一：将提供者注册到Eureka中

1：pom.xml

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

2:因为原来的项目是个单纯的boot项目，里面没有直接引入cloud内容，手工给他加载的依赖环境。

<spring-cloud.version>Hoxton.RC2</spring-cloud.version>

<dependencyManagement>

<dependencies>

<dependency>

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

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

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<repositories>

<repository>

<id>spring-milestones</id>

<name>Spring Milestones</name>

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

</repository>

</repositories>

3：yml的书写

server:

port: 9000

spring:

datasource:

driver-class-name: com.mysql.jdbc.Driver

url: jdbc:mysql://localhost:3306/ybls?characterEncoding=UTF-8

username: root

password: ilovelaohan

application:

name: provider

eureka:

client:

register-with-eureka: true

fetch-registry: true

service-url:

defaultZone: http://localhost:9999/eureka/

instance:

prefer-ip-address: true

4：找到主启动类

@EnableDiscoveryClient

@EnableEurekaClient

二：将消费者注册到Eureka中

1：pom.xml

需要哪个两个包。

2：yml里面进行配置

Eureka的注册地址是哪里

3：启动类上加注解

在Spring Cloud中，默认使用ribbon，来完成消费的集群和负载均衡的功能。

注册中心在指向提供者的时候，还需要完成集群和负载均衡的功能。

1：pom.xml 加入ribbon的包。

<dependency>

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

<artifactId>spring-cloud-starter-netflix-ribbon</artifactId>

</dependency>

2：在你的RestTemplate的类上加注解

@LoadBalanced

三：Eureka的自我保护机制

初衷：

所有的提供者，消费者都要注册到Eureka中，而且协议是Http协议。保持通讯关系，采用的机制是心跳机制。

在生产中，有可能有io堵塞，有可能有网络堵塞，有可能有断网。导致心跳不及时。但其实这个服务还活着。

检测到这种情况，Eureka没又立刻从注册中西移除这个服务。知道达到超时的最大上限，才真的从注册中心中进行移除。

Kill -9

Kill id

Kill -15