

## 18. Spring Cloud Alibaba

Spring Cloud Alibaba 是阿里巴巴提供的一套微服务开发一站式解决方案。

主要提供的功能：

- 服务限流降级
- 服务注册与发现
- 分布式配置中心
- 消息驱动
- 分布式事务

- 阿里云对象存储
- 阿里云短信

提供的组件：

- Sentinel
- Nacos

优势：

1. 中文文档
2. 没有另起炉灶，可以方便的集成到现有项目中
3. 阿里本身在高并发、高性能上的经验，让我们有理由相信这些组件足够可靠。

### 18.1 Nacos

Nacos 主要提供了服务发现、服务配置以及服务管理。

基本特性：

- 服务发现
- 动态配置
- 动态 DNS 服务
- 服务及元数据管理

#### 18.1 安装+配置中心

- Docker 安装
- 下载源码自己编译安装/下载编译好的安装包

首先下载安装包：<https://github.com/alibaba/nacos/releases/download/1.2.0-beta.1/nacos-server-1.2.0-beta.1.tar.gz>

然后解压，解压后，如果 win，直接在 bin 目录下双击 startup.cmd 启动。

如果Linux，bin 目录下执行 sh startup.sh -m standalone.

**注意，系统一定要配置好 jdk，测试一下 java 和 javac 两个命令要存在。**

Nacos 启动成功后，浏览器输入：<http://192.168.91.128:8848/nacos> 就能看到启动页面。

如果有登录页面，登录的默认用户名/密码都是 nacos。

先来看配置中心，类似于 Spring Cloud Config

首先在服务端配置，点击配置管理->配置列表->+

这里主要配置三个东西，Data ID、Group 以及要配置的内容。

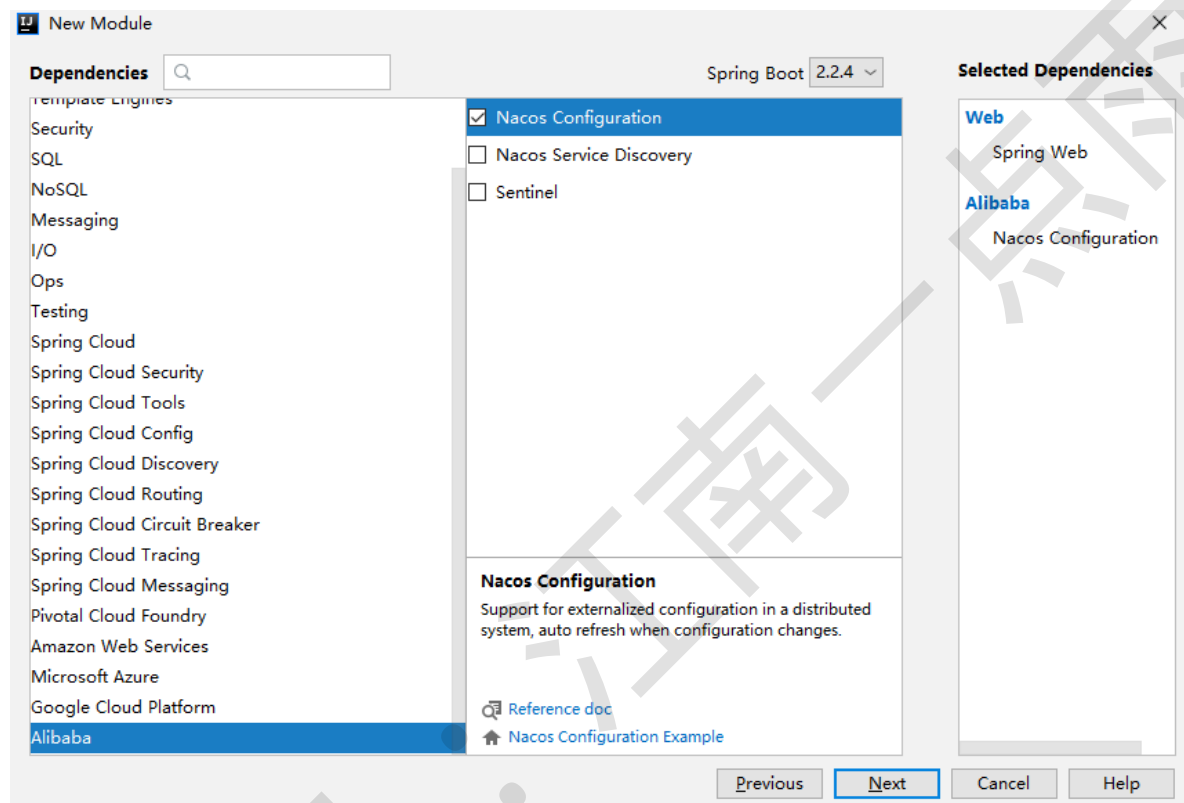
Data Id 的格式是 `${prefix}-${spring.profile.active}.${file-extension}`

