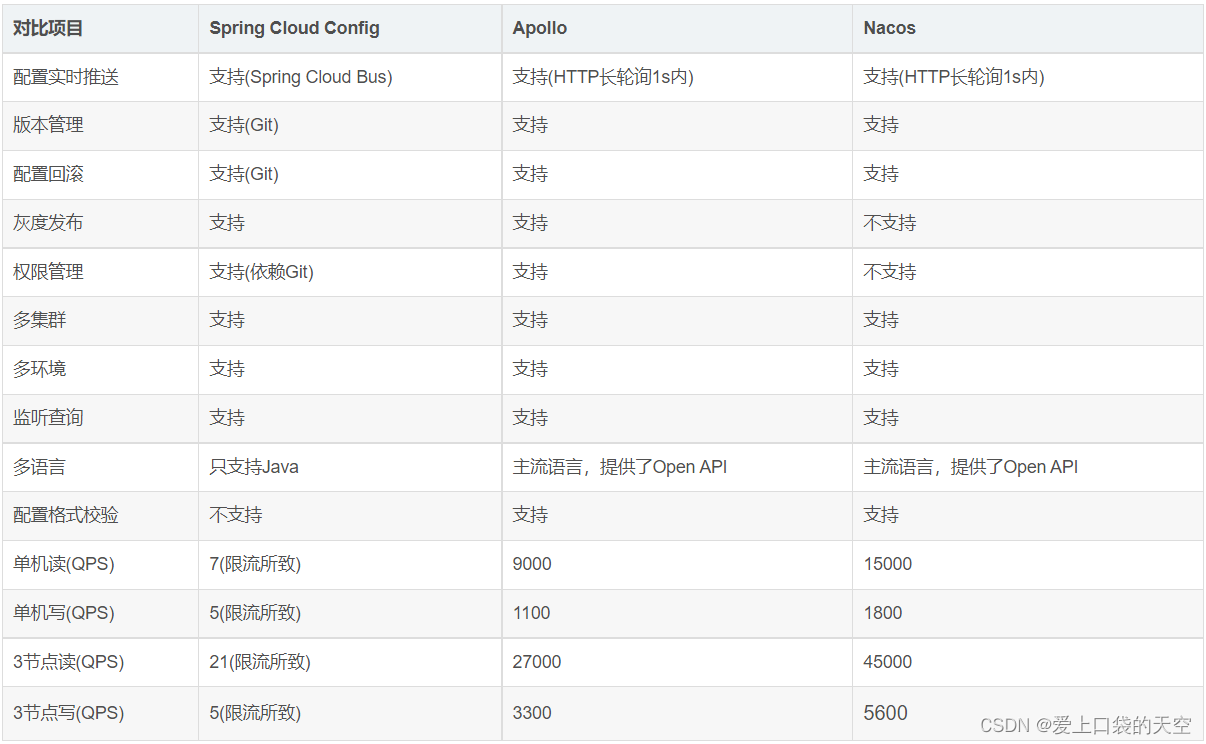
Nacos配置中心使用

官方文档： [https://github.com/alibaba/spring­cloud­alibaba/wiki/Nacos­config](https://github.com/alibaba/springcloudalibaba/wiki/Nacosconfig)

简介：Nacos 提供用于存储配置和其他元数据的 key/value 存储，为分布式系统中的外部化配置提供服务器端和客户端支持。使用 Spring Cloud Alibaba Nacos Config，您可以在 Nacos Server 集中管理你 Spring Cloud 应用的外部属性配置。

