

# Spring Cloud简介

Spring Cloud 为开发人员提供了快速构建分布式系统的一些工具，包括配置管理、服务发现、断路器、路由、微代理、事件总线、全局锁、决策竞选、分布式会话等等。它运行环境简单，可以在开发人员的电脑上跑。另外说明spring cloud是基于springboot的。

版本：

spring Boot 2.0.3.RELEASE

Spring Cloud Finchley.RELEASE

## 创建服务注册中心

采用的是Eureka作为服务注册与发现的组件。

1.创建一个maven主工程，这个pom.xml文件作为父pom文件，其他module工程继承该pom，如下：

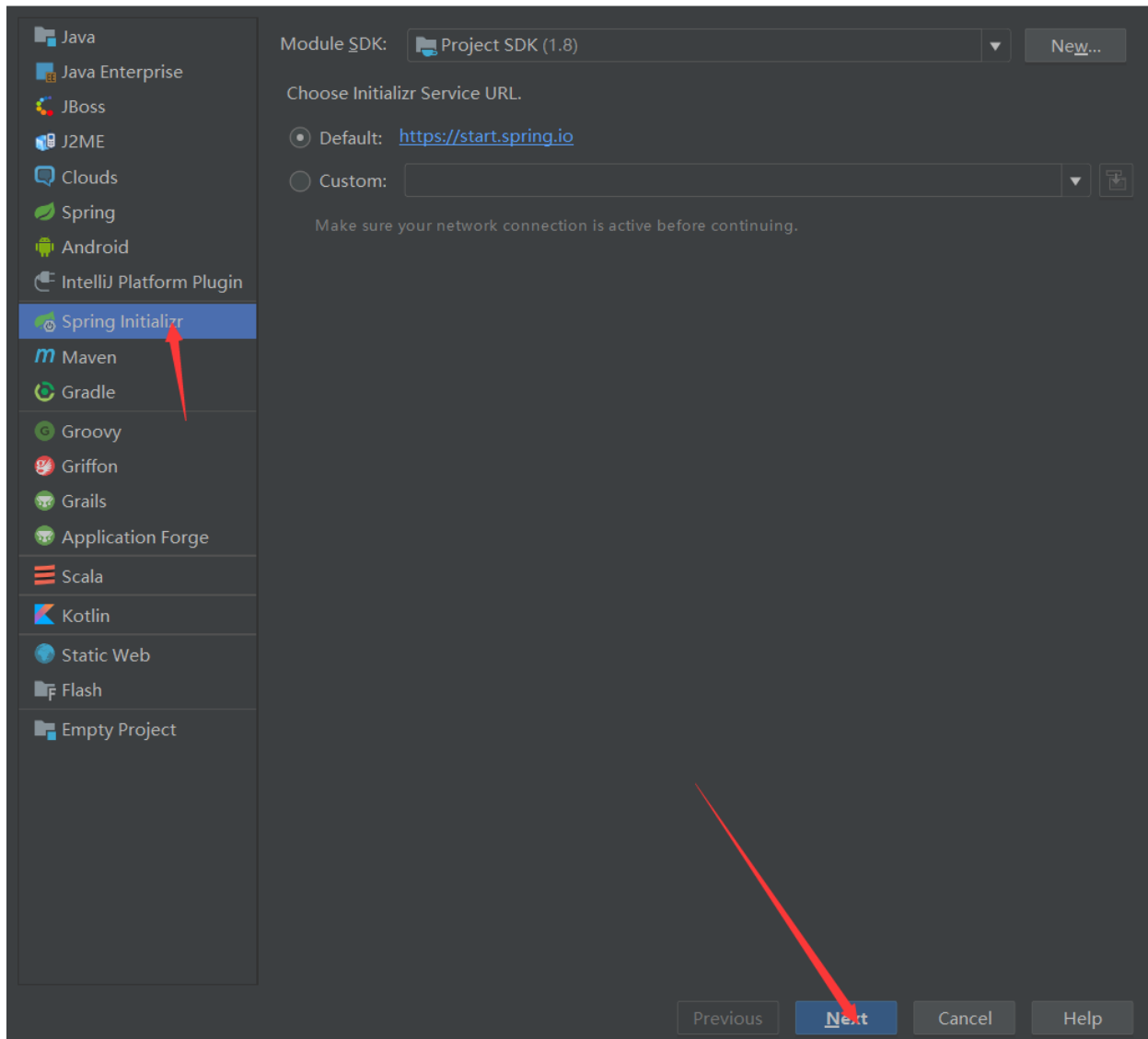
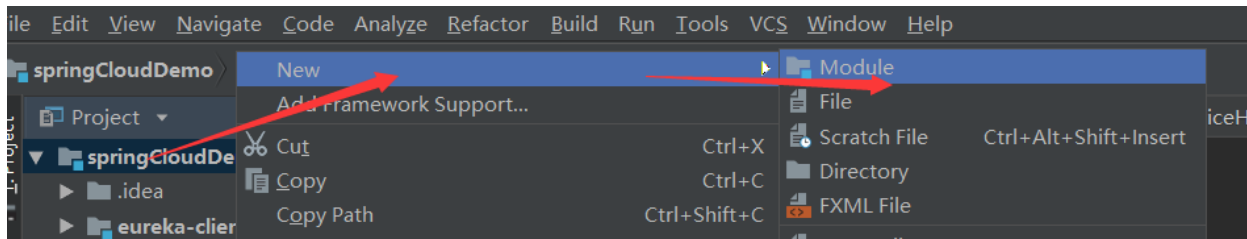
```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apach
e.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.gewdata</groupId>
8   <artifactId>springCloudDemo</artifactId>
9   <version>0.0.1-SNAPSHOT</version>
10  <packaging>pom</packaging>
11
12  <name>springCloudDemo</name>
13  <description>Demo project for Spring Boot</description>
14
15  <parent>
16    <groupId>org.springframework.boot</groupId>
17    <artifactId>spring-boot-starter-parent</artifactId>
18    <version>2.0.3.RELEASE</version>
19    <relativePath/>
20  </parent>
21
22  <properties>
23    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
```


```

24 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
25 <java.version>1.8</java.version>
26 <spring-cloud.version>Finchley.RELEASE</spring-cloud.version>
27 </properties>
28
29 <dependencies>
30
31 <dependency>
32 <groupId>org.springframework.boot</groupId>
33 <artifactId>spring-boot-starter-test</artifactId>
34 <scope>test</scope>
35 </dependency>
36
37 </dependencies>
38
39
40 <dependencyManagement>
41 <dependencies>
42 <dependency>
43 <groupId>org.springframework.cloud</groupId>
44 <artifactId>spring-cloud-dependencies</artifactId>
45 <version>${spring-cloud.version}</version>
46 <type>pom</type>
47 <scope>import</scope>
48 </dependency>
49 </dependencies>
50 </dependencyManagement>
51
52 <build>
53 <plugins>
54 <plugin>
55 <groupId>org.springframework.boot</groupId>
56 <artifactId>spring-boot-maven-plugin</artifactId>
57 </plugin>
58 </plugins>
59 </build>
60
61 </project>