`${prefix}` 的值，默认为 `spring.application.name` 的值

`${spring.profile.active}` 表示项目当前所处的和环境

`${file-extension}` 表示配置文件的扩展名

配置完成后，新建 Spring Boot 项目，加入 Nacos 依赖：



然后，新建 `bootstrap.properties` 配置文件，配置 nacos 信息：

```
spring.application.name=nacos
spring.cloud.nacos.server-addr=192.168.91.128:8848
spring.cloud.nacos.config.file-extension=properties
```

最后再提供一个测试 Controller：

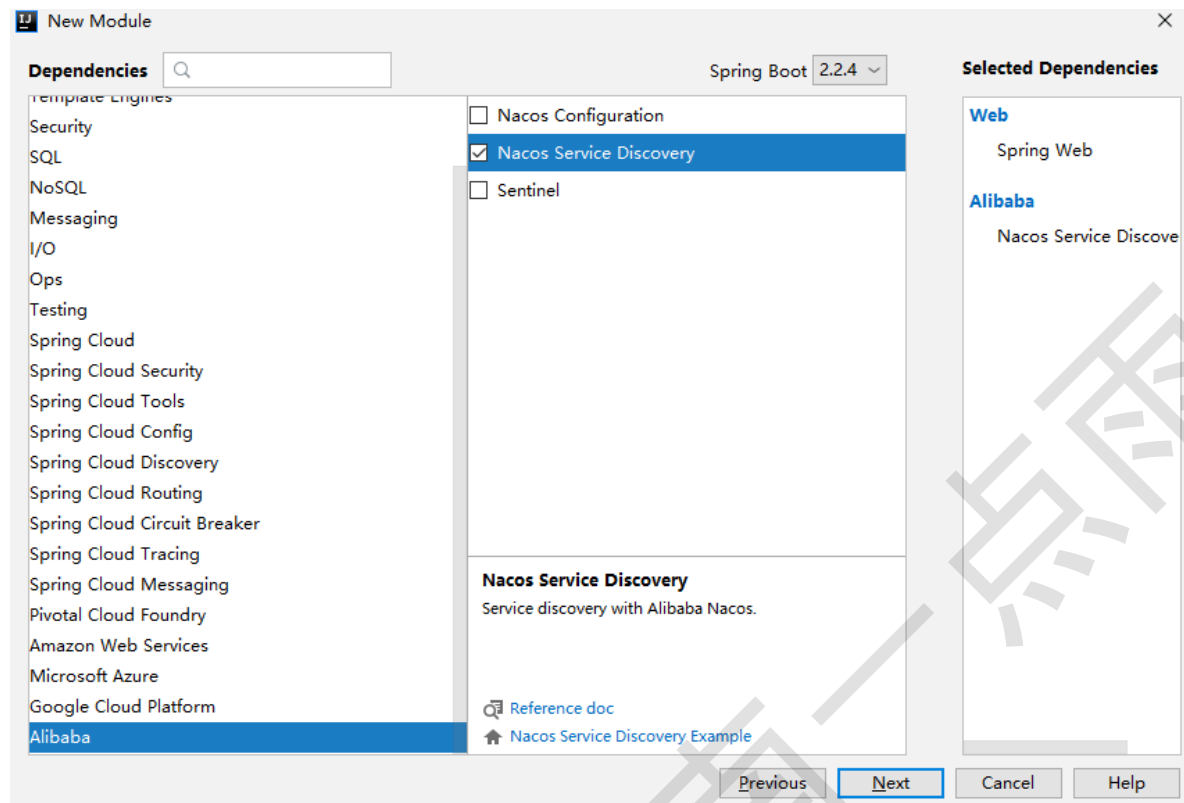
```
@RestController
@RefreshScope
public class HelloController {
    @Value("${name}")
    String name;

    @GetMapping("/hello")
    public String hello() {
        return name;
    }
}
```

## 18.2 注册中心

Nacos 做注册中心，可以代替 Eureka。

创建 Spring Boot 项目，添加依赖：



添加配置：

```
spring.application.name=nacos01
spring.cloud.nacos.discovery.server-addr=192.168.91.128:8848
```