1. 优势对比

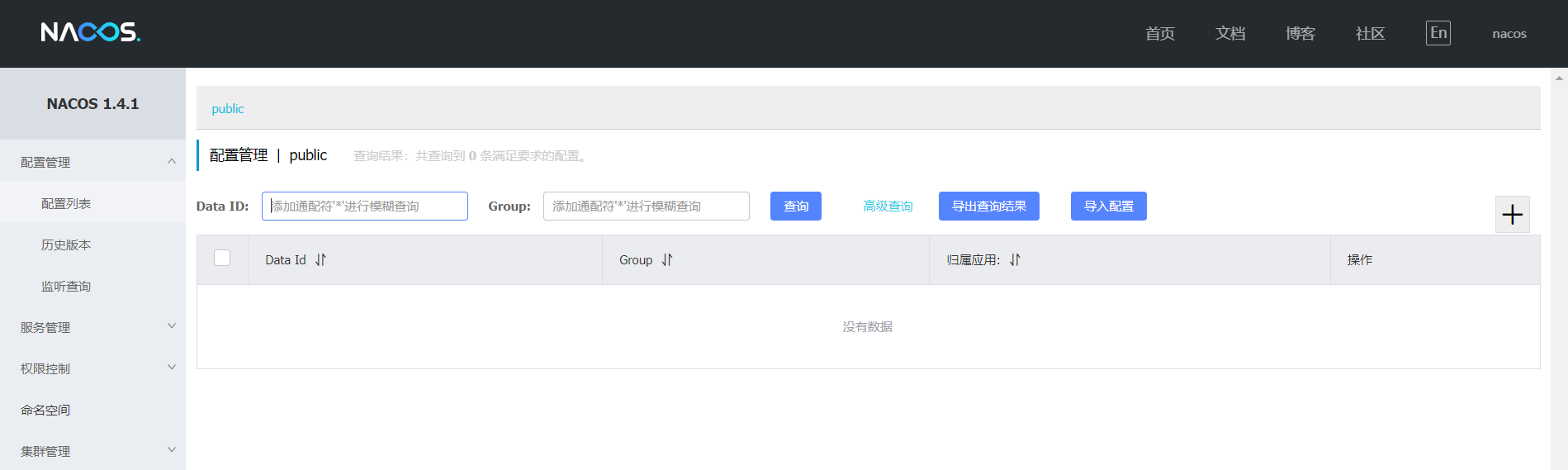


1. 快速开始（以windows环境下为例）

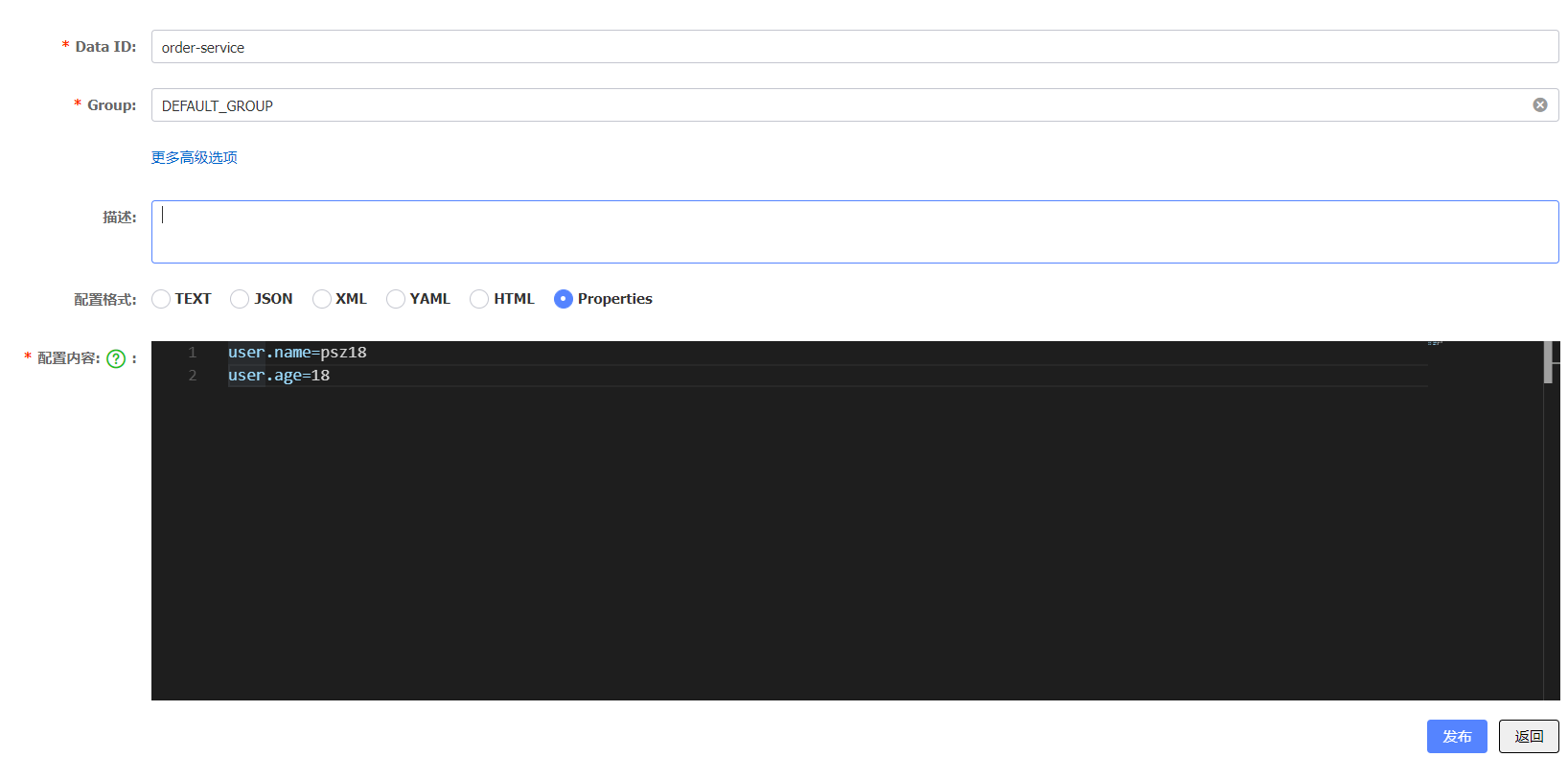
环境准备：64 bit OS，64 bit JDK 1.8+，Maven 3.2.x+

