**Springcloud+eureka**

springboot版本：2.1.3.RELEASE

springcloud整合Eureka

最终效果：

User-1000模块调用order-8001和order-8002微服务模块，调用策略为轮询调用。

# 注意事项

1，同一组微服务项目，只有端口不一样，例如order-8001和order-8002，只有端口不一样，其他代码都是一样的，即：spring.application.name:xxxxxxxx-name同一组必须是一样的。

2，调用策略由注解 @LoadBalanced实现



3，注册中心采用Eureka，Eureka项目本身也是一个springboot项目。

# Eureka项目

## 1，建立项目方式，新建一个springboot项目即刻。

## 2，添加依赖

|  |
| --- |
| *<?***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**>  <**parent**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-parent</**artifactId**>  <**version**>2.1.3.RELEASE</**version**>  <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**>  <**groupId**>com.zengziqiang</**groupId**>  <**artifactId**>eureka</**artifactId**>  <**version**>0.0.1-SNAPSHOT</**version**>  <**name**>eureka</**name**>  <**description**>Demo project for Spring Boot</**description**>   <**properties**>  <**java.version**>1.8</**java.version**>  </**properties**>   <**dependencies**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-web</**artifactId**>  </**dependency**>  *<!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-eureka-server -->* <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-eureka-server</**artifactId**>  <**version**>1.4.4.RELEASE</**version**>  </**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**>Greenwich.RELEASE</**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**>  </**project**> |

