

一 Ribbon 引入

使用 eureka 治理组件，仅仅是将一个集群中管理的信息从之前 nginx 静态配置 upstream 变成了进程中内存的动态数据双层 map。并不能解决负载均衡调用的问题（服务被调用，服务的相互调用）

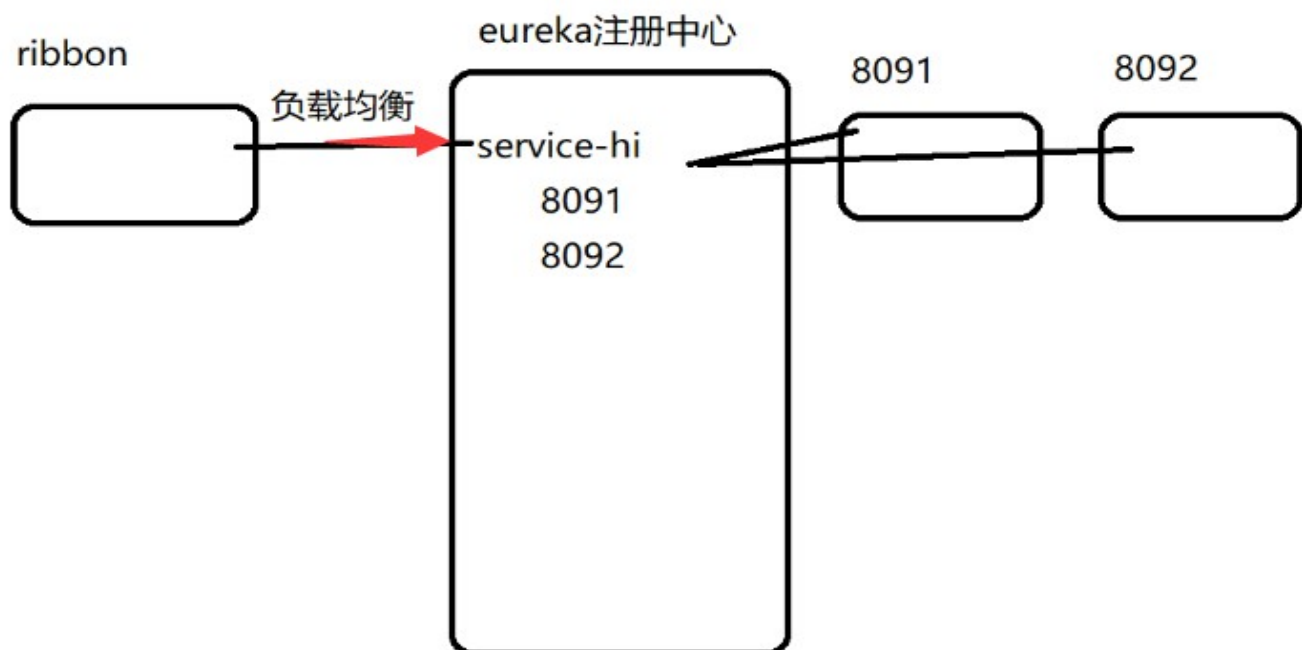
eureka 拆分后的项目（微服务）如何相互调用？

Ribbon

二 Ribbon 简介

负载均衡调用微服务的客户端组件。有了 ribbon，就可以通过服务名称负载均衡的访问注册到 eureka 注册中心的微服务。可以是轮询，也可以是权重，也可以是随机（轮询）

三 Ribbon 访问结构



Ribbon 需要通过“服务名称”调用对应的微服务，微服务信息只能从 eureka 注册中心中抓取，而只有 eureka 客户端具有抓取功能，所以 ribbon 需要配合 eureka 客户端实现负载均衡调用。

导入依赖时需要导入 ribbon 依赖，也需要导入 eureka 客户端依赖。

四 创建 Ribbon 工程

1 pom.xml 配置

1.A 继承 spring-boot

```
<!--继承 springboot-->
<parent>
    <artifactId>spring-boot-starter-parent</artifactId>
    <groupId>org.springframework.boot</groupId>
    <version>1.5.9.RELEASE</version>
</parent>
```

1.B 导入依赖

```
<!--eureka-client-->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
<!--ribbon-->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-ribbon</artifactId>
</dependency>
```

2 application.properties

```
#端口号
server.port=8094
#服务名称
spring.application.name=service-ribbon
#eureka -client
#ip 优先
eureka.instance.preferIpAddress=true
#注册中心地址
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
```

3 启动类

```
@SpringBootApplication
@EnableEurekaClient
public class StarterRibbon {
    public static void main(String[] args) {
        SpringApplication.run(StarterRibbon.class, args);
    }
    @Bean//注册为 bean
    @LoadBalanced//ribbon 组件会在这个对象上做一个记号，一旦这个对象发起 http 请求
        //经过 ribbon 负载均衡计算的拦截
        ///添加 ribbon 的拦截逻辑，否则无法实现服务调用
    public RestTemplate init(){
        return new RestTemplate();
    }
}
```

