eureka:尤里卡

springcloud基于springboot，因此项目构建的话还是以springboot为主

构建地址：

<https://start.spring.io/>

选择的依赖为Eureka Server

pom.xml

<?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 https://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>2.2.1.RELEASE</version>

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

</parent>

<groupId>com.ycm</groupId>

<artifactId>eurekaserver</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>eurekaserver</name>

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

<properties>

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

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

</dependencies>

<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>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

创建服务注册中心

需要使用到的组件是，SpringCloud中的Netflix Eureka，它是一个**服务发现**以及**服务注册**组件。

启动服务注册中心，在启动类中加入@EnableEurekaServer注解即可

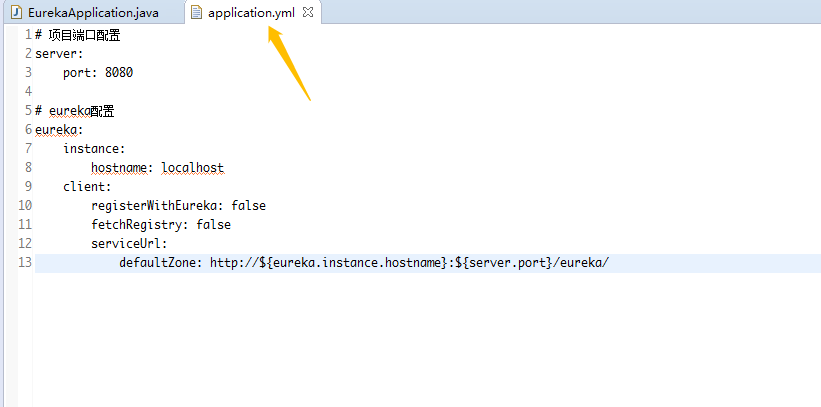
eureka是一个高可用的组件，它没有后端缓存，每一个实例注册之后需要向注册中心发送心跳（因此可以在内存中完成），在默认情况下erureka server也是一个eureka client ,必须要指定一个 server。

