**nacos**

https://nacos.io/zh-cn/docs/quick-start-spring-boot.html

# 配置中心

## nacos安装包下载

<https://github.com/alibaba/nacos/releases>

下载zip或者tar.gz都可以

解压：

|  |
| --- |
| unzip nacos-server-$version.zip 或者 tar -xvf nacos-server-$version.tar.gz    cd nacos/bin |

## nacos启动

### Linux/Unix/Mac

启动命令(standalone代表着单机模式运行，非集群模式):

sh startup.sh -m standalone

如果您使用的是ubuntu系统，或者运行脚本报错提示[[符号找不到，可尝试如下运行：

bash startup.sh -m standalone

### Windows

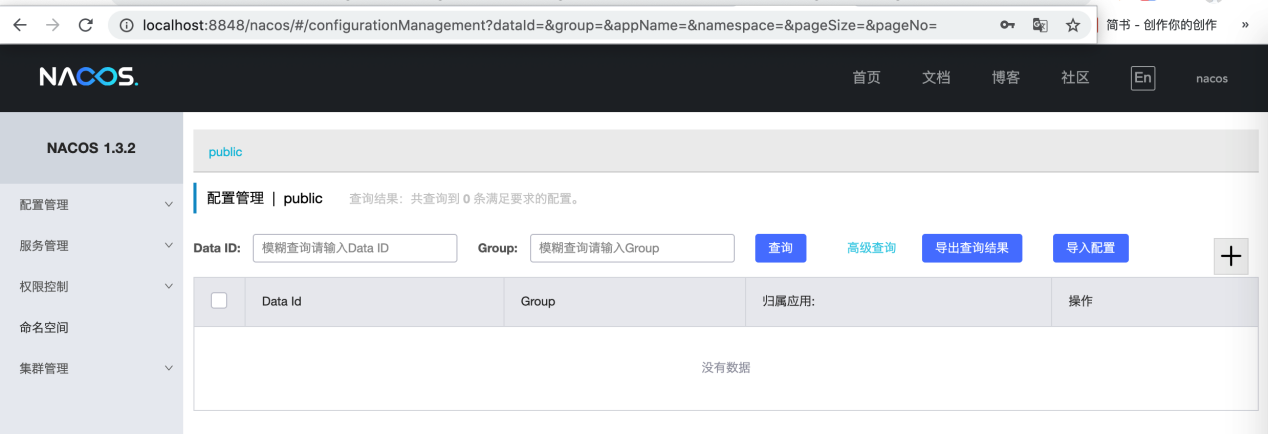
