

homepage 第一期：Eureka Server 的开发 ， 包括单节点和多节点的开发

作者：[ChenforCode](#)

原文链接：<https://ld246.com/article/1573377269426>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



1.第一步，建立一个springboot工程，这个地方不再赘述，但是有些点需要注意，因为这次是springcloud的应用，也就是有多个模块，那么在父模块的pom文件里，有一些和单模块应用的不同，记录一下pom文件

```
<?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>cn.chenforcode.homepage</groupId>
    <artifactId>imooc-homepage</artifactId>
    <version>1.0-SNAPSHOT</version>
    <modules>
        <module>homepage-eureka</module>
    </modules>

    <!-- 如果是父模块的pom文件的话，他的打包方式必须是pom -->
    <packaging>pom</packaging>
    <name>imooc-homepage-spring-cloud</name>
    <description>Project for ImoocHomepage SpringCloud</description>

    <!-- springboot项目都需要加的一个依赖 -->
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.1.4.RELEASE</version>
    </parent>

    <!-- 标注springcloud版本 -->
    <properties>
```

```

    <spring-cloud.version>Greenwich.RELEASE</spring-cloud.version>
</properties>

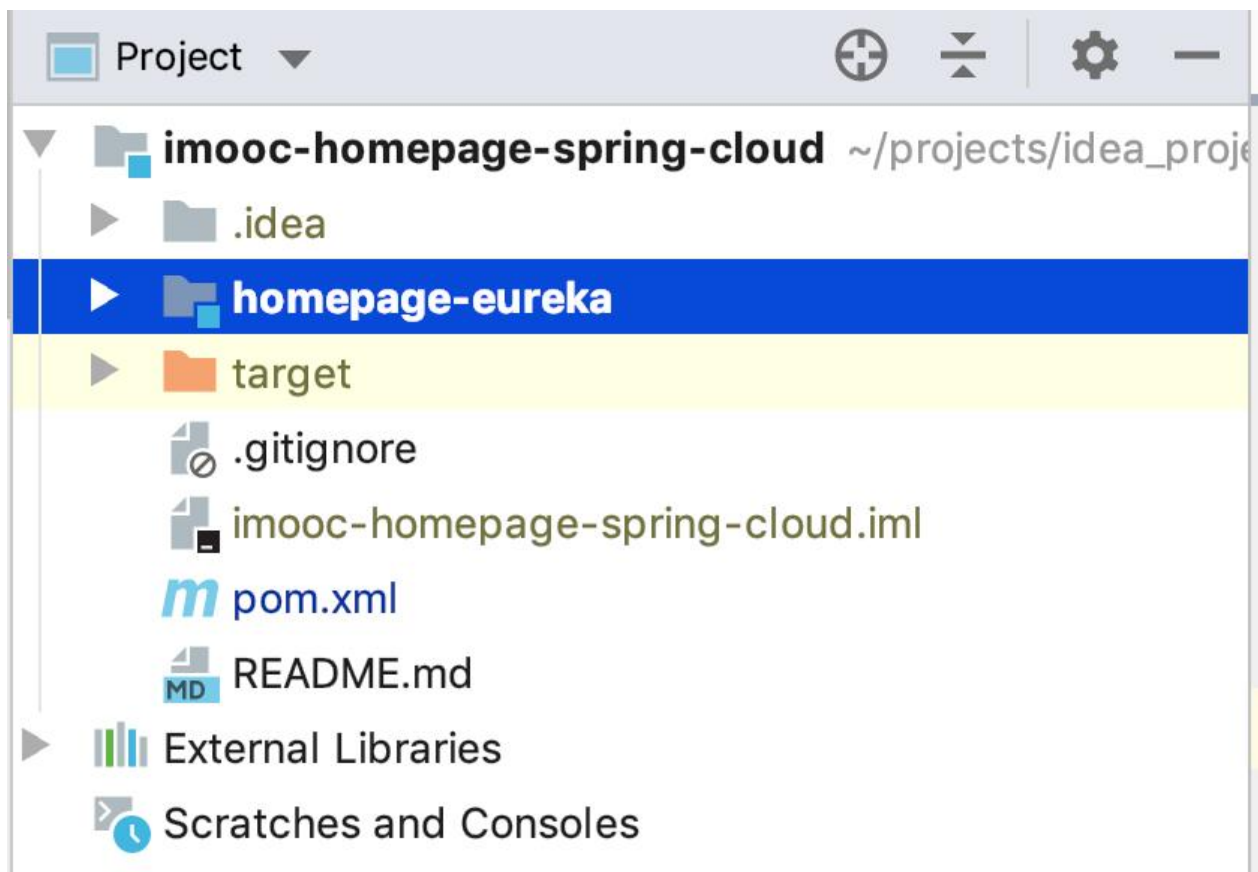
<!-- 引入相关的依赖 -->
<dependencies>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>

<!-- 管理springcloud的版本 -->
<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>

<!-- 配置maven远程仓库 -->
<repositories>
  <repository>
    <id>spring-milestones</id>
    <name>Spring Milestones</name>
    <url>http://repo.spring.io/milestone</url>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>
</project>