```

2.创建2个model工程，一个model工程作为服务注册中心（Eureka Server），另一个服务作为客户端（Eureka Client），如下图所示：



 New Module ×

### Project Metadata

Group:

Artifact:

Type:  ▼

Language:  ▼

Packaging:  ▼

Java Version:  ▼


Version:

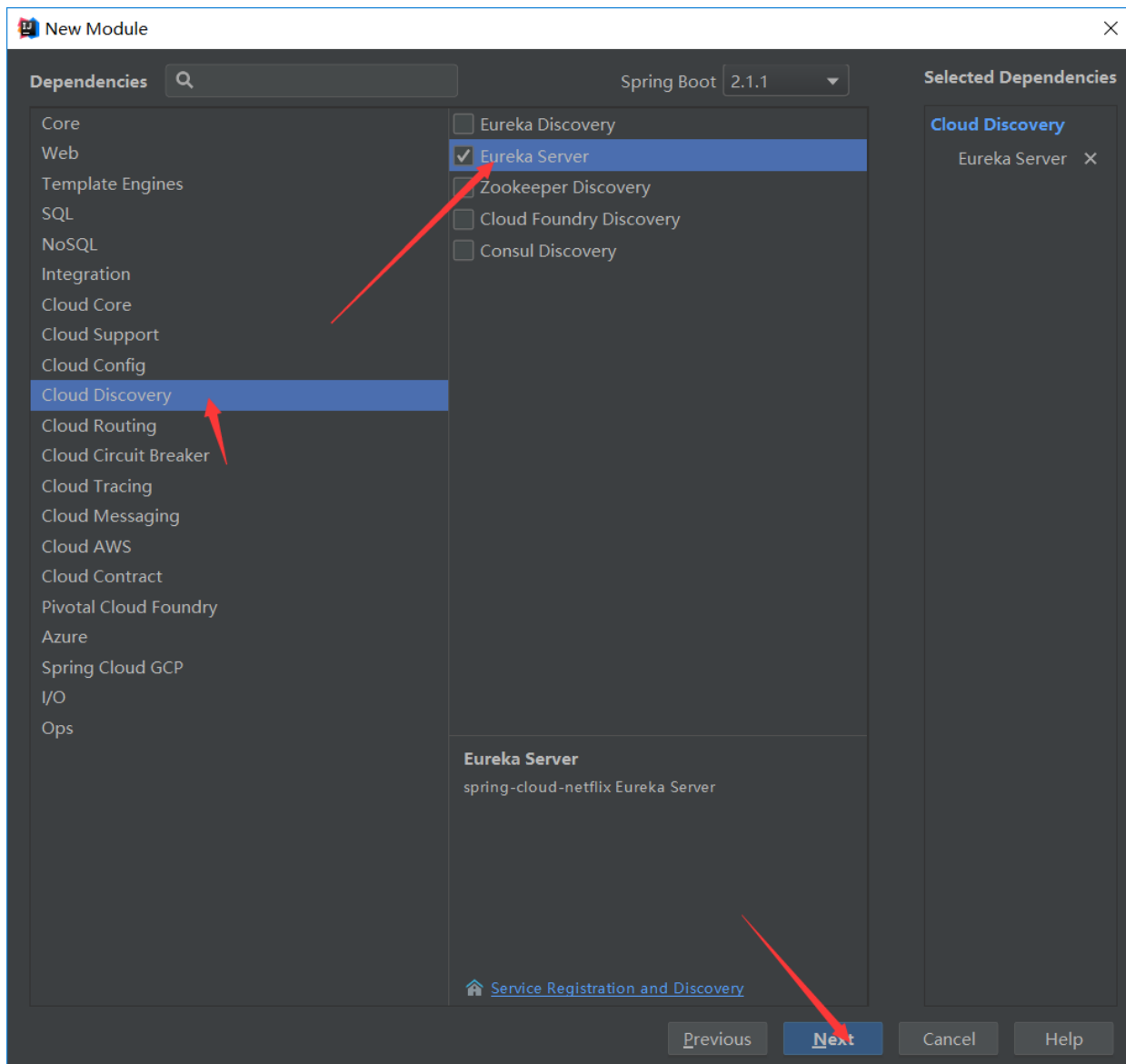
Name:

Description:

Package:

Previous Next Cancel Help





创建完工程后，其pom.xml继承了父pom文件，并引入spring-cloud-starter-netflix-eureka-server的依赖：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.
  w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apach
    e.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5
6   <groupId>com.gewdata</groupId>
7   <artifactId>eureka-server</artifactId>
8   <version>0.0.1-SNAPSHOT</version>
9   <packaging>jar</packaging>
10
11   <name>eureka-server</name>
12   <description>Demo project for Spring Boot</description>
13
```