启动命令：

cmd startup.cmd

或者双击startup.cmd运行文件。

## nacos首页

<http://localhost:8848/nacos>



## nacos关闭

### Linux/Unix/Mac

sh shutdown.sh

### Windows

cmd shutdown.cmd

或者双击shutdown.cmd运行文件。

# springboot集成nacos配置中心

## 创建命名空间



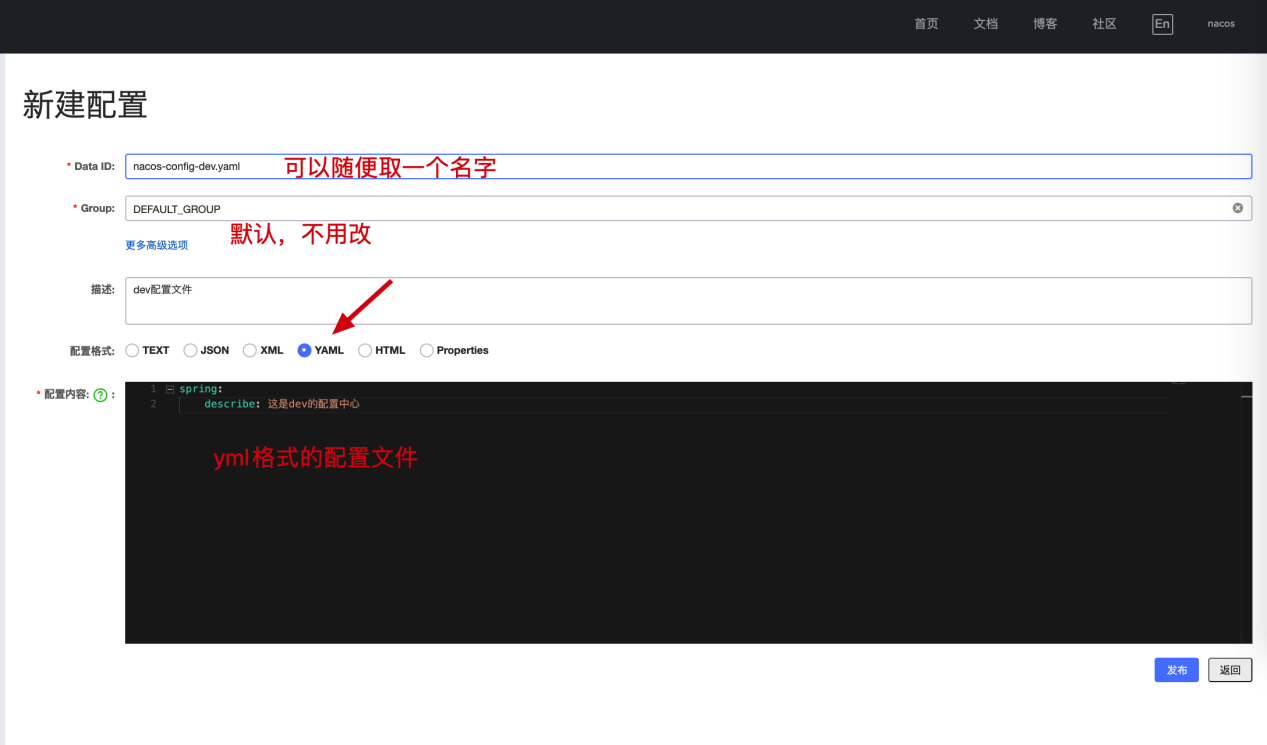
命名空间ID可以自己填，也可以系统自动生成，不能重复。

test命名空间创建方法同理。

## 创建配置列表

这里以dev和test环境为例，创建两个配置列表

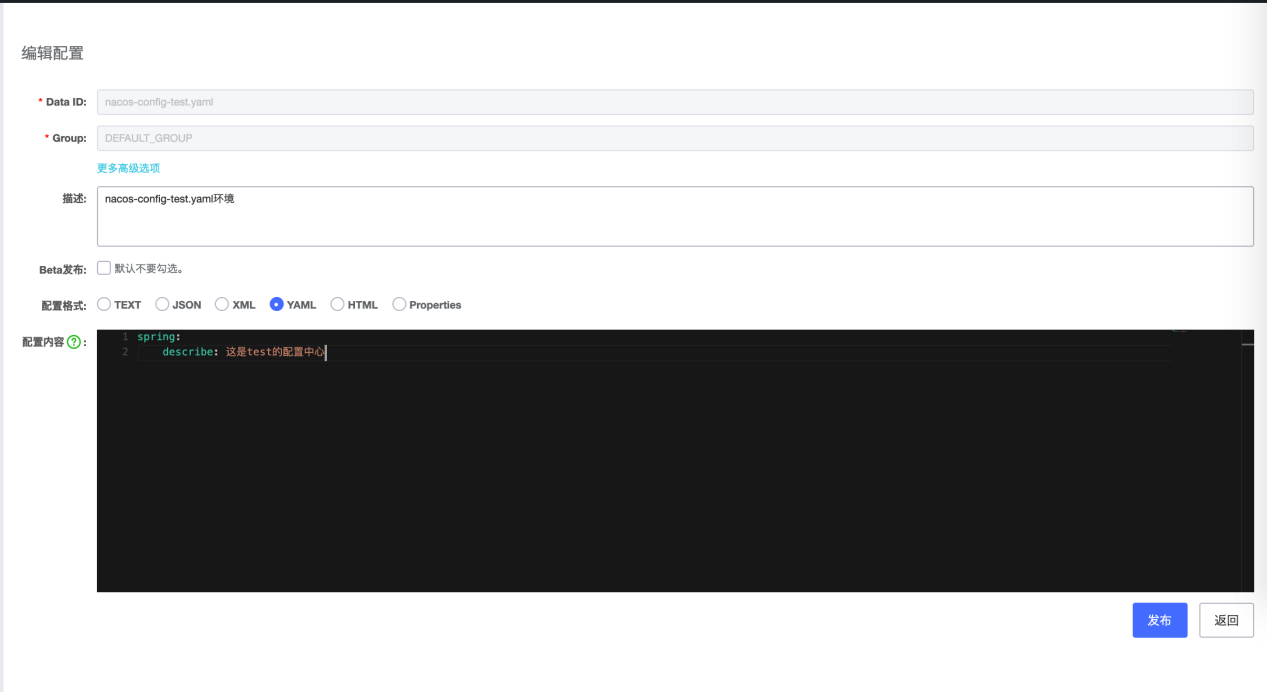




填写完成，点击发布



test配置中心同理：





好了，配置中心配置完成，去代码实现

