

服务生产与消费实践作业

使用第 2 章-Eureka 实践作业中建好的注册中心，之前我们已经建好了服务生产者，现在在服务生产者中添加接口，以方便服务消费者进行调用。

配置服务生产者

1. 添加 ServiceController 类：

```
package com.eureka;

import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

/**
 * Created by My on 2018/7/19.
 */

@RestController
public class ServiceController {
    @RequestMapping("hello/{name}")
    public String hello(@PathVariable String name) {
        return name + " say hello Eureka from port:8082";
    }
}
```

2. 参照该服务生产者再建立一个副本，服务端口改为 8083，关键代码如下：

```
# HTTP (Tomcat) port
server:
  port: 8083
spring:
  application:
    name: micro-service
eureka:
  client:
```

service-url:

defaultZone: http://localhost:8761/eureka/

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

```
/**  
 * Created by My on 2018/7/19.  
 */
```

@RestController

```
public class ServiceController {  
    @RequestMapping("hello/{name}")  
    public String hello(@PathVariable String name) {  
        return name + " say hello Eureka from port:8083";  
    }  
}
```

建立服务消费者

1. 在 IDEA 中创建一个 Spring Cloud 工程（服务消费者），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.eukeka</groupId>  
    <artifactId>micro-client</artifactId>  
    <version>0.0.1</version>  
    <packaging>jar</packaging>  
  
    <name>controller</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: 8081
spring:
  application:
    name: micro-client
eureka:
  client:
    service-url:
      defaultZone: http://localhost:8761/eureka/
# application parameters
app:
  service-url: http://MICRO-SERVICE/
```

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。

ControllerApplication 类：

```
package com.eukeka;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;
import org.springframework.cloud.client.loadbalancer.LoadBalanced;
import org.springframework.context.annotation.Bean;
import org.springframework.web.client.RestTemplate;

@SpringBootApplication
@EnableDiscoveryClient
public class ControllerApplication {

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

    @LoadBalanced
    @Bean
    RestTemplate restTemplate() {
        return new RestTemplate();
    }
}
```

4. 通过@LoadBalanced 注解，调用 netflix 的 ILoadBalancer 实现客户端负载均衡。然后通过 CallHelloService、CallHelloController 这两个类去调用服务提供者提供的 hello 接口，关键代码如下：

```
package com.eukeka;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

/**
 * Created by My on 2018/7/19.
 */
@Service
public class CallHelloService {

    @Value("${app.service-url}")
    private String appServiceUrl;

    @Autowired
    private RestTemplate restTemplate;

    public String callHello(String name) {
        // 是一个 http client
        ResponseEntity result = restTemplate.postForEntity(appServiceUrl + "hello/" + name, null,
String.class);
        return (String) result.getBody();
    }
}
```

```
package com.eukeka;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

/**
 * Created by My on 2018/7/19.
 */
```

```

@RestController
public class CallHelloController {

    @Autowired
    private CallHelloService callHelloService;

    @GetMapping("hello")
    public String hello(String name) {
        String result = callHelloService.callHello(name);
        return result;
    }
}

```

结果验证

1. 访问 <http://localhost:8761/> , 如下图所示 :

← → ↻ localhost:8761			
System Status			
Environment	test	Current time	2018-08-21T15:08:26 +0800
Data center	default	Uptime	03:14
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	12
DS Replicas			
localhost			
Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
MICRO-CLIENT	n/a (1)	(1)	UP (1) - DESKTOP-Q46K8MM:micro-client:8081
MICRO-SERVICE	n/a (2)	(2)	UP (2) - DESKTOP-Q46K8MM:micro-service:8082 , DESKTOP-Q46K8MM:micro-service:8083

2. 然后通过如下方式调用服务消费者可看到如下界面 :

← → ↻ localhost:8081/hello?name=devcloud
devcloud say hello Eureka from port:8082

3. 刷新页面 , 测试负载均衡 , 出现如下界面表示负载均衡配置成功 :

← → ↻ localhost:8081/hello?name=devcloud
devcloud say hello Eureka from port:8083