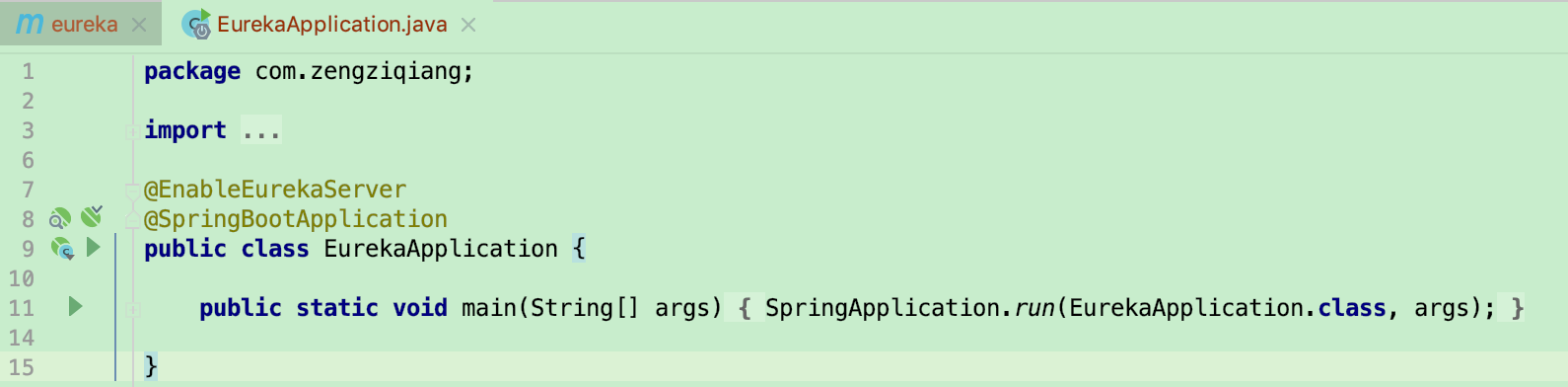
两个重要的依赖：

1. Eureka服务端依赖。
2. 通用依赖，注意版本



## 3，注解配置

在项目启动类上添加注解： @EnableEurekaServer



## 4，配置文件配置



主要是配置端口，主机名，等等。

## 5，启动项目

<http://localhost:8000/>

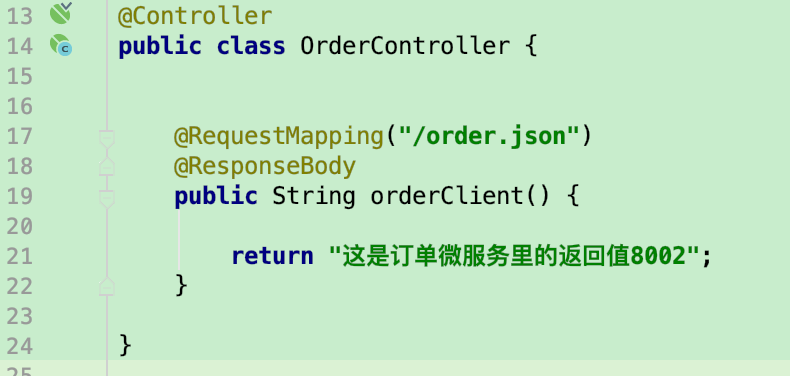
输入以上地址，出现Eureka管理界面即表示成功。

# order项目

order项目用于做集群的项目，新建一个项目即可，后续更改不同的端口Port后进行打包发布即可实现微服务集群。

创建一个简单的springboot项目即可。

## 1，order项目定义接口，用于暴露



此接口能正常访问。

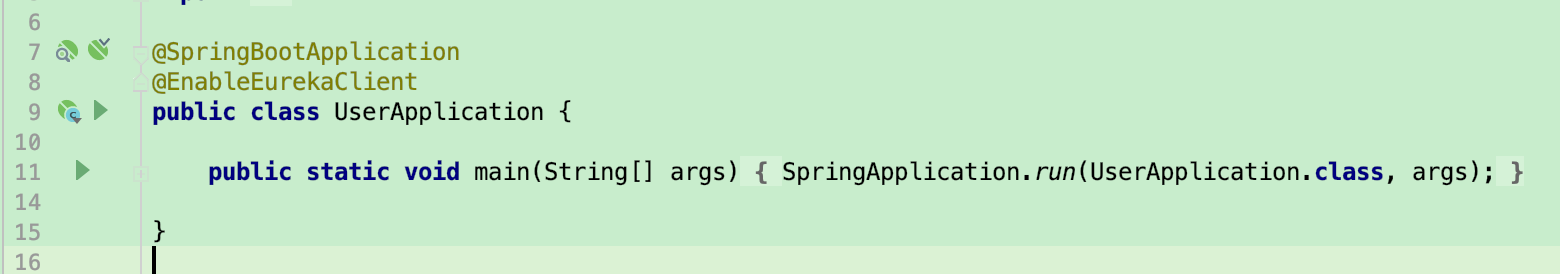
## 2，配置成为Eureka客户端项目

引入通用依赖：

|  |
| --- |
| <**dependencyManagement**>  <**dependencies**>  <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-dependencies</**artifactId**>  <**version**>Greenwich.RELEASE</**version**>  <**type**>pom</**type**>  <**scope**>import</**scope**>  </**dependency**>  </**dependencies**> </**dependencyManagement**> |
| <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-netflix-eureka-client</**artifactId**>  <**version**>2.1.0.RELEASE</**version**> </**dependency**> |

下面这个依赖为客户端依赖。

## 3，在启动类上添加注解 @EnableEurekaClient



注意是客户端的注解，与服务端对注解不一样。

## 4，配置文件配置

|  |
| --- |
| **server**:  **port**: 8881  **eureka**:  **prefer-ip-address**: true  **instance**:  **hostname**: localhost  **client**:  **service-url**:  **defaultZone**: http://${**eureka.instance.hostname**}:8000/eureka/  **spring**:  **application**:  **name**: order-8881 |

注意：

1，配置此项目的端口号

2，配置是否支持ip

3，配置主机名，跟Eureka服务器一样

4，连接地址中的端口微Eureka服务器上的端口

5，配置项目名字，多个微服务集群项目名字必须一样，比如有5个订单服务，必须是一个名字

OK，此时启动项目就可以将服务注册到Eureka服务器上，且在管理系统上能看到。

# API项目

API项目的配置与order项目的配置一样。

## 区别：

在使用远程调用时：

1，创建配置文件，加上注解



2，调用方式



# 健康检查

放到各个项目里

## 1，添加依赖

|  |
| --- |
| <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-actuator</**artifactId**> </**dependency**> |
| eureka:  client:  healthcheck:  enabled: true |

# ribbon就是@LoadBalance这个注解，用来做集群的。

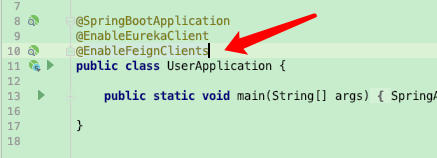
# Feign

用来传递多参数或参数是对象等情况

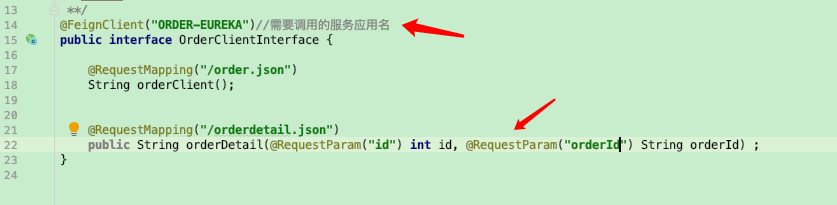
在需要调用的客户端加入依赖。比如user项目

|  |
| --- |
| *<!-- https://mvnrepository.com/artifact/org.springframework.cloud/spring-cloud-starter-openfeign -->* <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-openfeign</**artifactId**>  <**version**>2.1.0.RELEASE</**version**> </**dependency**> |

启动类上添加注解



定义一个接口



@FeignClient中为微服务名称

@RequestMapping()为需要调用的微服务中的URL

有多个参数，必须在参数前加上注解

方法调用：

