



链滴

SpringCloud 系列 --1.Eureka

作者: [289306290](#)

原文链接: <https://ld246.com/article/1583827386766>

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.搭建Eureka服务端

pom.xml中添加引用

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/m
ven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.crazyit.cloud</groupId>
  <artifactId>first-ek-server</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-dependencies</artifactId>
        <version>Dalston.SR1</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>

  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-eureka-server</artifactId>
    </dependency>
  </dependencies>

</project>
```

spring-cloud-starter-eureka-server 会把spring-boot-starter-web依赖自动引入,所以同时也支持web能力

```

# apple @ wujingjian in ~/coding/springcloud/codes/03/3.2/first-ek-server [15:17:24]
$ mvn dependency:tree
[INFO] Scanning for projects...
[INFO]
[INFO]
[INFO] Building first-ek-server 0.0.1-SNAPSHOT
[INFO]
[INFO]
[INFO] --- maven-dependency-plugin:2.8:tree (default-cli) @ first-ek-server ---
[INFO] org.crazyit.cloud:first-ek-server:jar:0.0.1-SNAPSHOT
[INFO] \- org.springframework.cloud:spring-cloud-starter-eureka-server:jar:1.3.1.RELEASE:compile
[INFO]   +- org.springframework.cloud:spring-cloud-starter:jar:1.2.2.RELEASE:compile
[INFO]     +- org.springframework.boot:spring-boot-starter:jar:1.5.3.RELEASE:compile
[INFO]       +- org.springframework.boot:spring-boot:jar:1.5.3.RELEASE:compile
[INFO]         \- org.springframework:spring-context:jar:4.3.8.RELEASE:compile
[INFO]       +- org.springframework.boot:spring-boot-autoconfigure:jar:1.5.3.RELEASE:compile
[INFO]         +- org.springframework.boot:spring-boot-starter-logging:jar:1.5.3.RELEASE:compile
[INFO]           +- ch.qos.logback:logback-classic:jar:1.1.11:compile
[INFO]             \- ch.qos.logback:logback-core:jar:1.1.11:compile
[INFO]           +- org.slf4j:jcl-over-slf4j:jar:1.7.25:compile
[INFO]             +- org.slf4j:jul-to-slf4j:jar:1.7.25:compile
[INFO]             \- org.slf4j:log4j-over-slf4j:jar:1.7.25:compile
[INFO]         +- org.springframework:spring-core:jar:4.3.8.RELEASE:compile
[INFO]           \- org.yaml:snakeyaml:jar:1.17:runtime
[INFO]     +- org.springframework.cloud:spring-cloud-context:jar:1.2.2.RELEASE:compile
[INFO]       \- org.springframework.security:spring-security-crypto:jar:4.2.2.RELEASE:compile
[INFO]     +- org.springframework.cloud:spring-cloud-commons:jar:1.2.2.RELEASE:compile
[INFO]       \- org.springframework.security:spring-security-rsa:jar:1.0.3.RELEASE:compile
[INFO]         \- org.bouncycastle:bcpkix-jdk15on:jar:1.55:compile
[INFO]           \- org.bouncycastle:bcprov-jdk15on:jar:1.55:compile
[INFO]   +- org.springframework.cloud:spring-cloud-netflix-eureka-server:jar:1.3.1.RELEASE:compile
[INFO]     +- org.springframework.boot:spring-boot-starter-web:jar:1.5.3.RELEASE:compile
[INFO]       +- org.springframework.boot:spring-boot-starter-tomcat:jar:1.5.3.RELEASE:compile
[INFO]         +- org.apache.tomcat.embed:tomcat-embed-core:jar:8.5.14:compile
[INFO]         +- org.apache.tomcat.embed:tomcat-embed-el:jar:8.5.14:compile
[INFO]         \- org.apache.tomcat.embed:tomcat-embed-websocket:jar:8.5.14:compile
[INFO]       +- org.hibernate:hibernate-validator:jar:5.3.5.Final:compile
[INFO]         +- javax.validation:validation-api:jar:1.1.0.Final:compile
[INFO]         +- org.jboss.logging:jboss-logging:jar:3.3.0.Final:compile
[INFO]         \- com.fasterxml:classmate:jar:1.3.1:compile
[INFO]       +- org.springframework:spring-web:jar:4.3.8.RELEASE:compile
[INFO]         +- org.springframework:spring-aop:jar:4.3.8.RELEASE:compile
[INFO]           \- org.springframework:spring-beans:jar:4.3.8.RELEASE:compile
[INFO]       \- org.springframework:spring-webmvc:jar:4.3.8.RELEASE:compile
[INFO]         \- org.springframework:spring-expression:jar:4.3.8.RELEASE:compile
[INFO]     +- org.springframework.boot:spring-boot-starter-actuator:jar:1.5.3.RELEASE:compile
[INFO]       \- org.springframework.boot:spring-boot-actuator:jar:1.5.3.RELEASE:compile
[INFO]     +- org.springframework.boot:spring-boot-starter-freemarker:jar:1.5.3.RELEASE:compile
[INFO]       +- org.freemarker:freemarker:jar:2.3.26-incubating:compile
[INFO]       \- org.springframework:spring-context-support:jar:4.3.8.RELEASE:compile
[INFO]     +- org.springframework.cloud:spring-cloud-netflix-core:jar:1.3.1.RELEASE:compile
[INFO]     +- org.springframework.cloud:spring-cloud-netflix-eureka-client:jar:1.3.1.RELEASE:compile
[INFO]       +- com.netflix.eureka:eureka-client:jar:1.6.2:compile
[INFO]         +- org.codehaus.jettison:jettison:jar:1.3.7:runtime
[INFO]           \- stax:stax-api:jar:1.0.1:runtime
[INFO]         +- com.netflix.netflix-commons:netflix-eventbus:jar:0.3.0:runtime
[INFO]           +- com.netflix.netflix-commons:netflix-infix:jar:0.3.0:runtime
[INFO]             +- commons-jxpath:commons-jxpath:jar:1.3:runtime

```

启动类添加注解 `@EnableEurekaServer`