在application配置文件中通过

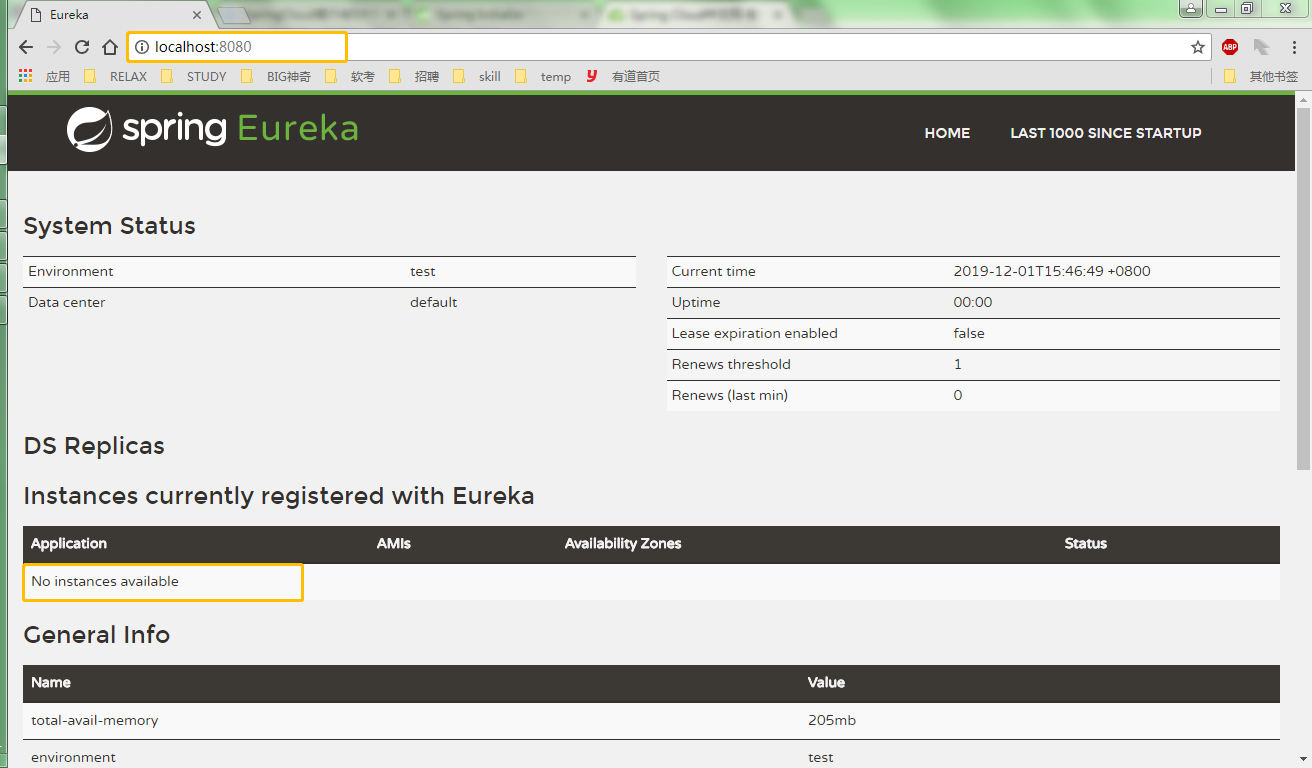
eureka.client.registerWithEureka：false

eureka.client.fetchRegistry：false

两个属性来表明自己是一个eureka server



启动工程后，通过http://localhost:端口号 来访问eureka server的web界面来查看相关的服务信息



因为还没有服务发现所以注册的实例暂时没有

服务提供者（client）

在pom文件很重相对于服务发现者（eureka server）增加了springmvc依赖

<dependency>

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

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

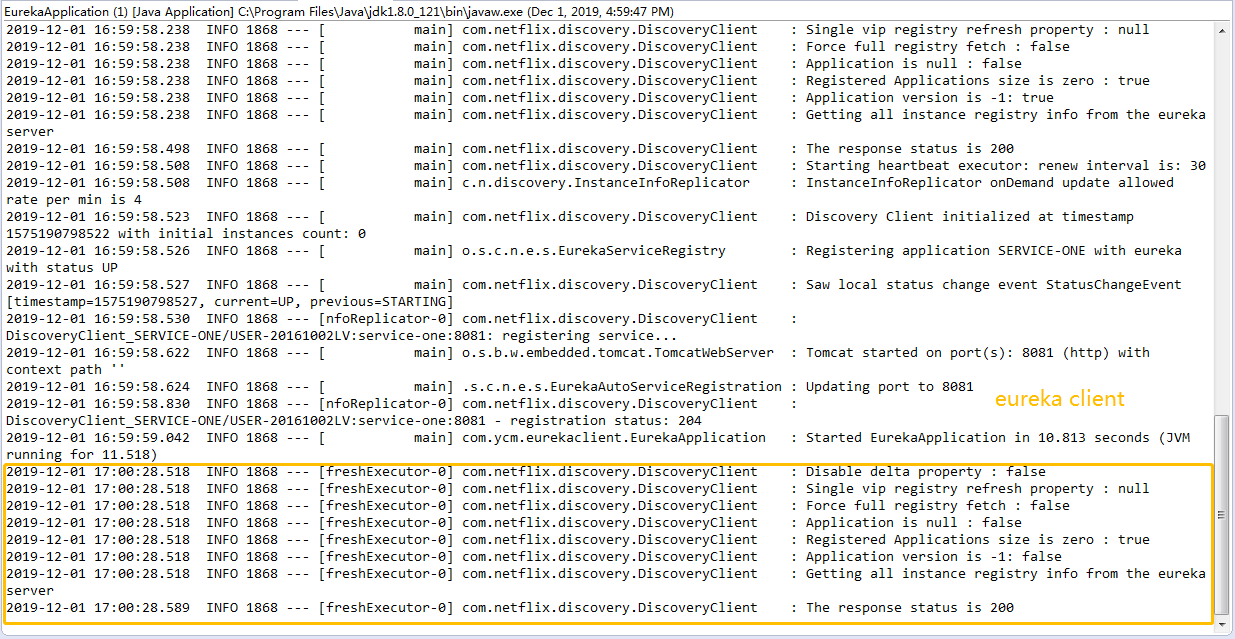
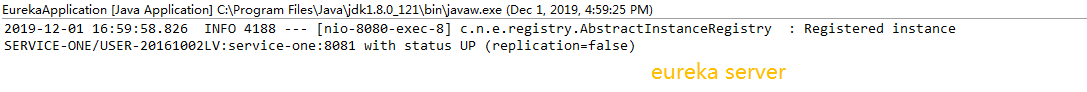
</dependency>

