

## Spring Boot Admin的诞生

Spring Boot Actuator提供了对单个Spring Boot的监控，信息包含：应用状态、内存、线程、堆栈等等，比较全面的监控了Spring Boot应用的整个生命周期。

但是这样监控也有一些问题：

第一，所有的监控都需要调用固定的接口来查看，如果全面查看应用状态需要调用很多接口，并且接口返回的Json信息不方便运营人员理解；

第二，如果Spring Boot应用集群非常大，每个应用都需要调用不同的接口来查看监控信息，操作非常繁琐低效。

在这样的背景下，就诞生了另外一个开源软件：**Spring Boot Admin**

## 什么是Spring Boot Admin?

Spring Boot Admin 是一个管理和监控Spring Boot 应用程序的开源软件。每个应用都认为是一个客户端，通过HTTP或者使用 Eureka注册到admin server中进行展示，Spring Boot Admin UI部分使用AngularJs将数据展示在前端。

Spring Boot Admin 是一个针对spring-boot的actuator接口进行UI美化封装的监控工具。他可以：在列表中浏览所有被监控spring-boot项目的基本信息，详细的Health信息、内存信息、JVM信息、垃圾回收信息、各种配置信息（比如数据源、缓存列表和命中率）等，还可以直接修改logger的level。

## 快速入门

server端（新建spring boot项目）

### 1 引入依赖

1. `<dependency>`
2. `<groupId>de.codecentric</groupId>`

```

3.     <artifactId>spring-boot-admin-server</artifactId>
4.     <version>1.5.6</version>
5. </dependency>
6. <dependency>
7.     <groupId>de.codecentric</groupId>
8.     <artifactId>spring-boot-admin-server-ui</artifactId>
9.     <version>1.5.6</version>
10. </dependency>

```

## 2 配置配置文件

```
server.port=8000
```

## 3 springboot 启动类

```

1. @Configuration
2. @EnableAutoConfiguration
3. @EnableAdminServer
4. public class SpringbootAdminServerApplication {
5.
6.     public static void main(String[] args) {
7.
8.         SpringApplication.run(SpringbootAdminServerApplication.class,
9.             args);
10.    }

```

# client端（新建springboot 项目）

## 1 引入依赖

```

1. <dependency>
2.     <groupId>de.codecentric</groupId>
3.     <artifactId>spring-boot-admin-starter-client</artifactId>
4.     <version>1.5.6</version>
5. </dependency>

```

## 2 配置配置文件

```

server.port=8001

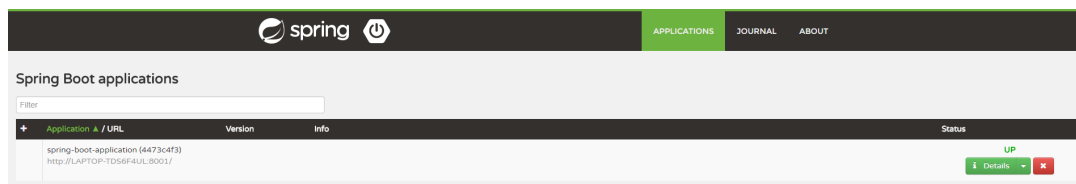
#配置Admin Server的地址
spring.boot.admin.url=http://127.0.0.1:8000
#关闭安全验证
management.security.enabled=false

```

## 测试

### 1 分别启动服务端与客户端

### 2 <http://127.0.0.1:8000> 浏览器访问服务的



如图：可以发现可以看到客户端已经连接上了

更多参考官网：<http://codecentric.github.io/spring-boot-admin/1.5.7/#securing-spring-boot-admin>