将自己注册到了服务里。
<h2 id="2-注册几个被调用服务">2. 注册几个被调用服务</h2>
我们启动了这三个 profile 文件。
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile com/e/dc530676d1e64dd894c6d57ce20504f4.png?imageView2/2/interlace/1/format/jpg">
配置文件中，<code>server.port</code> 分别是 8083,8084,8085，其他参数完全一致！同，我们在 controller 中设置了这样一个 rest 服务。
<pre><code class="highlight-chroma"> @Value("${server.port}")
  String port;

  @RequestMapping("/hi")
  public String hi(){
  return port+"端口为您服务！";
  }

</code></pre>
这样方便知道我们具体调用了哪个服务。
<h2 id="3-注册一个consumer">3. 注册一个 consumer</h2>
需要额外的依赖，使用了 feign 来进行远程调用。
pom.xml：
<pre><code class="highlight-chroma"><!--远程调用-->
  <dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-feign</artifactId>
  </dependency>

</code></pre>
创建一个接口：
<pre><code class="highlight-chroma">@Component
@FeignClient(name = "eureka-client-1")
public interface HelloRemoteInterface {

    @RequestMapp
ng(value = "/hi")
    public String hi(
;
}

</code></pre>
这里的 name 就是你的那个服务的 application.name。根据名字来调用，才能实现负载均衡嘛

使用接口，创建一个测试的 controller：
<pre><code class="highlight-chroma"><span class="highlight
cl">@RestController
public class Cons
merController {

    @Autowired
    HelloRemoteInt
rface helloRemoteInterface;

    @RequestMapp
ng("/hello")
    public String hel
o() {
        return helloR
moteInterface.hi();
    }
}

</code></pre>
同时，将我们的这个服务也注册到服务中心。
配置文件：
<pre><code class="highlight-chroma"><span class="highlight
cl">server:
  port: 8086

spring:
  application:
    name: eureka-c
nsumer

eureka:
  client:
    service-url:
      defaultZone: h
tp://localhost:8761/eureka/

</code></pre>
启动方法：
<pre><code class="highlight-chroma"><span class="highlight
cl">@SpringBootApplication
@EnableEurekaCli
nt
@EnableFeignClie
ts

public class SprinCloudEurekaConsumerApplication {

    public static vod main(String[] args) {
        SpringApplication.run(SpringCloudEurekaConsumerApplication.class, args);
    }
}

</code></pre>
<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile com/e/e200f56350a5408ab5d8d2d09ccd4149.png?imageView2/2/interlace/1/format/jpg">
到这里，我们已经启动了好多个服务，如上图所示，在 IDEA 中，采用不同的 profile 来启动的式，有一个单机的 server，三个普通的服务（端口号不一样），还有我们的消费服务，下一节，你可从截图中看的更明白。
<h2 id="4-测试consumer与负载均衡">4. 测试 consumer 与负载均衡</h2>
到上一步为止，我们已经可以看到在 Eureka 的 dashboard 中已经有了好多个服务，如下图，要包括：

一个服务注册 server
一个消费者，用来进行远程调用
三个普通的 client（其端口不一样，来模拟分布式）

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile com/e/66626d1255c34fe79994b492633864b6.png?imageView2/2/interlace/1/format/jpg">
这时候，我们调用我们的 consumer 服务，浏览器里输入 <code>http://localhost:8086/hello /code>
得到的结果是不一样的，一共有三个：

8083 端口为您服务！
8084 端口为您服务！
8085 端口为您服务！

正好就是我们想要的结果。
不断的进行测试下去会发现 3 种结果交替出现，说明服务中心自动提供了服务均衡负载的功能。果我们将服务提供者的数量在提高为 N 个，测试结果一样，请求会自动轮询到每个服务端来处理。</ >
<h2 id="示例源码">示例源码</h2>
所有源码在我的 github 仓库里，传送门：https://github.com/xjtushilei/spring-cloud-simples.git
<h2 id="支持">支持</h2>
如果你喜欢~ 给个星