

Eureka 实践作业

服务注册中心

1. 在 IDEA 中创建一个 Spring Cloud 工程，引入 Eureka-Server 包，pom 文件整体如下：

```
<?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>com.emest</groupId>
    <artifactId>eureka-server</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

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

    <!-- spring boot 的 parent 配置文件，有大部分 spring boot 需要用的 Jar 包 -->
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.3.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
    </properties>

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

```

</dependencies>

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>Camden.SR7</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>

```

2. application.yml 配置

```

#单例模式启动 Eureka Server
server:
  port: 8761 # 启动端口
eureka:
  client:
    registerWithEureka: false #false:不作为一个客户端注册到注册中心
    fetchRegistry: false #此客户端是否获取 eureka 服务器注册表上的注册信息
    serviceUrl:
      defaultZone: http://localhost:8761/eureka

```

配置解释：

- registerWithEureka：表示是否将自己注册到 Eureka Server，默认为 true。由于当前应用就是 Eureka Server，故而设置为 false。

- fetchRegistry：表示是否从 Eureka Server 获取注册信息，默认为 true。因为这是一个单

点的 Eureka Server，不需要同步其他的 Eureka Server 节点的数据，故而设置为 false。

- defalseZone：设置与 Eureka Server 交互的地址，查询服务和注册服务都需要依赖这个地址。

3. 编写启动类，在启动类上添加@EnableEurekaServer 注解，声明这是一个 Eureka Server。

```
package com.emest;

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

@EnableEurekaServer
@SpringBootApplication
public class EurekaServerApplication {

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

4. 访问 <http://localhost:8761/>，如下图所示：

The screenshot shows the Spring Eureka web interface. The header includes the Spring Eureka logo and navigation links for HOME and LAST 1000 SINCE STARTUP. The main content area is divided into three sections:

- System Status**: A table showing environment (test) and data center (default) on the left, and system metrics on the right.
- DS Replicas**: A section showing the local replica as 'localhost'.
- Instances currently registered with Eureka**: A table with columns for Application, AMIs, Availability Zones, and Status. It currently shows 'No instances available'.

System Status	
Environment	test
Data center	default
Current time	2018-07-25T15:24:45 +0800
Uptime	00:00
Lease expiration enabled	false
Renews threshold	3
Renews (last min)	0

DS Replicas	
localhost	

Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
No instances available			

将微服务注册到 Eureka Server 上

1. 在 IDEA 中创建一个 Spring Cloud 工程，pom 文件整体如下：

```
<?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>com.eureka</groupId>
    <artifactId>applicaton-service</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

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

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.3.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
    </properties>

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

    <dependencyManagement>
        <dependencies>
            <dependency>
```

```

        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-dependencies</artifactId>
        <version>Camden.SR7</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>

```

2. application.yml 配置

```

# HTTP (Tomcat) port
server:
  port: 8082
spring:
  application:
    name: micro-service
eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka/

```

3. 编写启动类，在启动类上添加@SpringBootApplication @EnableDiscoveryClient 注解，声明这是一个 Eureka Client。

```

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

@SpringBootApplication

```

```

@EnableDiscoveryClient
public class ServiceApplication {

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

```

4. 启动后，重新访问 <http://localhost:8761/>，如下图所示，我们可以看到刚刚启动的服务已经注册到我们的服务注册中心中。

The screenshot shows the Eureka Service Registry web interface in a browser window. The address bar shows 'localhost:8761'. The page is divided into several sections:

- System Status:** A table showing system configuration.

Environment	test	Current time	2018-08-21T11:54:23 +0800
Data center	default	Uptime	00:00
		Lease expiration enabled	false
		Renews threshold	3
		Renews (last min)	0
- DS Replicas:** A section showing the local data center replicas, with 'localhost' listed.
- Instances currently registered with Eureka:** A table showing the registered instances.

Application	AMIs	Availability Zones	Status
MICRO-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-Q46K8MM:micro-service:8082
- General Info:** A section partially visible at the bottom of the screenshot.