

[🏠 / 开发指南 / 微服务手册](#)[👤 芋道源码](#) [📅 2022-04-04](#)

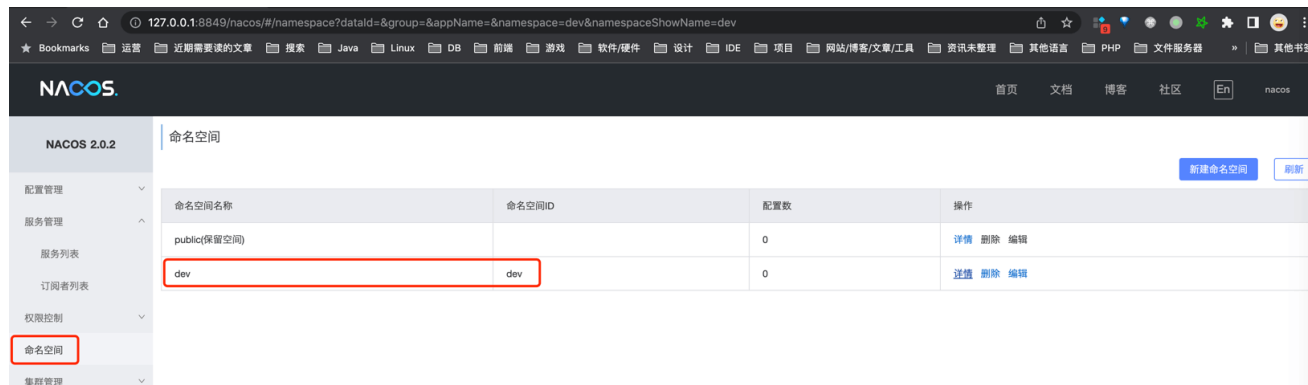
配置中心 Nacos

1. 配置中心 Nacos

项目使用 Nacos 作为配置中心，实现配置的动态管理。

1.1 搭建 Nacos Server

- ① 参考《芋道 Nacos 极简入门》[🔗](#) 文章的「2. 单机部署（最简模式）」或「3. 单机部署（基于 MySQL 数据库）」小节。
- ② 点击 Nacos 控制台的 [命名空间] 菜单，创建一个 ID 和名字都为 `dev` 的命名空间，稍后会使用到。如下图所示：



1.2 项目接入 Nacos

友情提示：以 `yudao-module-system` 服务为例子。

1.2.1 引入依赖

在 `yudao-module-system-biz` 模块的 `pom.xml` [🔗](#) 中，引入 Nacos 对应的依赖。如下所示：

```
<!-- Spring Cloud 基础 -->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-bootstrap</artifactId>
</dependency>

<!-- Config 配置中心相关 -->
```

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

1.2.2 添加配置

在 `bootstrap-local.yaml` 中，添加 `nacos.config` 配置。如下所示：

```
--- ##### 配置中心相关配置 #####

spring:
  cloud:
    nacos:
      # Nacos Config 配置项，对应 NacosConfigProperties 配置属性类
      config:
        server-addr: 127.0.0.1:8848 # Nacos 服务器地址
        namespace: dev # 命名空间。这里使用 dev 开发环境
        group: DEFAULT_GROUP # 使用的 Nacos 配置分组，默认为 DEFAULT_GROUP
        name: # 使用的 Nacos 配置集的 dataId，默认为 spring.application.name
        file-extension: yaml # 使用的 Nacos 配置集的 dataId 的文件拓展名，同时也是 N
```

- `spring.cloud.nacos.config.namespace` 配置项：设置为 `dev`，就是刚创建的命名空间

1.2.3 配置管理

① 参考《[芋道 Spring Cloud Alibaba 配置中心 Nacos 入门](#)》文档，学习 Nacos 配置中心的使用。

② 按照需要，将不同环境存在差异的 `application-local.yaml` 和 `application-dev.yaml` 中的配置，迁移到 Nacos 配置中心。

一般情况下，不建议将 `application.yaml` 中的配置，迁移到 Nacos 配置中心。因为 `application.yaml` 中的配置，是通用的配置，无需动态管理。

疑问：为什么项目中的 ``application-{env}.yaml`` 中的配置，没有放到 Nacos 配置中心中？

主要考虑大家《[快速启动](#)》可以更简单。

实际项目中，是建议放到 Nacos 配置中心，进行配置的动态管理的。

操作过程中，可能会碰到的问题：

- [IdTypeEnvironmentPostProcessor 与 Nacos 配置中心加载顺序问题](#)