然后再提供一个测试 Controller：

```
@RestController
public class HelloController {
    @Value("${server.port}")
    Integer port;
    @GetMapping("/hello")
    public String hello() {
        return "hello:" + port;
    }
}
```

再将项目打包，启动两个实例。

```
java -jar nacos01-0.0.1-SNAPSHOT.jar --server.port=8080
java -jar nacos01-0.0.1-SNAPSHOT.jar --server.port=8081
```

再来创建一个 consumer，配置基本与 nacos01 一致。

## 18.2 Sentinel

- 使用场景丰富
- 有完备的实时监控
- 广泛的开源生态

Sentinel 整体上可以分为两个核心部分：

- 核心库
- 控制台

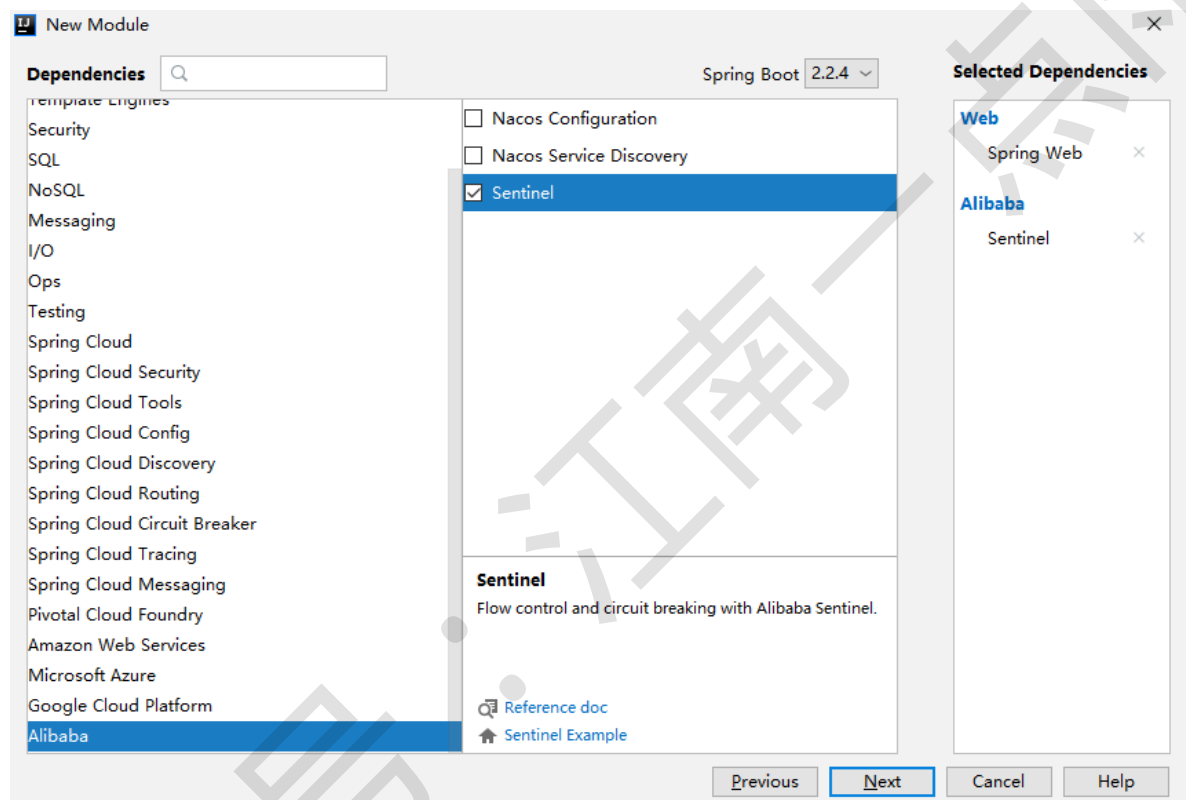
## 18.2.1 安装

首先下载控制台 jar，这是一个 Spring Boot 工程，下载好之后，直接用 Spring Boot 启动命令启动它即可。

下载地址：<https://github.com/alibaba/Sentinel/releases/download/1.7.1/sentinel-dashboard-1.7.1.jar>

## 18.2.2 应用

创建一个新的 Spring Boot 工程，添加 Sentinel 依赖。



项目创建成功后，配置 sentinel 控制台地址：

```
spring.application.name=sentinel
spring.cloud.sentinel.transport.dashboard=localhost:8080

server.port=8081
```

再创建一个测试 Controller。