## 新建springboot项目

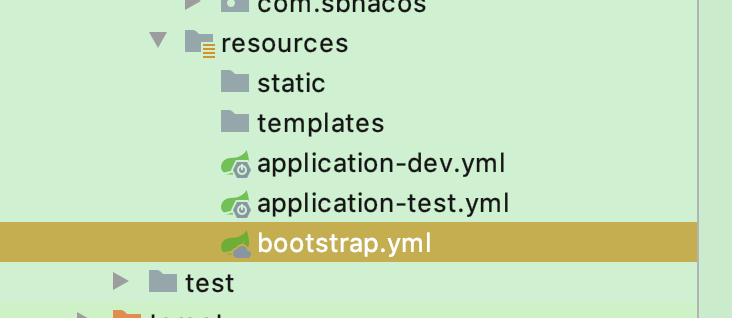
。。。

## 导入依赖

|  |
| --- |
| *<!-- 用于加载bootstrap.yml -->* <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-context</**artifactId**>  <**version**>2.2.3.RELEASE</**version**> </**dependency**>  *<!-- springboot 2.x对应0.2.x -->*  <**dependency**>  <**groupId**>com.alibaba.boot</**groupId**>  <**artifactId**>nacos-config-spring-boot-starter</**artifactId**>  <**version**>0.2.4</**version**> </**dependency**> |