从 <https://github.com/alibaba/nacos/releases网站上下载Nacos.zip>（这里以1.4.1版本为例）并解压后，在bin目录下，sh startup.sh -m standalone 以单机模式启动。在浏览器中输入 <http://localhost:8848/nacos/#/login> 以打开Nacos登录页面，以nacos作为账号与密码进行登录。

登录后显示管理页面如下：



点击右侧“+”号按钮，新建一个配置



按照上图输入，点击“发布“按钮就会新增一个配置，并在列表中可以查得到



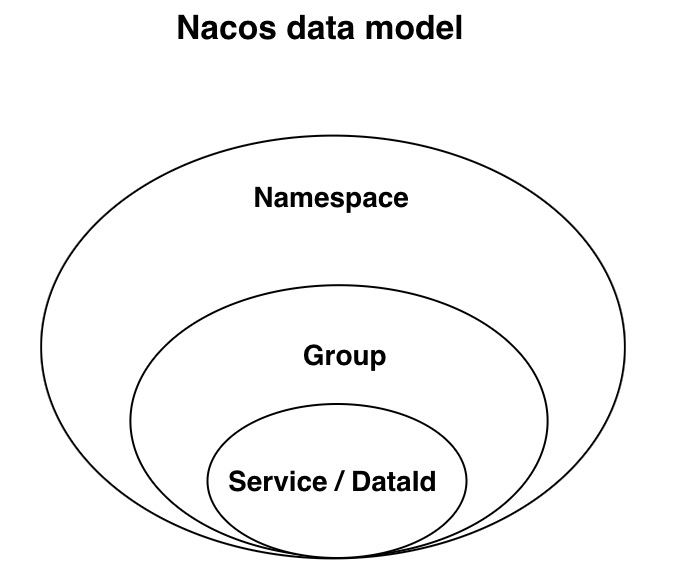
至此，我们在Nacos中成功新建了一个配置文件。该配置文件格式是.properties，Namespace是public，Group是DEFAULT\_GROUP，data Id是order-service。接下来就是将该配置信息配置到springCLoud工程中。

说明：

Namespace：代表不同环境，如开发、测试、生产环境。 nacos默认有一个保留的public的Namespace，但我们可以新建如dev、prod等命名空间。在那个命名空间下新建的配置就属于那个命名空间。

Group：代表某项目，如XX医疗项目、XX电商项目。新建配置时手动输入；

DataId：每个项目下往往有若干个工程（微服务），每个配置集(DataId)是一个工程（微服务）的主配置文件。新建配置时手动输入。



1. 配置Nacos-config服务

首先引入如下依赖：

<dependency>

<groupId>com.alibaba.cloud</groupId>

<artifactId>spring‐cloud‐starter‐alibaba‐nacos‐config</artifactId>

</dependency>

然后创建bootstrap.yml文件，并简单配置如下：

spring:

application:

name: order-service

cloud:

nacos:

config:

server-addr: 127.0.0.1:8848 #nacos服务器地址端口

file-extension: properties #要读取的配置文件拓展后缀名

# 如此就会获取`${spring.application.name}.${spring.cloud.nacos.config.file‐extension}`，即order-service. properties的配置文件。

程序中可以通过 applicationContext.getEnvironment().getProperty("user.name") 来获取user.name，并且是动态更新的。若[通过 @Value("${user.name}")](mailto:之后可以在程序中中通过@Value(%22$%7buser.name%7d%22)) 获取配置中的user.name，在Nacos中修改了配置后，不想重启项目而进行配置信息的更新，需要在对应的class上添加 @RefreshScope 注解。

至此，在Nacos管理页面中配置的order-service.properties文件信息就会配置到工程中。Nacos以这种方式同意管理配置。

1. 配置多文件以及优先级

Nacos加载${spring.application.name}.${spring.cloud.nacos.config.file‐extension}配置文件时，同样会加载${spring.application.name}-${profile}.${file-extension}配置文件。若bootstarp.yml中配置了 spring.profiles.active=dev，则就会加载order-service-dev. properties配置文件。可以依据这个来加载不同环境的配置文件。

通过自定义扩展的 Data Id 配置，既可以解决多个应用间配置共享的问题，又可以支持一个应用有多个配置文件。可以如下配置以使工程加载更多的配置文件

spring:

cloud:

nacos:

config:

server-addr: 127.0.0.1:8848

file-extension: properties

# 不同工程的通用配置 支持共享的 DataId

shared-configs:

- data-id: filename. extension

refresh: true

# 支持一个应用多个 DataId 的配置

extension-configs:

- data-id: filename. extension

refresh: true

namespace: public

Spring Cloud Alibaba Nacos Config 目前提供了三种配置能力从 Nacos 拉取相关的配置。

A: 通过 spring.cloud.nacos.config.shared-configs[n].data-id 支持多个共享 Data Id 的配置

B: 通过 spring.cloud.nacos.config.extension-configs[n].data-id 的方式支持多个扩展 Data Id 的配置

C: 通过内部相关规则(应用名、应用名+ Profile )自动生成相关的 Data Id 配置

当三种方式共同使用时，他们的一个优先级关系是:A < B < C