2. 配置管理

友情提示：该功能是从 Boot 项目延用到 Cloud 项目，一般情况下不会使用到，使用 Nacos 管理配置即可。

在 [基础设施 -> 配置管理] 菜单，可以查看和管理配置，适合业务上需要动态的管理某个配置。

例如说：创建用户时，需要配置用户的默认密码，这个密码是不会变的，但是有时候需要修改这个默认密码，这个时候就可以通过配置管理来修改。

参数主键	参数分组	参数名称	参数键名	参数键值	系统内置	是否敏感	备注	创建时间	操作
1	ui	主框架页-默认...	sys.index.skinN...	skin-blue	系统内置	否	蓝色 skin-blue...	2021-01-05 17:03:48	修改 删除
2	biz	用户管理-账号...	sys.user.init-pa...	123456	系统内置	否	初始化密码 123...	2021-01-05 17:03:48	修改 删除
3	ui	主框架页-侧边...	sys.index.sideT...	theme-dark	系统内置	否	深色主题theme...	2021-01-05 17:03:48	修改 删除
4	1	xxx	demo.test	10	自定义	否	5	2021-01-19 03:10:26	修改 删除
5	xxx	xxx	xxx	xxx	自定义	是	xxx	2021-02-09 20:06:47	修改 删除
6	biz	登陆验证码的开关	yudao.captcha...	true	自定义	是		2022-02-17 00:03:11	修改 删除

对应的后端代码是 `yudao-module-infra` 的 `config` 业务模块。

2.1 配置的表结构

`infra_config` 的表结构如下：

```
CREATE TABLE `infra_config` (  
  `id` int NOT NULL AUTO_INCREMENT COMMENT '参数主键',  
  `group` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NUL  
  `type` tinyint NOT NULL COMMENT '参数类型',  
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NUL  
  `key` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `value` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NL  
  `sensitive` bit(1) NOT NULL COMMENT '是否敏感',  
  `remark` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAL  
  `creator` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAL  
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `updater` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAL  
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_  
  `deleted` bit(1) NOT NULL DEFAULT b'0' COMMENT '是否删除',
```

```
PRIMARY KEY (`id`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode
```

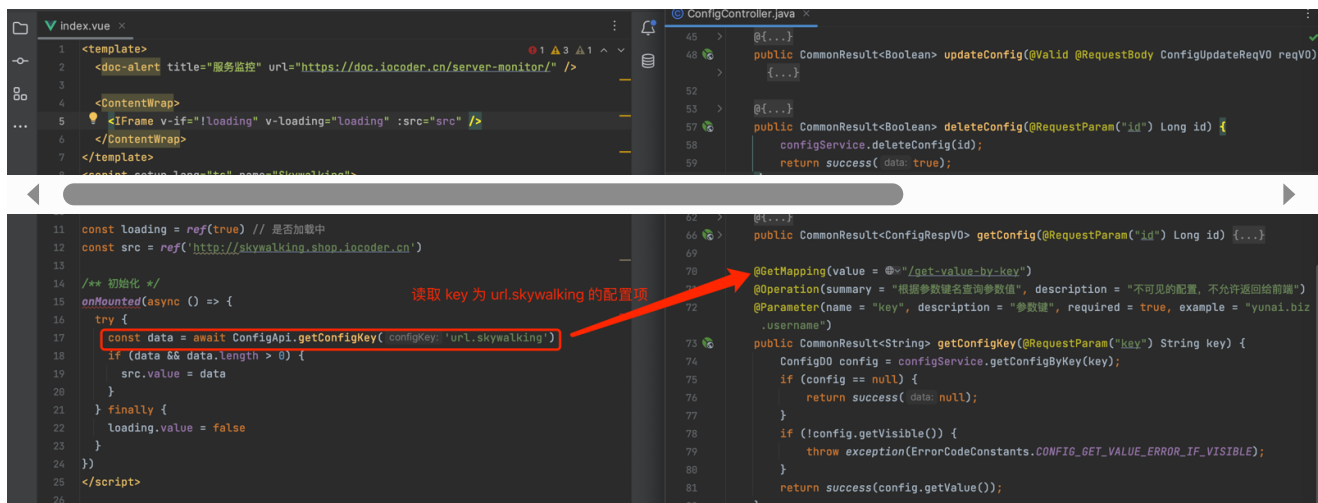
- key 字段，对应到 Spring Boot 配置文件的配置项，例如说 `yudao.captcha.enable` 、`sys.user.init-password` 等等。

2.2 后端案例

TODO 芋艿：待补充

2.3 前端案例

后端提供了 `/admin-api/infra/config/get-value-by-key` [🔗](#) RESTful API 接口，返回指定配置项的值。前端的使用示例如下图：



← 注册中心 Nacos

服务网关 Spring Cloud Gateway →



Theme by Vdoing | Copyright © 2019-2023 芋道源码 | MIT License