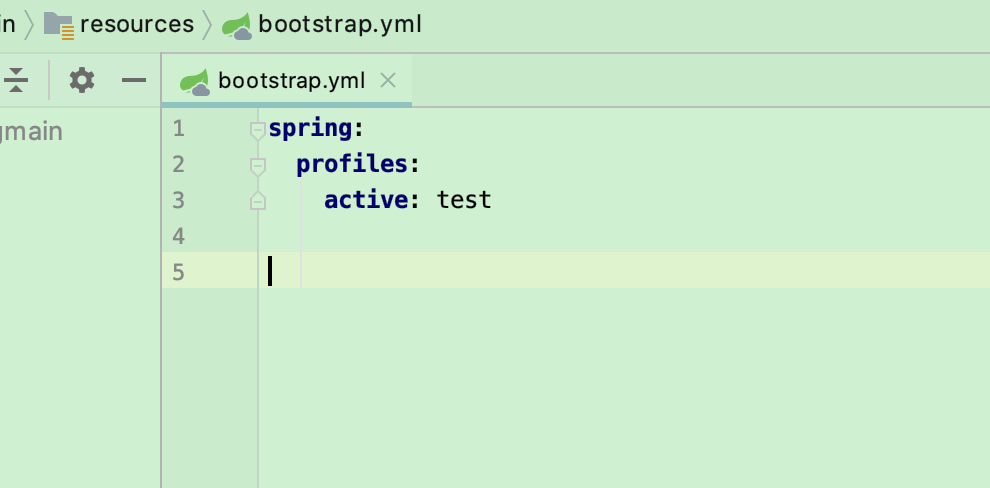
## 配置文件编写

创建三个配置文件bootstrap.yml application-dev.yml application-test.yml



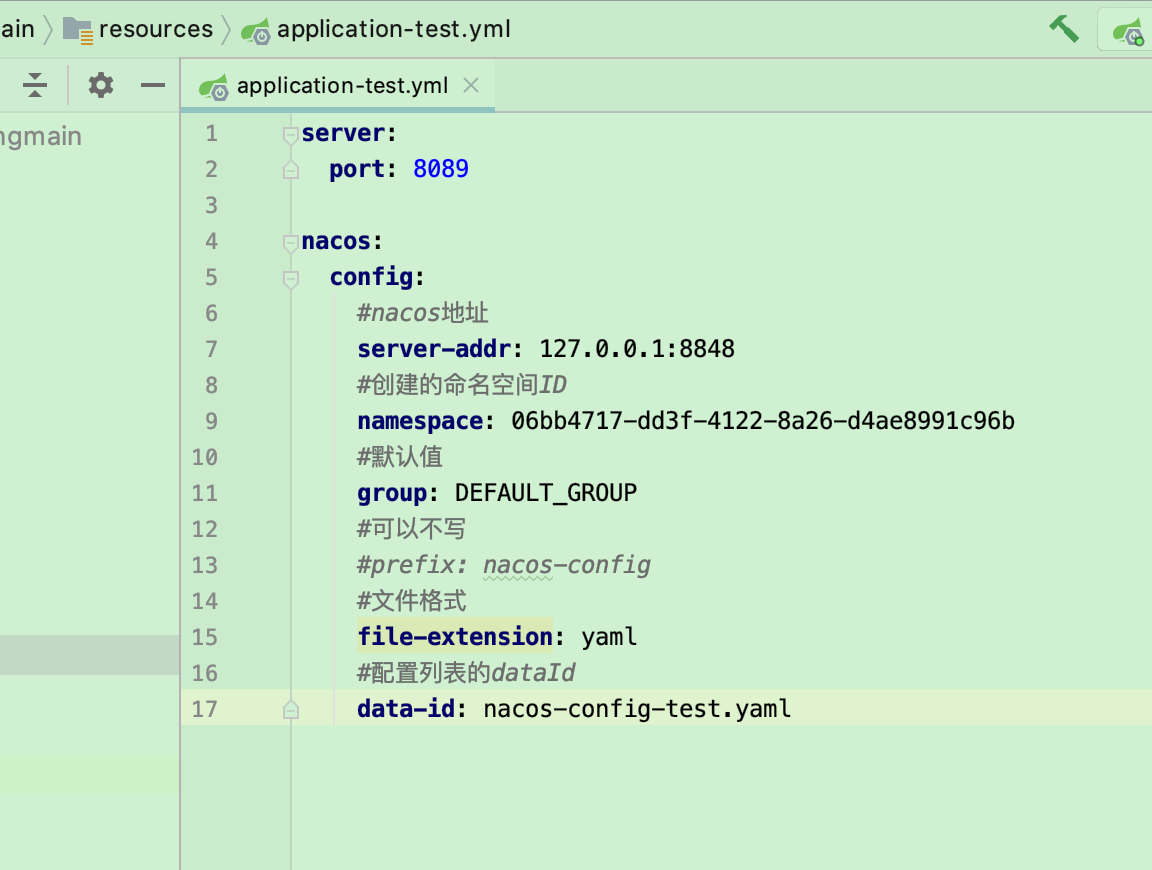
### bootstrap.yml

|  |
| --- |
| **spring**:  **profiles**:  **active**: test |



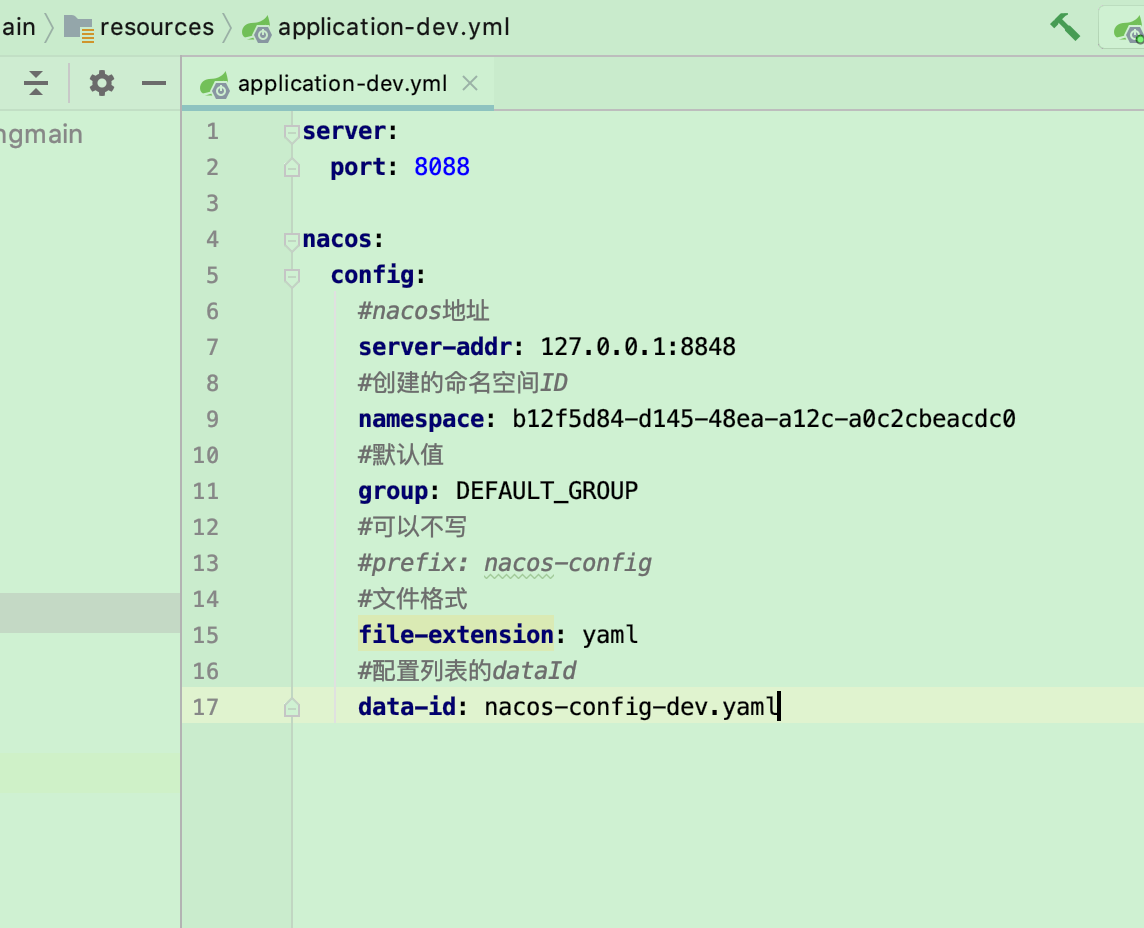
### application-test.yml

|  |
| --- |
| **server**:  **port**: 8089  **nacos**:  **config**:  *#nacos地址* **server-addr**: 127.0.0.1:8848  *#创建的命名空间ID* **namespace**: 06bb4717-dd3f-4122-8a26-d4ae8991c96b  *#默认值* **group**: DEFAULT\_GROUP  *#可以不写  #prefix: nacos-config  #文件格式* **file-extension**: yaml  *#配置列表的dataId* **data-id**: nacos-config-test.yaml |