```
package org.crazyit.cloud;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;
```

```
@SpringBootApplication
@EnableEurekaServer
public class FirstServer {
```

```

public static void main(String[] args) {
    new SpringApplicationBuilder(FirstServer.class).run(args);
}
}

```

在resource目录下面的applicaiton.yml文件中指定端口等属性

registerWithEureka: false

是否将自己的信息注册到Eureka服务器,默认值 true

fetchRegistry: false

是否到Eureka服务器中抓取注册信息,默认值是 true

server:

port: 8761

eureka:

client:

registerWithEureka: false

fetchRegistry: false

logging:

level:

com.netflix: INFO

启动服务浏览器访问如下:

The screenshot shows the Spring Eureka web interface. The top navigation bar includes 'HOME' and 'LAST 1000 SINCE STARTUP'. The main content area is divided into several sections:

- System Status:** A table showing environment details.

Environment	test	Current time	2020-03-10T15:31:09+0800
Data center	default	Uptime	00:02
		Lease expiration enabled	false
		Renews threshold	1
		Renews (last min)	0
- DS Replicas:** A section showing 'localhost' as a data center replica.
- Instances currently registered with Eureka:** A table with columns 'Application', 'AMIs', 'Availability Zones', and 'Status'. It currently shows 'No instances available'.
- General Info:** A table showing system metrics.

Name	Value
total-avail-memory	542mb
environment	test
num-of-cpus	8
current-memory-usage	119mb (21%)
server-uptime	00:02
registered-replicas	http://localhost:8761/eureka/
unavailable-replicas	http://localhost:8761/eureka/
available-replicas	

2.服务提供者

a.pom.xml中引入依赖

```

<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-eureka</artifactId>
  </dependency>
</dependencies>

```

完整代码如下:

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/m
ven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.crazyit.cloud</groupId>
  <artifactId>first-ek-service-provider</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-dependencies</artifactId>
        <version>Dalston.SR1</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>

  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-config</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-eureka</artifactId>
    </dependency>
  </dependencies>

</project>

```

b. application.yml中配置

```

spring:
  application:
    name: first-service-provider
eureka:
  instance:

```

```
hostname: localhost
client:
  serviceUrl:
    defaultZone: http://localhost:8761/eureka/
logging:
  level:
    com.netflix: INFO
```

将应用名称配置为first-server-provider,该服务将会被注册到端口为8761的Eureka服务器, eureka.instance.hostname配置该服务实例的主机名称.

在启动类上面使用 @EnableEurekaClient注解,声明是一个Eureka客户端,

```
package org.crazyit.cloud;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.cloud.netflix.eureka.EnableEurekaClient;
```

```
@SpringBootApplication
@EnableEurekaClient
public class FirstServiceProvider {

    public static void main(String[] args) {
        new SpringApplicationBuilder(FirstServiceProvider.class).run(args);
    }
}
```