配置文件

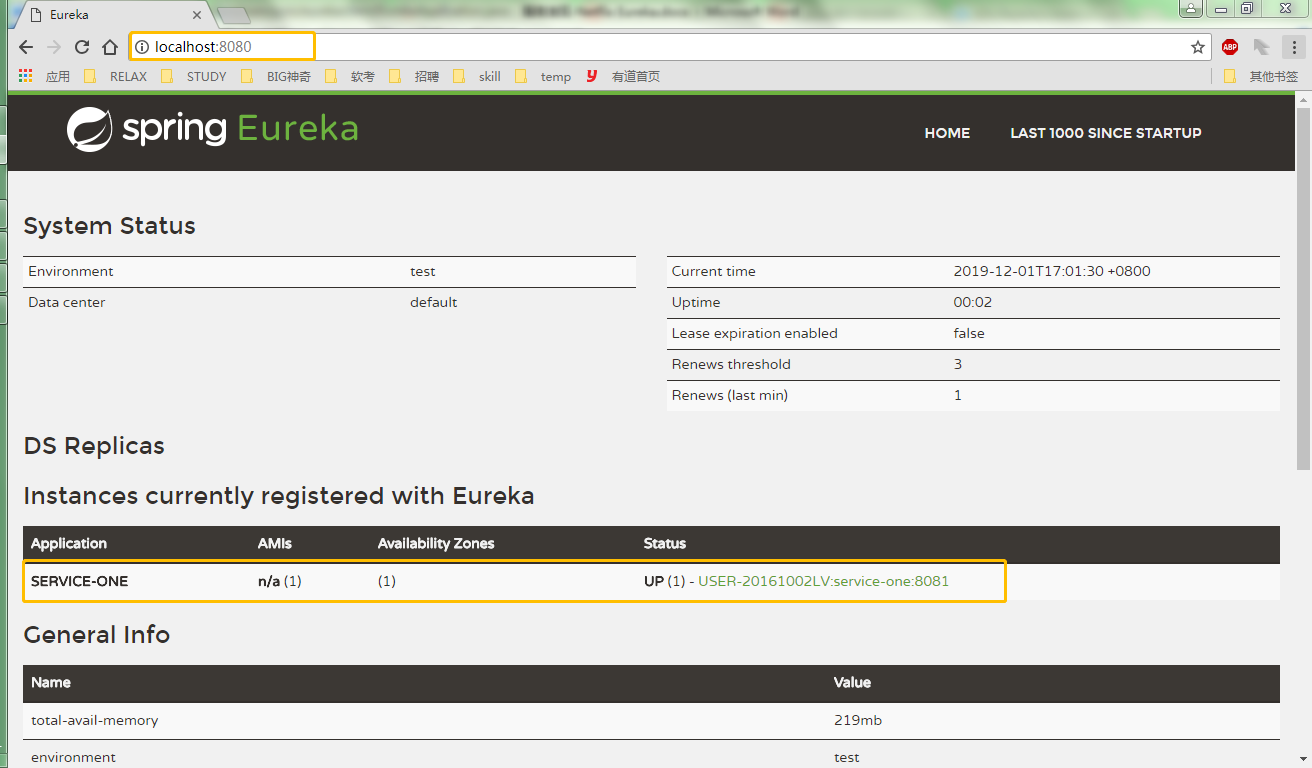


注意：eureka server和eureka client的context-path都是””，也就是没有设置项目的上下文，locahost:port就可以访问，而且eureka.client.serviceUrl.defaultZone**都是/eureka结尾** ，如果eureka server设置了**server.servlet.context-path** ，那么eureka client中就是context-path的具体路径/eureka的写法。

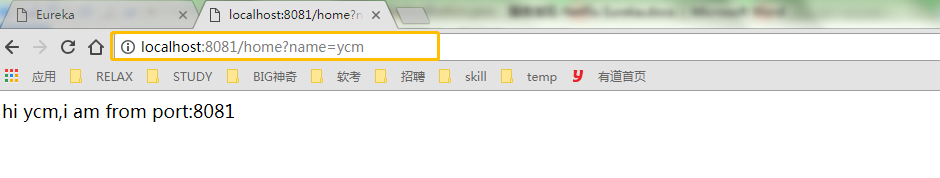
eureka server与eureka client控制台相关信息



eureka server显示注册进来的eureka client



eureka client正常访问



[**原文：史上最简单的 SpringCloud 教程 | 第一篇： 服务的注册与发现（Eureka）**](https://blog.csdn.net/forezp/article/details/69696915)