```

14 <parent>
15 <groupId>com.gewdata</groupId>
16 <artifactId>springCloudDemo</artifactId>
17 <version>0.0.1-SNAPSHOT</version>
18 </parent>
19
20
21 <dependencies>
22 <dependency>
23 <groupId>org.springframework.cloud</groupId>
24 <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
25 </dependency>
26 </dependencies>
27
28 </project>

```

### 3.启动一个服务注册中心：

```

1 import org.springframework.boot.SpringApplication;
2 import org.springframework.boot.autoconfigure.SpringBootApplication;
3 import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;
4
5 @SpringBootApplication
6 @EnableEurekaServer
7 public class EurekaServerApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(EurekaServerApplication.class, args);
11     }
12 }

```

4.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eureka server也是一个eureka client，必须要指定一个 server。eureka server的配置文件application.yml：

```

1 server:
2   port: 8761
3
4 eureka:
5   instance:
6     hostname: localhost
7   client:
8     registerWithEureka: false

```

```
9  fetchRegistry: false
10  serviceUrl:
11  defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/
12
13  spring:
14  application:
15  name: eurka-server
```

通过eureka.client.registerWithEureka : false和fetchRegistry : false来表明自己是一个eureka server.

5.启动服务，访问http://localhost:8761出现界面。

## 创建服务提供者

当client向server注册时，它会提供一些元数据，例如主机和端口，URL，主页等。Eureka server 从每个client实例接收心跳消息。如果心跳超时，则通常将该实例从注册server中删除。

1.创建一个spring boot项目，同上；

2.pom.xml文件如下：

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apach
e.org/xsd/maven-4.0.0.xsd">
4  <modelVersion>4.0.0</modelVersion>
5
6  <groupId>com.gewdata</groupId>
7  <artifactId>eureka-client1</artifactId>
8  <version>0.0.1-SNAPSHOT</version>
9  <packaging>jar</packaging>
10
11  <name>eureka-client1</name>
12  <description>Demo project for Spring Boot</description>
13
14  <parent>
15  <groupId>com.gewdata</groupId>
16  <artifactId>springCloudDemo</artifactId>
17  <version>0.0.1-SNAPSHOT</version>
18  </parent>
19
```

```

20 <dependencies>
21 <dependency>
22 <groupId>org.springframework.cloud</groupId>
23 <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
24 </dependency>
25 <dependency>
26 <groupId>org.springframework.boot</groupId>
27 <artifactId>spring-boot-starter-web</artifactId>
28 </dependency>
29 </dependencies>
30
31 <build>
32 <plugins>
33 <plugin>
34 <groupId>org.springframework.boot</groupId>
35 <artifactId>spring-boot-maven-plugin</artifactId>
36 </plugin>
37 </plugins>
38 </build>
39
40 </project>

```

### 3.通过注解@EnableEurekaClient 表明自己是一个eurekaclient :

```

1 package com.gewdata;
2
3 import org.springframework.beans.factory.annotation.Value;
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6 import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RequestParam;
9 import org.springframework.web.bind.annotation.RestController;
10
11 @SpringBootApplication
12 @EnableEurekaClient
13 @RestController
14 public class EurekaClient1Application {
15
16     public static void main(String[] args) {
17         SpringApplication.run(EurekaClient1Application.class, args);
18     }

```



```
19
20 @Value("${server.port}")
21 private String port;
22
23 @RequestMapping("/hi")
24 public String home(@RequestParam(value = "name", defaultValue = "gewdat
a") String name) {
25     return "hi " + name + " ,i am from port:" + port;
26 }
27 }
```

#### 4.application.yml配置文件如下：

```
1 server:
2   port: 8762
3
4 spring:
5   application:
6     name: eureka-client1
7
8 # 注册时需要知道server的地址eureka.client.service-url.defaultZone
9 eureka:
10   client:
11     serviceUrl:
12       defaultZone: http://localhost:8761/eureka/
```

#### 5.启动工程，打开<http://localhost:8761>，可以发现有个服务已经注册在服务中了。