spring cloud

1.父工程：

<parent>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <groupId>org.springframework.boot</groupId>  
 <version>2.6.6</version>  
</parent>  
<dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>com.alibaba.cloud</groupId>  
 <artifactId>spring-cloud-alibaba-dependencies</artifactId>  
 <version>2021.0.1.0</version>  
 <scope>import</scope>  
 <type>pom</type>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-dependencies</artifactId>  
 <version>2021.0.1</version>  
 <scope>import</scope>  
 <type>pom</type>  
 </dependency>  
 </dependencies>  
</dependencyManagement>

1. 第一个微服务 配置

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
</dependency>  
<dependency>  
 <groupId>com.alibaba.cloud</groupId>  
 <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-loadbalancer</artifactId>  
</dependency>

application.yml 配置

server:  
 port: 9000  
  
spring:  
 application:  
 name: consumer  
 cloud:  
 nacos:  
 discovery:  
 server-addr: localhost:8848

主程序：

@SpringBootApplication  
@EnableDiscoveryClient  
public class ConsumerApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(ConsumerApplication.class,args);  
 }  
}

1. 网关配置：

Pom：

<dependency>  
 <groupId>com.alibaba.cloud</groupId>  
 <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-gateway</artifactId>  
</dependency>  
<dependency> //因为版本太高了 spring gateway已经不支持 loadbalance 了  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-loadbalancer</artifactId>  
</dependency>

Application.yml

server:  
 port: 8001  
  
spring:  
 application:  
 name: gateway  
 cloud:  
 nacos:  
 discovery:  
 server-addr: localhost:8848  
 gateway:  
 routes:  
 - id: myconsumer  
  uri: lb://consumer  
 predicates:  
 - Path=/consumer/\*\*  
 *# enabled: true 不知道什么原因 这个开启然后关闭才会好使 很奇怪 实在是太奇怪了 真的是太奇怪了 哎*

主程序没有多余的配置哦

Openfeign 的使用哦

导报哦

<!-- 因为springcloud2021.0.1.0以上就完全没有Netflix了所以自然就没有ribbon了所以这个需要自己引-->

<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-loadbalancer</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-openfeign</artifactId>  
</dependency>

1. 在主启动了 添加Enable注解

@EnableFeignClients

1. 创建接口 接口里的方法为被调用方的方法

@FeignClient("provider") //里面是被调用的 项目名字哦 spring.application.name:这个东西  
public interface MyFeign {  
 @GetMapping("/provider/test")  
 String test();  
 @GetMapping("/provider2/test")  
 String test2();  
}

1. 调用使用 哦

@Autowired  
MyFeign myFeign;

@GetMapping("testfeign") // 这个如果多个的话 需要@FeignClient(value = "provider",contextId = "1")

public String testfeign(){  
 System.*out*.println(myFeign.test2());  
 return myFeign.test();  
}

如果 不在同一个模块的话 不能在同一个父工程下-

关于nacos配置哦

1.导包 bootstrap版本高了不自动加载了所以需要导包

<dependency>  
 <groupId>com.alibaba.cloud</groupId>  
 <artifactId>spring-cloud-starter-alibaba-nacos-config</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-bootstrap</artifactId>  
</dependency>

2.然后将application.yaml 改成bootstarp.yaml

server:  
 port: 9000  
  
spring:  
 profiles:  
 active: dev  
 application:  
 name: consumer  
 cloud:  
 nacos:  
 discovery:  
 server-addr: localhost:8848  
 config:  
 server-addr: localhost:8848  
 file-extension: yaml *#指定yaml格式配置*  
 namespace: *#这个要填ID* group: DEFAULT\_GROUP *#默认是这个 DEFAULT\_GROUP*

1. 添加自动刷新注解

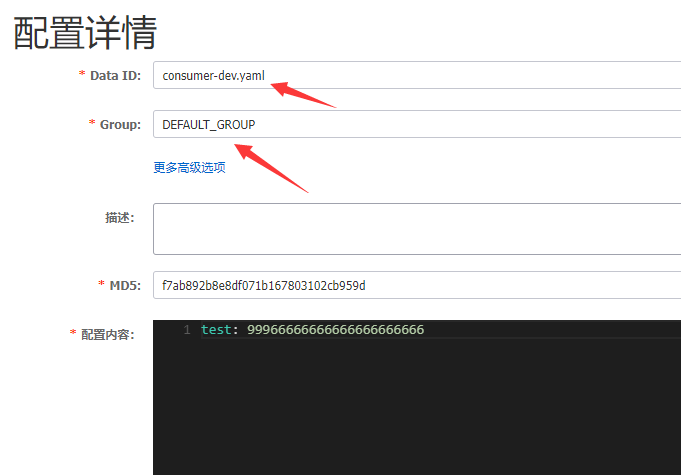
@RefreshScope // 开启配置更新

1. 测试

@Value("${name}")  
String name;

1. Nacos配置 那个是

${spring.application.name}-${spring.profiles.active}.${file-extension}



Sentinel的使用哦

1. 导报

<dependency>  
 <groupId>com.alibaba.cloud</groupId>  
 <artifactId>spring-cloud-starter-alibaba-sentinel</artifactId>  
</dependency>

1. 配置

sentinel:  
 transport:  
 dashboard: localhost:8080  
 port: 8827 #这个是向sentinel 发消息的端口

1. localhost:8080访问sentinel 设置限流
2. 测试

@GetMapping("test")  
@SentinelResource(value = "test",blockHandler = "test\_block",  
 fallback = "test\_fallback")  
public String test(){  
 double random = Math.*random*();  
 if(random > 0.5){ //模拟逻辑错误  
 int x = 1/0;  
 }  
 return "HaHaHa!!!" + test;  
}  
//block热点的错误才会 如果是逻辑的就不会去降级  
public String test\_block(BlockException e){  
  
 return "对不起你被限流了";  
}  
//fallback是逻辑错误 如果是限流的就不会去降级  
public String test\_fallback(){  
  
 return "对不起你出现错误了";  
}

1. Sentinel的 永久配置问题 具体的配置你需要查看文档哦

sentinel:  
 transport:  
 dashboard: localhost:8080  
 prot: 8719  
 web-context-unify: false  
 *#持久化配置* datasource:  
 ds1:  
 nacos:  
 server-addr: localhost:8848  
 dataId: ${spring.application.name}  
 groupId: DEFAULT\_GROUP  
 data-type: json  
 rule-type: flow  
  
 *#激活feign对sentinel的支持*feign:  
 sentinel:  
 enabled: true