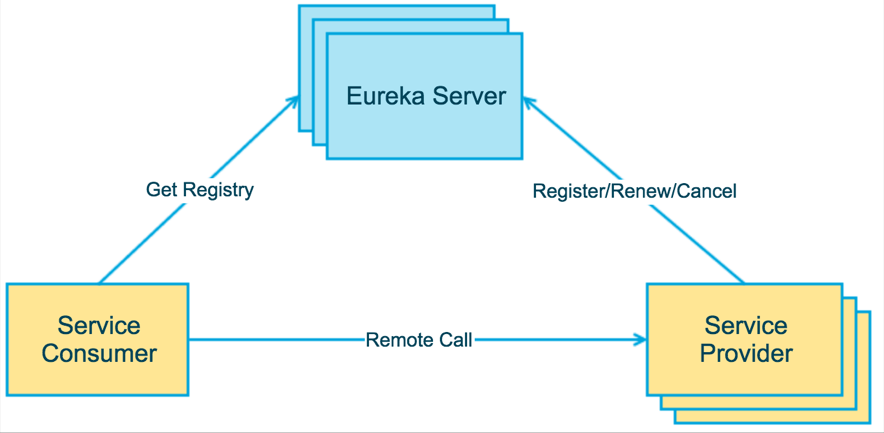
# 基本介绍

## Eureka的组件

Eureka由两个组件组成：Eureka服务器和Eureka客户端。**Eureka服务器用作服务注册服务器**。**Eureka客户端是用来简化与服务器的交互、作为轮询负载均衡器，并提供服务的故障切换支持**。Netflix在其生产环境中使用的是另外的客户端，它提供基于流量、资源利用率以及出错状态的加权负载均衡。

## Eureka的基本架构



1. Eureka Server

提供服务注册和发现

1. Service Provider

服务提供方，**将自身服务注册到Eureka**，从而使服务消费方能够找到。

1. Service Consumer

服务消费方，从Eureka获取注册服务列表，从而能够**消费服务**。

# Eureka Sever环境搭建

## pom.xml添加依赖

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-eureka-server</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

## 添加注解启动代码

@SpringBootApplication

@EnableEurekaServer

public class SpringCloudEurekaApplication {

public static void main(String[] args) {

SpringApplication.run(SpringCloudEurekaApplication.class, args);

}

}

## 添加配置文件 application.properties

spring.application.name=spring-cloud-eureka

server.port=8000

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

eureka.client.serviceUrl.defaultZone=http://localhost:${server.port}/eureka/

1. eureka.client.register-with-eureka ：表示是否将自己注册到Eureka Server，默认为true。
2. eureka.client.fetch-registry ：表示是否从Eureka Server获取注册信息，默认为true。
3. eureka.client.serviceUrl.defaultZone ：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。默认是http://localhost:8761/eureka ；多个地址可使用 , 分隔。

# Eureka服务提供与调用

## 服务提供

* 1. 配置POM

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-eureka</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

* 1. 配置文件

spring.application.name=spring-cloud-producer

server.port=9000

eureka.client.serviceUrl.defaultZone=http://localhost:8000/eureka/

* 1. 添加启动类

**@EnableDiscoveryClient 用来注册服务**

@SpringBootApplication

@EnableDiscoveryClient

public class ProducerApplication {

public static void main(String[] args) {

SpringApplication.run(ProducerApplication.class, args);

}

}

* 1. controller层

@RestController

public class HelloController {

@RequestMapping("/hello")

public String index(@RequestParam String name) {

return "hello "+name+"，this is first messge";

}

}

## 消费服务

* 1. pom包配置

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-eureka</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

* 1. 配置文件

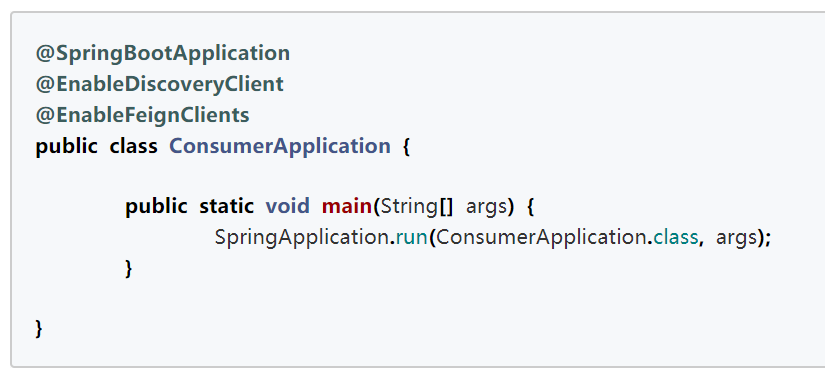
spring.application.name=spring-cloud-consumer

server.port=9001

eureka.client.serviceUrl.defaultZone=http://localhost:8000/eureka/

* 1. 启动类

启动类添加@EnableDiscoveryClient和@EnableFeignClients注解。



1. @EnableDiscoveryClient :启用服务注册与发现
2. @EnableFeignClients：启用feign进行远程调用
   1. feign调用实现

注解@FeignClient用来进行调用服务，name代表服务名称

@FeignClient(name= "spring-cloud-producer")

public interface HelloRemote {

@RequestMapping(value = "/hello")

public String hello(@RequestParam(value = "name") String name);

}

* 1. web层调用远程服务

将HelloRemote注入到controller层，像普通方法一样去调用即可。