### application-dev.yml

|  |
| --- |
| **server**:  **port**: 8088  **nacos**:  **config**:  *#nacos地址* **server-addr**: 127.0.0.1:8848  *#创建的命名空间ID* **namespace**: b12f5d84-d145-48ea-a12c-a0c2cbeacdc0  *#默认值* **group**: DEFAULT\_GROUP  *#可以不写  #prefix: nacos-config  #文件格式* **file-extension**: yaml  *#配置列表的dataId* **data-id**: nacos-config-dev.yaml |



## 代码编写

### application启动类

添加以下注解

|  |
| --- |
| @NacosPropertySource(dataId = **"${nacos.config.data-id}"**,autoRefreshed = **true**) |

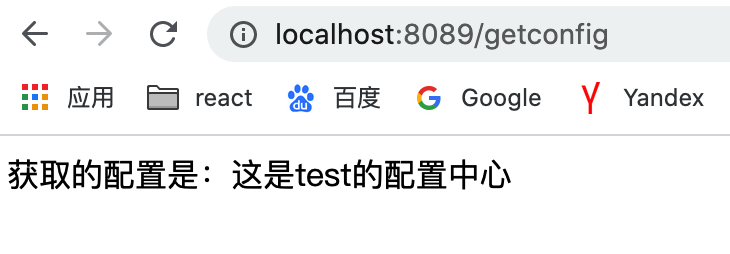
dataId对应配置列表中的dataId,根据打包环境的不同，通过${}引入配置文件中的不同的值

|  |
| --- |
| **package** com.sbnacos;  **import** com.alibaba.nacos.spring.context.annotation.config.NacosPropertySource; **import** org.springframework.boot.SpringApplication; **import** org.springframework.boot.autoconfigure.SpringBootApplication;  @NacosPropertySource(dataId = **"${nacos.config.data-id}"**,autoRefreshed = **true**) @SpringBootApplication **public class** SbNacosApplication {   **public static void** main(String[] args) {  SpringApplication.*run*(SbNacosApplication.**class**, args);  }  } |

### 添加接口测试

|  |
| --- |
| **package** com.sbnacos.controller;  **import** com.alibaba.nacos.api.config.annotation.NacosValue; **import** org.springframework.web.bind.annotation.GetMapping; **import** org.springframework.web.bind.annotation.RestController;  */\*\*  \** ***@author*** *iszengziqiang@163.com  \*/* @RestController **public class** NacosController {   @NacosValue(value = **"${spring.describe}"**, autoRefreshed = **true**)  **private** String **config**;   @GetMapping(**"/getconfig"**)  **public** String getSpringAbcFromNacos() {  **return "获取的配置是："** + **config**;  }  } |

