# Spring Cloud Alibaba

## 第一代SpringCloud

SpringCloud是基于**SpringBoot基础之上开发的微服务框架**，SpringCloud是一套目前非常完整的微服务解决方案框架，其内容包含服务治理、注册中心、配置管理、断路器、智能路由、微代理、控制总线、全局锁、分布式会话等。

SpringCloud包含众多的子项目

SpringCloud config 分布式配置中心

SpringCloud netflix 核心组件

Eureka:服务治理 注册中心

Hystrix:服务保护框架

Ribbon:客户端负载均衡器

Feign：基于ribbon和hystrix的声明式服务调用组件

Zuul: 网关组件,提供智能路由、访问过滤等功能。

SpringCloud中国社区：https://springcloud.cc/

## 第二代SpringCloud

SpringCloud第二代分布式组件，全部改为自己研发。

Spring Cloud Gateway

客户端软负载均衡 loadbalancer

spring-cloud-r4j(Resilience4J) 服务保护框架

## Alibaba SpringCloud

Sentinel：把流量作为切入点，从流量控制、熔断降级、系统负载保护等多个维度保护服务的稳定性。

**Nacos：一个更易于构建云原生应用的动态服务发现、配置管理和服务管理平台。**

RocketMQ：一款开源的分布式消息系统，基于高可用分布式集群技术，提供低延时的、高可靠的消息发布与订阅服务。

Dubbo：Apache Dubbo™ 是一款高性能 Java RPC 框架。

Seata：阿里巴巴开源产品，一个易于使用的高性能微服务分布式事务解决方案。

Alibaba Cloud ACM：一款在分布式架构环境中对应用配置进行集中管理和推送的应用配置中心产品。

Alibaba Cloud OSS: 阿里云对象存储服务（Object Storage Service，简称 OSS），是阿里云提供的海量、安全、低成本、高可靠的云存储服务。您可以在任何应用、任何时间、任何地点存储和访问任意类型的数据。

Alibaba Cloud SchedulerX: 阿里中间件团队开发的一款分布式任务调度产品，提供秒级、精准、高可靠、高可用的定时（基于 Cron 表达式）任务调度服务。

Alibaba Cloud SMS: 覆盖全球的短信服务，友好、高效、智能的互联化通讯能力，帮助企业迅速搭建客户触达通道。

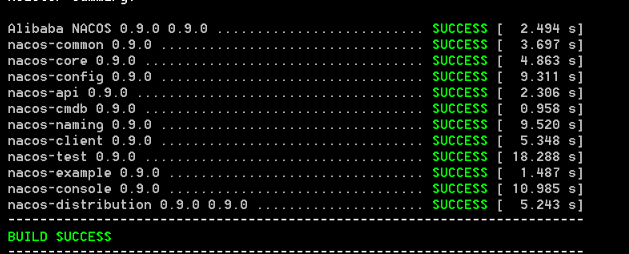
Alibaba SpringCloud

https://github.com/alibaba/spring-cloud-alibaba/blob/master/README-zh.md

## Nacos注册中心环境搭建

Nacos致力于帮助您发现、配置和管理微服务。Nacos提供了一组简单易用的特性集，帮助您快速实现动态服务发现、服务配置、服务元数据及流量管理。Nacos帮助您更敏捷和容易地构建、交付和管理微服务平台。Nacos是构建以“服务”为中心的现代应用架构 (例如微服务范式、云原生范式) 的服务基础设施。

1. 下载源码 git clone <https://github.com/alibaba/nacos.git>
2. 进入到该压缩目录F:\path\alibabaCloud\nacos-develop
3. mvn -Prelease-nacos clean install –U （打入到本地仓库中）



可能会有点耗时间 10分钟

进入到F:\path\alibabaCloud\nacos-develop\distribution\target\nacos-server-1.1.0\nacos\bin

启动startup.cmd

测试访问打开浏览器访问： http://localhost:8848/nacos 由于我使用的0.9.0 需要登录才可以访问 默认 账号密码为 nacos

## 基于Nacos构建服务注册与发现

mayikt\_producer

### 生产者

#### Maven依赖

|  |
| --- |
| <**parent**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-parent</**artifactId**>  <**version**>2.0.5.RELEASE</**version**> </**parent**> <**dependencies**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-web</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-alibaba-nacos-discovery</**artifactId**>  </**dependency**> </**dependencies**> <**dependencyManagement**>  <**dependencies**>  <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-dependencies</**artifactId**>  <**version**>Finchley.SR1</**version**>  <**type**>pom</**type**>  <**scope**>import</**scope**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-alibaba-dependencies</**artifactId**>  <**version**>0.2.2.RELEASE</**version**>  <**type**>pom</**type**>  <**scope**>import</**scope**>  </**dependency**>  </**dependencies**> </**dependencyManagement**> |

#### Bootstrap配置内容

|  |
| --- |
| **spring**:  **cloud**:  **nacos**:  **discovery**:  **server-addr**: 127.0.0.1:8848  **config**:  **server-addr**: 127.0.0.1:8848  **application**:  **name**: mayikt\_producer **server**:  **port**: 8082 |