```

2.建立homepage-eureka模块，目前整个项目的结构如下图



3.完成改模块的pom文件编写

```
<?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">
  <parent>
    <artifactId>imooc-homepage</artifactId>
    <groupId>cn.chenforcode.homepage</groupId>
    <version>1.0-SNAPSHOT</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>

  <artifactId>homepage-eureka</artifactId>

  <!-- 指定这个子模块的版本 -->
  <version>1.0-SNAPSHOT</version>
  <!-- 指定打包的方式 -->
  <packaging>jar</packaging>

  <!-- 模块名及描述信息 -->
  <name>homepage-eureka</name>
  <description>Spring Cloud Eureka</description>

  <dependencies>
    <!-- Eureka Server:提供服务注册和服务发现 -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
```

```

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

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

4.编写启动类

```

package cn.chenforcode.homepage;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

/**
 * @author <a href="http://www.chenforcode.cn">PKUCoder</a>
 * @date 2019/11/9 4:34 下午
 * @description Eureka模块应用启动，只需要你加入@EnableEurekaServer就可以将这个模块变成
 * 一个eureka server
 * 同时还需要在pom文件中加入相应的依赖：spring-cloud-starter-netflix-eureka-server
 */
@SpringBootApplication
@EnableEurekaServer
public class EurekaApplication {
    public static void main(String[] args) {
        SpringApplication.run(EurekaApplication.class, args);
    }
}

```

5.编写配置文件

```

spring:
  application:
    name: homepage-eureka
server:
  port: 8000
eureka:
  instance:
    hostname: localhost
  client:
    # 表示是否从eureka server中获取注册信息 默认为true
    fetch-registry: false
    # 表示是否将自己注册到eureka server中 默认为true
    register-with-eureka: false
    service-url:

```

```
defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/
```

6.这个时候启动项目，已经可以看到eureka server的后台页面

7.开始搭建多个节点的eureka server，首先模拟出来多个主机，但是都对应本机的ip地址，当然如果多台服务器的话可以忽略这一步，

8.在终端中输入 `sudo vim /etc/hosts`

然后在里边输入

```
127.0.0.1 server1
```

```
127.0.0.1 server2
```

```
127.0.0.1 server3
```

然后保存退出

9.编写bootstrap.yml文件，在springboot启动过程中，bootstrap的加载顺序是先于application.yml的，所以将eureka的相关配置都写入bootstrap。先把application给注释掉

```
spring:
  application:
    name: homepage-eureka
    profiles: server1
server:
  port: 8000
eureka:
  instance:
    hostname: server1
    prefer-ip-address: false
  client:
    service-url:
      defaultZone: http://server2:8001/eureka/, http://server3:8002/eureka/
```

```
spring:
  application:
    name: homepage-eureka
    profiles: server2
server:
  port: 8001
eureka:
  instance:
    hostname: server2
    prefer-ip-address: false
  client:
    service-url:
      defaultZone: http://server1:8000/eureka/, http://server3:8002/eureka/
```

```
spring:
  application:
    name: homepage-eureka
    profiles: server3
server:
```

```
port: 8002
eureka:
  instance:
    hostname: server3
    prefer-ip-address: false
  client:
    service-url:
      defaultZone: http://server1:8000/eureka/, http://server2:8001/eureka/
```

10.将项目打成jar包，在工程根目录下，即imooc-homepage-spring-cloud目录下输入
mvn clean package -Dmaven.test.skip=true -U，第一次打包的过程显得如此的漫长

12.打包完成后进入homepage-eureka模块的target目录下

运行

第一个终端输入

```
java -jar homepage-eureka-1.0-SNAPSHOT.jar --spring.profiles.active=server1
```

第二个终端输入

```
java -jar homepage-eureka-1.0-SNAPSHOT.jar --spring.profiles.active=server2
```

第三个终端

```
java -jar homepage-eureka-1.0-SNAPSHOT.jar --spring.profiles.active=server3
```

如果你确认前面的步骤都没有错的话，那么如果在启动的过程中抛出异常的话不用担心，这是因为每个服务都需要注册到另外的两个服务上，但是另外两服务可能还没有稳定，还没有注册好，如果三个务都稳定下来，就不会再抛出异常了

13.分别进入三个后台管理页面

即进入<http://127.0.0.1>

<http://127.0.0.1>

<http://127.0.0.1>

如果看到了

← → ↻ ⓘ localhost:8000

应用 百度一下，你就知道 学习 项



System Status

Environment

Data center

DS Replicas

server2

server3

← → ↻ ⓘ localhost:8001

 应用  百度一下，你就知道  学习  项目管理 



System Status

Environment

Data center


DS Replicas

server1

server3

← → ↻ ⓘ localhost:8002

应用 百度一下, 你就知道 学习 项目



System Status

Environment

Data center

DS Replicas

server1

server2

就是没问题的

这里说明了homepage-eureka有三个副本，地址分别是下边三个

Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
Homepage-Eureka	n/a (3)	(3)	UP (3) - pkucoder.local:homepage-eureka:8002 , pkucoder.local:homepage-eureka:8001 , pkucoder.local:homepage-eureka:8000

这里说明了已经注册的副本和可用的副本

General Info

Name	Value
total-avail-memory	828mb
environment	test
num-of-cpus	12
current-memory-usage	204mb (24%)
server-uptime	00:05
registered-replicas	http://server2:8001/eureka/, http://server3:8002/eureka/
unavailable-replicas	
available-replicas	http://server2:8001/eureka/,http://server3:8002/eureka/