test环境运行效果：



dev环境运行效果：

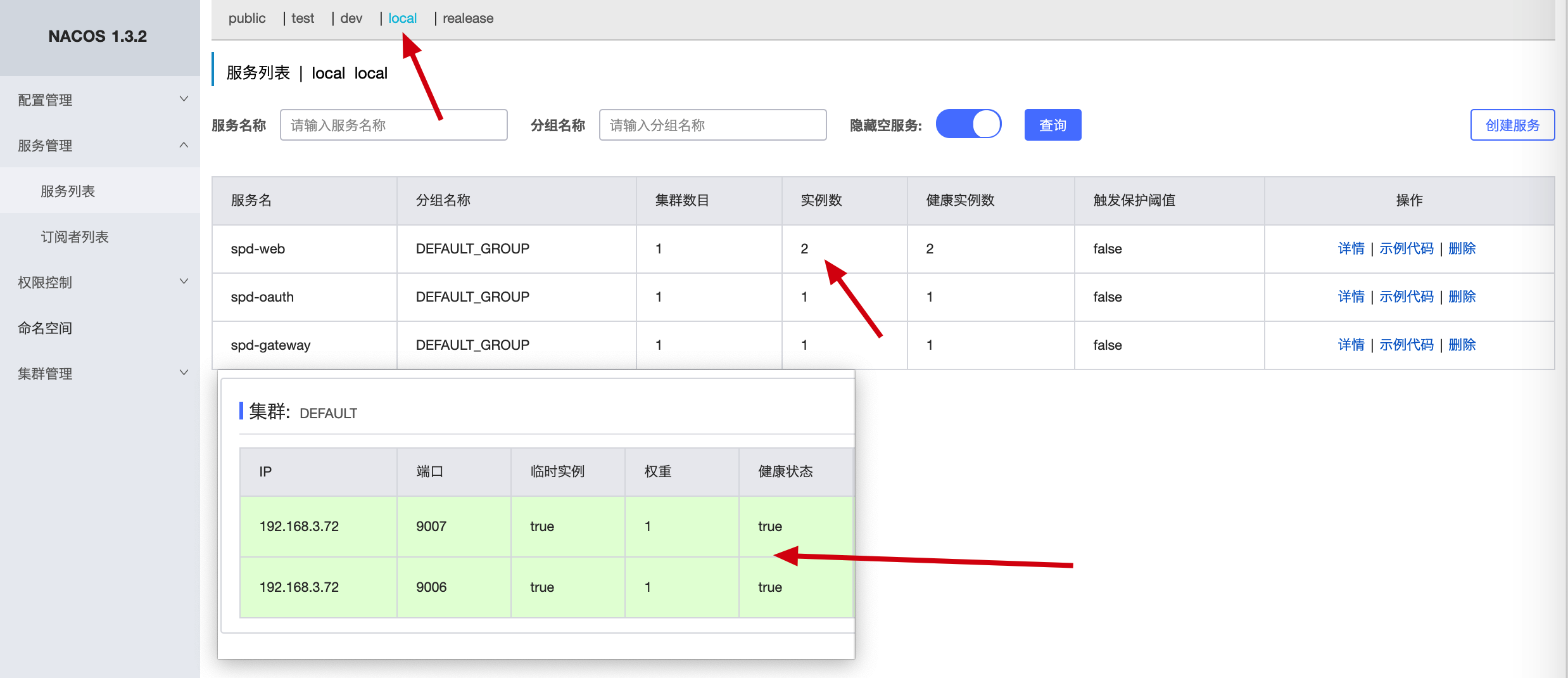


修改配置后，请求后值会发生改变。

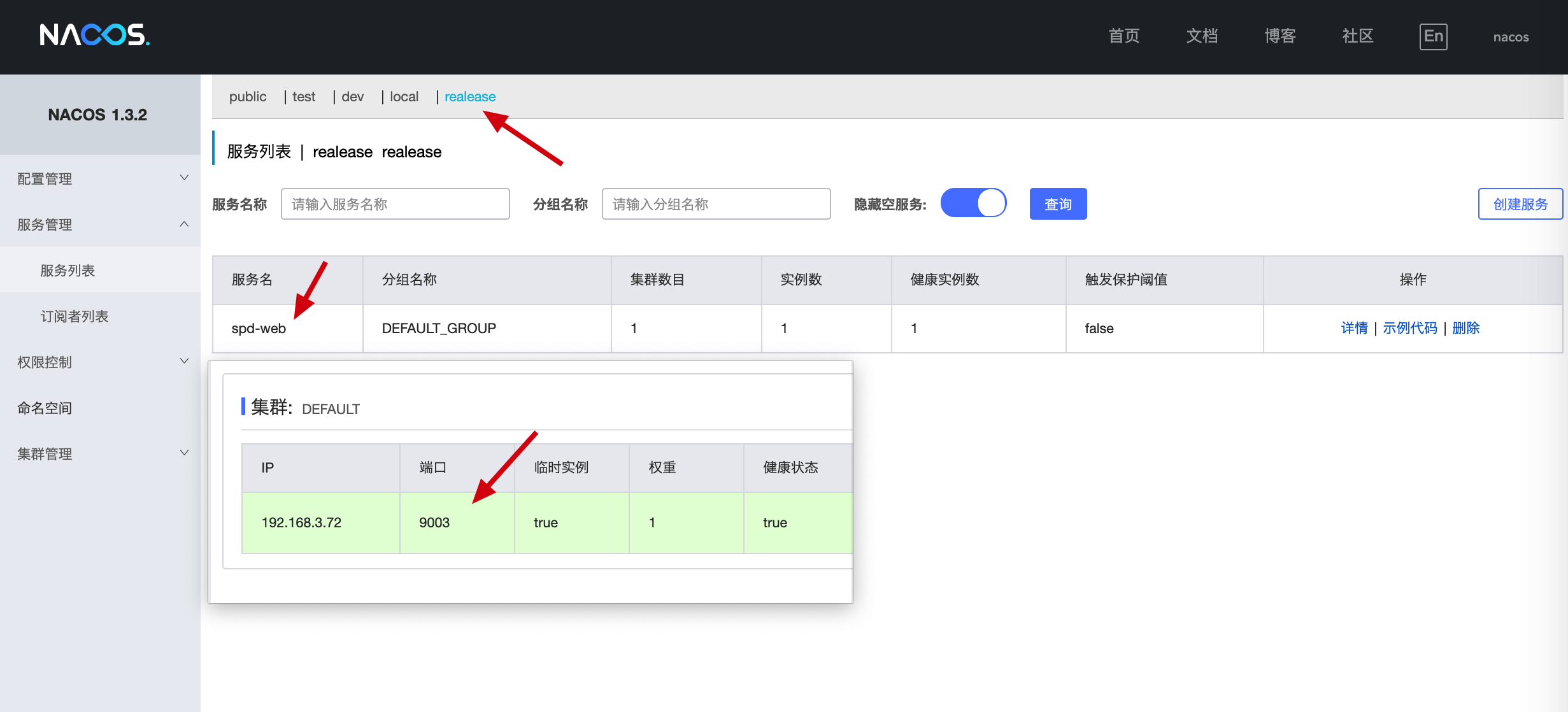
## nacos的命名空间

|  |
| --- |
| **spring**:  **cloud**:  **nacos**:  **discovery**:  **server-addr**: localhost:8848  **namespace**: local |

nacos集群的命名空间是有隔离功能的，需要发布在同一个命名空间才能访问



如图，在local命名空间里有spd-web两个服务，realease命名空间里也有一个spd-web服务，



spd-web服务里写了一个获取当前服务端口的方法

|  |
| --- |
| **package** com.bkgtsoft;  **import** com.bkgtsoft.spd.common.annotation.EnableLoginArgResolver; **import** org.springframework.beans.factory.annotation.Value; **import** org.springframework.boot.SpringApplication; **import** org.springframework.boot.autoconfigure.SpringBootApplication; **import** org.springframework.cloud.client.discovery.EnableDiscoveryClient; **import** org.springframework.cloud.openfeign.EnableFeignClients; **import** org.springframework.web.bind.annotation.GetMapping; **import** org.springframework.web.bind.annotation.RestController;  **import** java.util.Date;  @RestController @EnableFeignClients @EnableDiscoveryClient @EnableLoginArgResolver @SpringBootApplication(scanBasePackages = **"com.bkgtsoft"**) **public class** SpdWebApplication {   **public static void** main(String[] args) {  SpringApplication.*run*(SpdWebApplication.**class**, args);  }   @Value(**"${server.port}"**)  **private** String **port**;   @GetMapping(value = **"/abcd"**)  **public** String getPorts() {  System.***out***.println(**port**);  **return new** Date().toString() + **"===="** + **port**;  } } |

访问网关：

|  |
| --- |
| http://localhost:19001/spd-web/abcd |

只会返回9006和9007端口，轮询方式返回。

不会返回9003.停掉9006和9007后，9003也不会返回。说明是命名空间隔离的。