编写一个简单的controller

```
package org.crazyit.cloud;
```

```
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
public class FirstController {

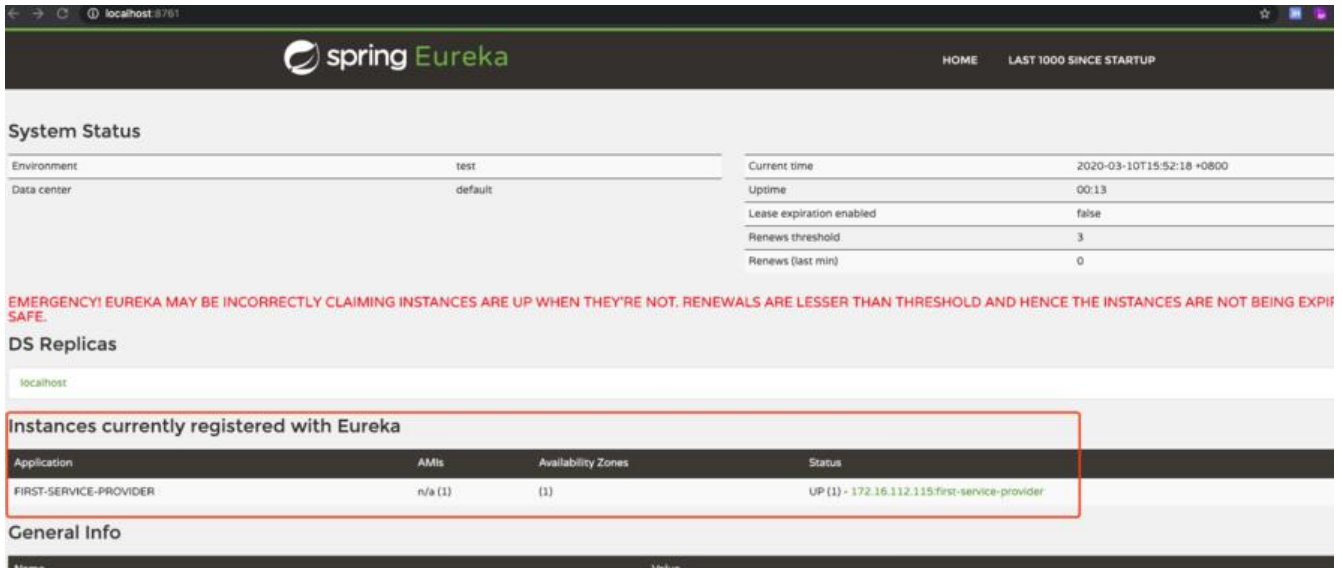
    @RequestMapping(value = "/person/{personId}", method = RequestMethod.GET,
        produces = MediaType.APPLICATION_JSON_VALUE)
    public Person findPerson(@PathVariable("personId") Integer personId) {
        Person person = new Person(personId, "Crazyit", 30);
        return person;
    }
}
```

对应的实体为:

```
package org.crazyit.cloud;
```

```
public class Person {  
    private Integer id;  
    private String name;  
    private Integer age;  
    public Person() {  
        super();  
    }  
    public Person(Integer id, String name, Integer age) {  
        super();  
        this.id = id;  
        this.name = name;  
        this.age = age;  
    }  
    public Integer getId() {  
        return id;  
    }  
    public void setId(Integer id) {  
        this.id = id;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public Integer getAge() {  
        return age;  
    }  
    public void setAge(Integer age) {  
        this.age = age;  
    }  
}
```

运行后访问eureka可以看到我们的实例



3.服务调用者

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
MLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/m
ven-4.0.0.xsd" >
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.crazyit.cloud</groupId>
  <artifactId>first-ek-service-invoker</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-dependencies</artifactId>
        <version>Dalston.SR1</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>

  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-config</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-eureka</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-ribbon</artifactId>
    </dependency>
  </dependencies>
</project>
```



```
    </dependency>
  </dependencies>
</project>
```

配置文件:

```
server:
  port: 9000
spring:
  application:
    name: first-service-invoker
eureka:
  instance:
    hostname: localhost
  client:
    serviceUrl:
      defaultZone: http://localhost:8761/eureka/
```

在启动类加上 @EnableDiscoveryClient注解

注意: @EnableEurekaClient注解已经包含了@EnableDiscoveryClient的功能,也就是说,一个Eureka客户端,本身就具有服务发现的能力。

```
package org.crazyit.cloud;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication
@EnableDiscoveryClient
public class FirstInvoker {

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

编写简单的Controller

```
package org.crazyit.cloud;

import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

@RestController
@Configuration
public class InvokerController {
```

```

@Bean
@LoadBalanced
public RestTemplate getRestTemplate() {
    return new RestTemplate();
}

@RequestMapping(value = "/router", method = RequestMethod.GET,
    produces = MediaType.APPLICATION_JSON_VALUE)
public String router() {
    RestTemplate restTpl = getRestTemplate();
    // 根据应用名称调用服务
    String json = restTpl.getForObject(
        "http://first-service-provider/person/1", String.class);
    return json;
}
}

```

注意: RestTemplate的Bean被@LoadBalanced注解修饰后,这个RestTemplate实例就具有访问分布式服务的能力了。

启动服务eureka中有两个实例

The screenshot shows the Spring Eureka dashboard. The 'System Status' section displays environment details (test, default) and system metrics (current time, uptime, lease expiration, renewals threshold, and renewals). A red warning message indicates that renewals are less than the threshold. The 'DS Replicas' section shows 'localhost'. The 'Instances currently registered with Eureka' section is highlighted with a red box and contains the following table:

Application	AMIs	Availability Zones	Status
FIRST-SERVICE-INVOKER	n/a (1)	(1)	UP (1) - 172.16.112.115:first-service-invoker:3000
FIRST-SERVICE-PROVIDER	n/a (1)	(1)	UP (1) - 172.16.112.115:first-service-provider

The 'General Info' section is partially visible at the bottom.

并且访问 localhost:9000/router结果如下:

The screenshot shows a browser window with the URL 'localhost:9000/router'. The response is displayed as JSON:

```

{
  "id": 1,
  "name": "Crazyit",
  "age": 30
}

```