#### 相关代码

|  |
| --- |
| @SpringBootApplication @RestController @EnableDiscoveryClient **public class** MemberApiService {   @Value(**"${mayikt.name}"**)  **private** String **name**;   @RequestMapping(**"/getMember"**)  **public** String getMember() {  **return "会员服务接口"** + **name**;  }   **public static void** main(String[] args) {  SpringApplication.*run*(MemberApiService.**class**);  } } |

### 消费者

mayikt\_consumer

#### Maven依赖

|  |
| --- |
| <**parent**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-parent</**artifactId**>  <**version**>2.0.5.RELEASE</**version**> </**parent**> <**dependencies**>  <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-web</**artifactId**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-alibaba-nacos-discovery</**artifactId**>  </**dependency**> </**dependencies**> <**dependencyManagement**>  <**dependencies**>  <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-dependencies</**artifactId**>  <**version**>Finchley.SR1</**version**>  <**type**>pom</**type**>  <**scope**>import</**scope**>  </**dependency**>  <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-alibaba-dependencies</**artifactId**>  <**version**>0.2.2.RELEASE</**version**>  <**type**>pom</**type**>  <**scope**>import</**scope**>  </**dependency**>  </**dependencies**> </**dependencyManagement**> |

#### Bootstrap配置内容

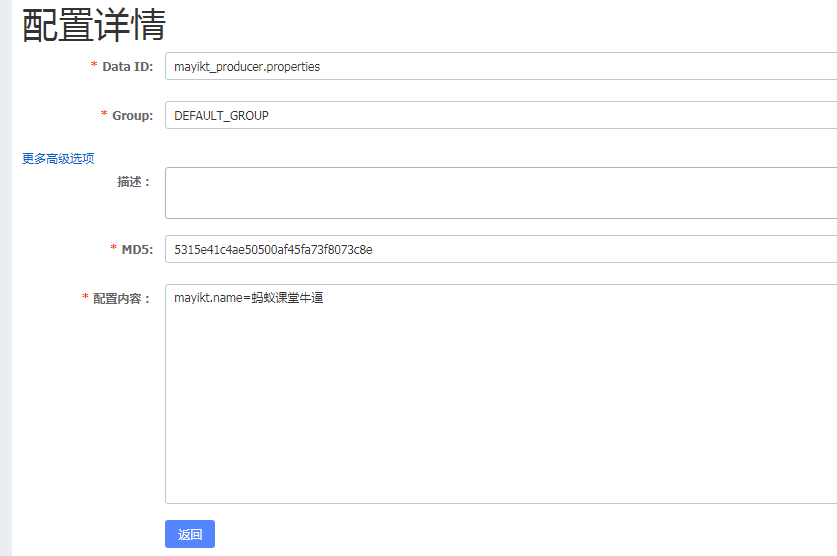
|  |
| --- |
| **spring**:  **cloud**:  **nacos**:  **discovery**:  **server-addr**: 127.0.0.1:8848  **application**:  **name**: mayikt\_consumer **server**:  **port**: 8085 |

#### 相关代码

|  |
| --- |
| */\*\*  \** ***@author*** *蚂蚁课堂创始人-余胜军QQ644064779  \** ***@title:*** *OrderConsumer  \** ***@description:*** *每特教育独创第五期互联网架构课程  \** ***@date*** *2019/7/819:51  \*/* @RestController @EnableDiscoveryClient @SpringBootApplication **public class** OrderConsumer {  @Autowired  **private** LoadBalancerClient **loadBalancerClient**;   @RequestMapping(**"/orderToMember"**)  **public** String orderToMember() {  *// 1.获取当前负载均衡集群节点地址* ServiceInstance mayiktProducer = **loadBalancerClient**.choose(**"mayikt\_producer"**);  String url = mayiktProducer.getUri() + **"/getMember"**;  RestTemplate restTemplate = **new** RestTemplate();  String result = restTemplate.getForObject(url, String.**class**);  **return** result;  }   **public static void** main(String[] args) {  SpringApplication.*run*(OrderConsumer.**class**);  } } |

## 基于Nacos构建分布式配置中心

创建 data id：一定要与服务注册名称相同，不然无法获取。



### Maven依赖信息

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

### 配置信息新增

|  |
| --- |
| **spring**:  **cloud**:  **nacos**:  **discovery**:  **server-addr**: 127.0.0.1:8848  **config**:  **server-addr**: 127.0.0.1:8848 |

### 相关代码

|  |
| --- |
| @SpringBootApplication @RestController @EnableDiscoveryClient @RefreshScope **public class** MemberApiService {   @Value(**"${mayikt.name}"**)  **private** String **name**;   @RequestMapping(**"/getMember"**)  **public** String getMember() {  **return "会员服务接口"** + **name**;  }   **public static void** main(String[] args) {  SpringApplication.*run*(MemberApiService.**class**);  } } |

SpringCloud 主要解决微服务中遇到的一些问题？ 分布式问题？

1. rpc远程调用服务治理的问题 eureka、zk、redis、consul nacos
2. 分布式一些解决方案

>>>>1.分布式任务调度平台 xxl-job

>>>>2.分布式事务问题（alibabaSpringCloud Seata）

>>>>3.分布式配置中心 （nacos实现）

误会 SpringCloud Alibaba 基于SpringCloud实现二次开发提高扩展组件和功能