



链滴

商城笔记 2----- 引入 Spring Cloud Alibaba 组件

作者: [wenyl](#)

原文链接: <https://ld246.com/article/1683264937146>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



git文档地址[spring-cloud-alibaba/README-zh.md at 2022.x · alibaba/spring-cloud-alibaba · GitHub](#)

spring cloud alibaba组件版本说明[版本说明 · alibaba/spring-cloud-alibaba Wiki · GitHub](#)

nacos配置

引入nacos组件依赖

在common工程中引入依赖

```
<dependency>
  <groupId>com.alibaba.cloud</groupId>
  <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
</dependency>
```

配置

在各个工程的yml中添加nacos配置

```
spring:
  cloud:
    nacos:
      config:
        server-addr: 127.0.0.1:8848
```

开启服务注册发现

在服务启动类添加注解

@EnableDiscoveryClient

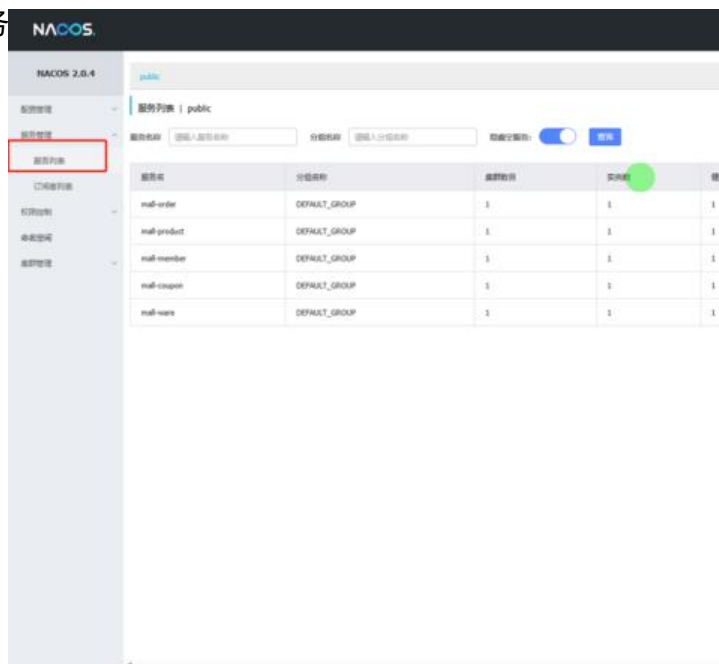
启动nacos

在bin目录下执行

startup.cmd -m standalone

启动工程

启动后访问<http://127.0.0.1/nacos>就可以看到注册的服务



将nacos作为配置中心

nacos参考: [nacos配置中心 - 问尤龙の时光 \(wenyoulong.com\)](http://wenyoulong.com)

@RefreshScope
@Value

这两个注解可以用于获取nacos配置中心得配置

OpenFeign调用

引入openFeign依赖

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

这里有个feign以来的问题, 启动如果报以下错误, 处理方法如下

No Feign Client for loadBalancing defined. Did you forget to include spring-cloud-starter-loadbalancer

Spring Cloud Feign 在Hoxton.M2 RELEASED版本之后不再使用Ribbon而是使用spring-cloud-loadbalancer

因此需要对spring cloud alibaba依赖做处理，剔除ribbon依赖，引入loadbalancer

```
<dependency>
  <groupId>com.alibaba.cloud</groupId>
  <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
  <exclusions>
    <exclusion>
      <groupId>com.netflix.ribbon</groupId>
      <artifactId>ribbon</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-loadbalancer</artifactId>
</dependency>
```

测试

在coupon工程得CouponController中新建方法

```
@RequestMapping("/member/list")
public R memberCoupon(){
    CouponEntity entity = new CouponEntity();
    entity.setCouponName("优惠券");
    return R.ok().put("coupons", Collections.singletonList(entity));
}
```

在member工程中开启远程服务调用，启动类上添加注解，basePackages即远程调用接口所在的包路径

```
@EnableFeignClients(basePackages = {"com.mall.member.feign"})
```

在对应的包下添加CouponFeignService接口，FeignClient注解中得服务名，就是coupon服务注册nacos中的服务名称

```
@FeignClient("mall-coupon")
public interface CouponFeignService {
    @RequestMapping("/coupon/coupon/member/list")
    public R memberCoupons();
}
```

在MemberController中添加接口访问coupon服务

```
@Resource
private CouponFeignService couponFeignService;
@RequestMapping("/coupons")
public R test(){
```

```
MemberEntity member = new MemberEntity();
member.setNickname("龙");
R memberCoupons = couponFeignService.memberCoupons();
return Objects.requireNonNull(R.ok().put("member", member)).put("coupons",memberCoupons.get("coupons"));
}
```

启动工程访问改接口即可

Spring cloud gateway网关

参考[spring cloud gateway配置 - 问尤龙の时光 \(wenyoulong.com\)](http://wenyoulong.com)