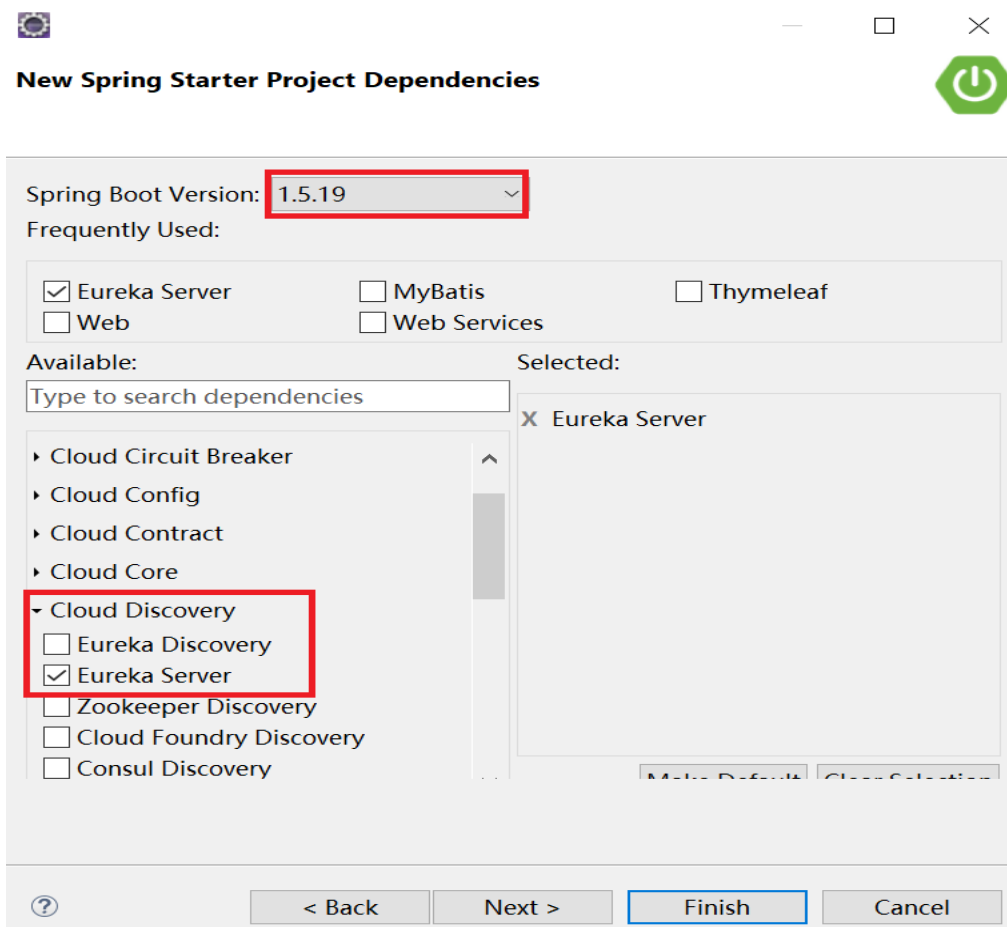
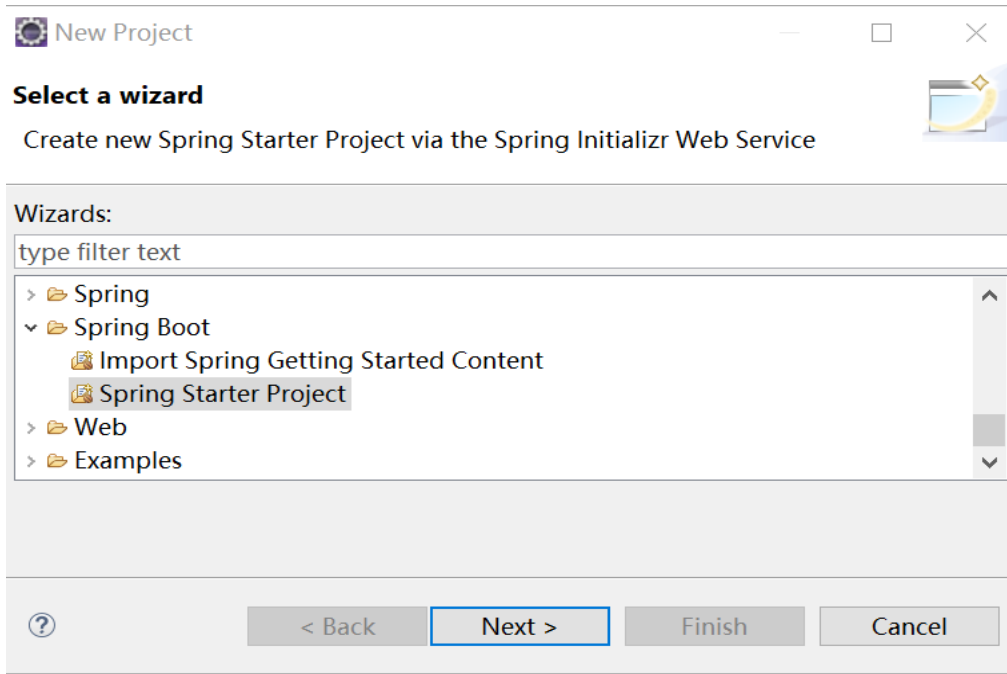
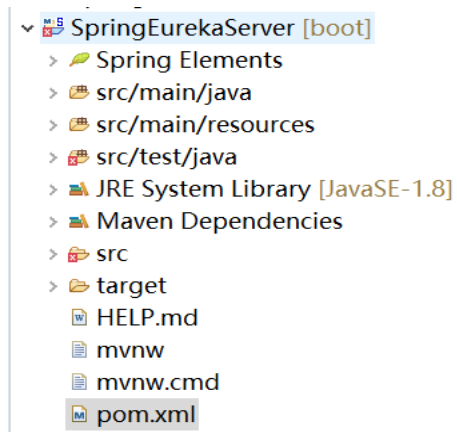


