此文档是建立在[服务的注册于发现(Eureka)](onenote:" \l "服务的注册于发现(Eureka)&section-id={B46804C8-2079-4343-A399-0DFB75448724}&page-id={6088FC12-1292-4553-8CEC-7B6F6D0723A2}&end&base-path=https://d.docs.live.net/6599c42f24631913/文档/总结/SpringCloud.one)之上。

Spring Cloud有两种服务调用方式，本文介绍的是feign。

Feign是一个声明式的伪Http客户端，它使得写Http客户端变得更简单。使用Feign，只需要创建一个接口并注解。它具有可插拔的注解特性，可使用Feign 注解和JAX-RS注解。Feign支持可插拔的编码器和解码器。Feign默认集成了Ribbon，并和Eureka结合，默认实现了负载均衡的效果。

简而言之：

* Feign 采用的是基于接口的注解
* Feign 整合了ribbon

**创建feign模块**

New Module 
Dependencies 
Core 
web 
Template Engines 
SQL 
NOSQL 
Integration 
Cloud Core 
Cloud Config 
Cloud Discovery 
Cloud Routing 
Cloud Circuit Breaker 
Cloud Tracing 
Cloud Messaging 
Cloud AWS 
Cloud Contract 
Pivotal Cloud Foundry 
Azure 
spring Cloud GCP 
ops 
spring Boot 20.1 
Zuul 
Gateway 
Ribbon 
Feign 
Declarative REST clients with spring-cloud-netflix 
Feign 
Previous 
Selected Dependencies 
Cloud Routing 
Next 
Feign 
Cancel 
Help 

配置文件

**eureka:**

**client:**

**serviceUrl:**

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

**server:**

**port:**8765

**spring:**

**application:**

**name:**service-feign

在程序的启动类上加上@EnableFeignClients注解开启Feign的功能：

n@SpringBootApp1icatio 
@EnableDiscoveryCl i ent 
•Enabl eFeignC1 i ents 
public class ServiceFeignApp1ication { 
public static void main (StringC] args) [ 
SpringApp1 ication. run(ServiceFeignApp1 ication. class, 
args) 

定义一个feign接口，通过@ FeignClient（“服务名”），来指定调用哪个服务。比如在代码中调用了service-demo服务的“/demo”接口，代码如下：

@FeignClient (value 
@Service 
service-demo") 
public interface FeignServiceTest { 
@RequestMapping (value • 
"Idemo", method RequestMethod. GED 
String sayHiFromCl ient0ne (@RequestParam(va1ue = " rta1Ll ) String name) 

注意以下：

①@RequestMapping(value="/demo",method=RequestMethod.*GET*)中的value值需要和

@ FeignClient中调用的服务的@RequestMapping中的value值一致

@RequestMapping /demo 
public String home (ßRequestPara111 String name) { 
return "hi" + name + 
i am from " + port: 

②@RequestParam(value="name")中的value值需要和

@ FeignClient中调用的服务的@RequestParam中修饰的参数值一致

public String home gRequestParam String name) ( 
return "hi" + name + 
i am from " + port; 

在Web层的controller层，对外暴露一个”/test”的API接口，通过上面定义的Feign客户端FeignServiceTest 来消费服务。代码如下

@RestController 
public class TestController { 
@Autowired 
private FeignServiceTest feignServiceTest, 
@RequestMapping(va1ue 
'test", method = RequestMethod. GED 
public String sayHi (@RequestParam String name) 
return feignServiceTest. sayHiFromClient0ne (name) , 

启动程序，多次访问[http://localhost:8765/test?name=](http://localhost:8765/hi?name=forezp)齐涛,浏览器交替显示

hi齐涛, i am from 8762

hi齐涛, i am from 8763