创建 RestTemplate 对象添加了一个注解@LoadBalanced,扫描注解之后,所有 RestTemplate 对象的请求都会经过 LoadBalancerInterceptor 实现拦截 拦截方法 intercept

4 创建接口(controller)

```
@RestController
```

```

public class HelloController {
    @Autowired
    private HelloService helloService;
    //ribbon 工程可以直接访问的接口 ribbon/hello
    @RequestMapping("ribbon/hello")
    public String sayHi(String name){
        return "RIBBON:"+helloService.sayHi(name);
    }
}

```

5 创建 service

```

@Service
public class HelloService {
    @Autowired
    private RestTemplate template;
    public String sayHi(String name) {
        //想办法调用 8091 8092 8093 负载均衡
        //通过 template 对象,发起请求调用 service-hi
        String url="http://service-hi/client/hello?name="+name;
        String responseBody = template.getForObject(url, String.class);
        //hello name ,I am from 8091/8092/8093
        return responseBody;
    }
}

```

getForObject 的参数解释

- url: 想要用 restTemplate 的访问的资源 url 地址

- **service-hi 为服务名称**

#####微服务u配置

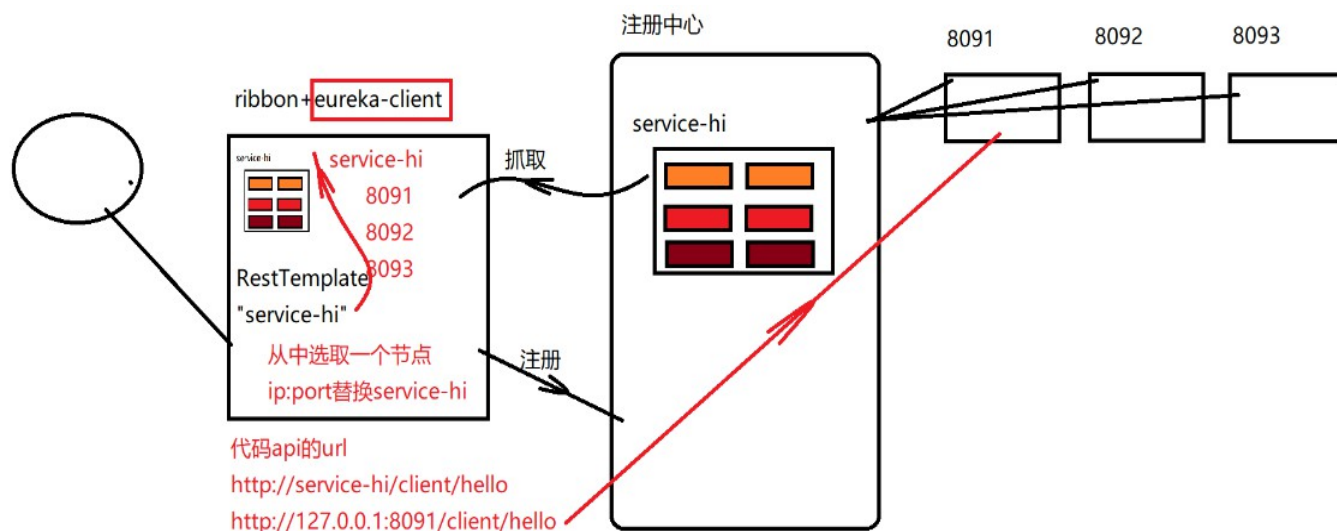
spring.application.name=service-hi
eureka.instance.prefer-ip-address=true

- eureka.client.service-url.defaultZone=http://127.

- responseType: 响应体内容数据类型的反射对象, String.class, 会将相应体的内容输出成字符串数据。如果服务器返回的是 json, 可以直接使用 json 对应的类型做参数, 比如: 访问的服务器 {"id":1,"name":"刘首付"}, String.class 解析相应体类型, 就是 json 字符串本身。自定义一个对应的 domain 类型

User {int id,String name} User.class

五 Ribbon 拦截逻辑



ribbon 工程启动时, eureka 客户端启动时抓取了注册中心的双层 map 数据

当 restTemplate 发送请求 api 调用时,进入到 ribbon 的拦截逻辑
将 url 地址中的 service-hi 解析出来得到服务名称从双层 map(第一层 map 的 key 值不就是服务名称),拿到这个服务所有实例提供者

拿到一个 list 8091 8092 经过负载均衡计算 Rule 规则得到轮询结果
拿到具体访问节点。

然后将服务名称替换为访问节点的域名、端口拼接 url 地址,请求发送出去

项目开发中, 可以将 ribbon 组件添加到微服务项目中, 实现负载均衡调用其它组件。

虽然微服务之间的调用实现了负载均衡, 那么前端调用时如何实现?
引入 zuul (网关) 组件