1. 建立 eurekaservice

2. New Spring Starter Project



3. test 报错，删除即可



4. 修改启动 class，引入@EnableEurekaServer，

```
package com.example.EurekaServer;

import org.springframework.boot.SpringApplication;

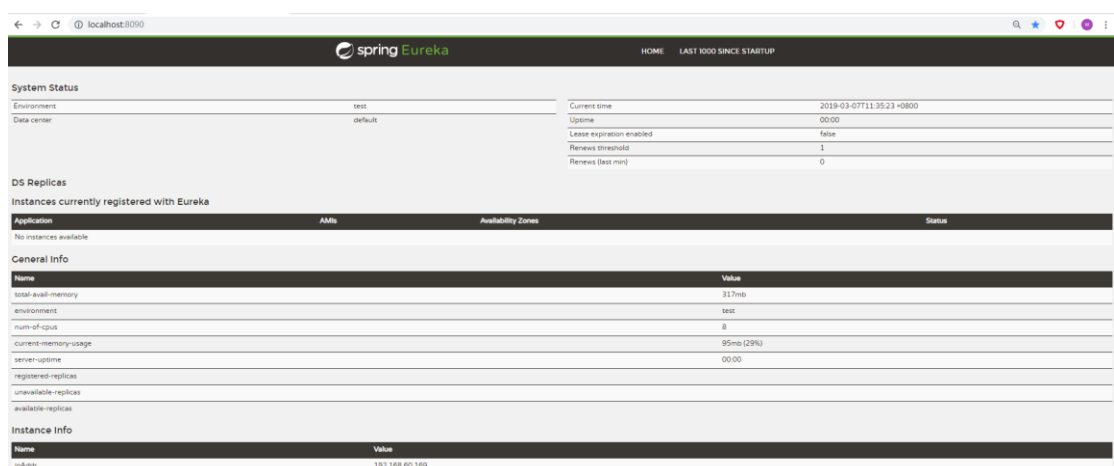
@SpringBootApplication
@EnableEurekaServer
public class SpringEurekaServerApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringEurekaServerApplication.class, args);
    }
}
```

application.properties 加入服务参数

```
application.properties
1 #服务端口号
2 server.port=8090
3
4
5 eureka.instance.hostname=localhost
6 #不需要注册为客户端
7 eureka.client.register-with-eureka=false
8 #不需要服务检索
9 eureka.client.fetch-registry=false
10
11 eureka.client.serviceUrl.defaultZone=http://${eureka.instance.hostname}:${server.port}/eureka/
12
13 #关闭自我保护
14 eureka.server.enable-self-preservation=false
15
16
17
```

5. 启动服务，url 输入：localhost: 8090. 画面显示如下



创建 Eureka Client 模块

6. 按上面的方法，创建 Client。设置配置档，引入注解

SpringEurekaClient1/pom.xml application.properties SpringEurekaClient1Application.java

```
15 <description>demo project for Spring Boot</description>
16
17 <properties>
18   <java.version>1.8</java.version>
19   <spring-cloud.version>Edgware.SR5</spring-cloud.version>
20 </properties>
21
22 <dependencies>
23   <dependency>
24     <groupId>org.springframework.boot</groupId>
25     <artifactId>spring-boot-starter-actuator</artifactId>
26   </dependency>
27
28   <dependency>
29     <groupId>org.springframework.boot</groupId>
30     <artifactId>spring-boot-starter-web</artifactId>
31   </dependency>
32   <dependency>
33     <groupId>org.springframework.cloud</groupId>
34     <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
35   </dependency>
36   <dependency>
37     <groupId>org.springframework.boot</groupId>
38     <artifactId>spring-boot-starter-test</artifactId>
39     <scope>test</scope>
40   </dependency>
41
42   <!-- devtools 热启动 -->
43   <dependency>
44     <groupId>org.springframework.boot</groupId>
45     <artifactId>spring-boot-devtools</artifactId>
46   </dependency>
47 </dependencies>
48
49 <dependencyManagement>
```

SpringEurekaClient1/pom.xml application.properties SpringEurekaClient1App

```
1 #配置eureka服务器地址
2 eureka.client.serviceUrl.defaultZone=http://localhost:8090/eureka/
3 #eurekaclient的服务端口
4 server.port=8091
5
6 spring.application.name=eurekaclient1
```

SpringEurekaClient1/pom.xml application.properties SpringEurekaClient1Application.java

```
1 package com.example.EurekaClient1;
2
3 import org.springframework.boot.SpringApplication;
4
5
6
7 @SpringBootApplication
8 @EnableEurekaClient
9 public class SpringEurekaClient1Application {
10
11     public static void main(String[] args) {
12         SpringApplication.run(SpringEurekaClient1Application.class, args);
13     }
14 }
15
16
```

7. 启动客户端，查看服务端。能看到 client:8091, lient:8092

Spring Eureka

HOME LAST 1000 SINCE STARTUP

System Status

Environment

Data center

test

default

Current time

Uptime

Lease expiration enabled

Renews threshold

Renews (last min)

2019-03-07T15:11:58 +0800

00:40

false

5

4

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

DS Replicas

Instances currently registered with Eureka

Application

IDs

Availability Zones

Status

EUREKACLIENT1

n/a (1)

(1)

UP (1) - LAPTOP-QHEX34KZ-eurekaclient1.8091

EUREKACLIENT2

n/a (1)

(1)

UP (1) - LAPTOP-QHEX34KZ-eurekaclient2.8092

General Info

Name

Value

total-avail-memory

314mb

environment

test

num-of-cpus

8

current-memory-usage

162mb (51%)

server-up-time

00:40

registered-replicas

unavailable-replicas

available-replicas

Instance Info

Name

Value

ipAddr

192.168.60.189

status

UP

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.