

Spring Boot Admin

1. 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。

2 监控单体应用

要使用 springboot-admin 需要一个 server 端用于显示监控,所以首先配置一个服务端

2.1 配置SpringBoot Server

2.1.1 pom 文件

```
<dependencies>

    <!-- https://mvnrepository.com/artifact/de.codecentric/spring-boot-
admin-server -->
    <dependency>
        <groupId>de.codecentric</groupId>
        <artifactId>spring-boot-admin-server</artifactId>
        <version>1.5.7</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/de.codecentric/spring-boot-
admin-server-ui -->
    <dependency>
        <groupId>de.codecentric</groupId>
        <artifactId>spring-boot-admin-server-ui</artifactId>
        <version>1.5.7</version>
    </dependency>
</dependencies>
```

2.1.2 application.yml

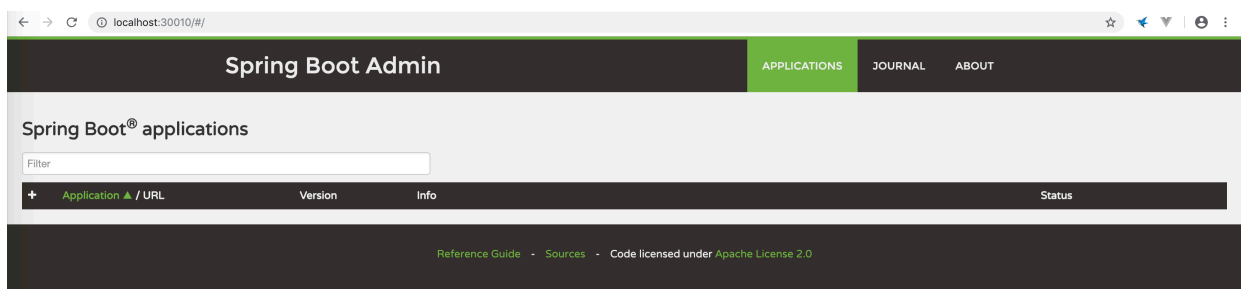
```
server:
  port: 30010 #主要用于访问显示的
```

2.1.3 主程序

```
@SpringBootApplication
@EnableAdminServer //开启 admin server
public class AdminStarter {
    public static void main (String[] args){
        SpringApplication.run(AdminStarter.class,args);
    }
}
```

2.1.4 启动

启动后访问<http://localhost:30010/>



2.2 配置 springboot client

我们只要将我们的普通的 springboot 程序中添加 springboot admin client 即可,然后就可以通过 server 查看了,以下内容只是在原始项目基础上添加

2.2.1 pom.xml

在项目的 pom 文件中添加依赖依赖

```
<dependency>
  <groupId>de.codecentric</groupId>
  <artifactId>spring-boot-admin-starter-client</artifactId>
  <version>1.5.7</version>
</dependency>
```

2.2.2 application.yml

在里面添加 admin server 地址即可

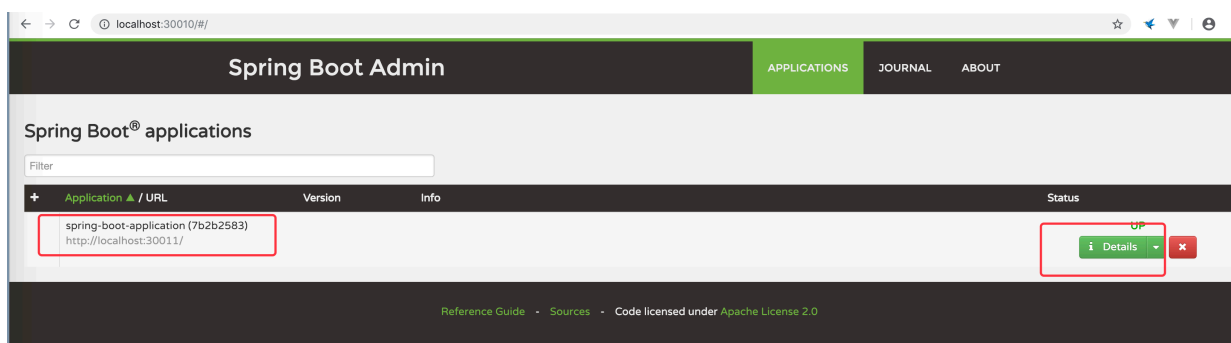
```
server:
  port: 30010 #项目端口
spring:
  boot:
    admin:
      url: http://localhost:30010 #添加 server 的地址
```

2.2.3 主程序

```
//这个为了演示而写的主程序,实际开发中直接启动自己的主程序就可以
@SpringBootApplication
public class AdminClientApplication {
    public static void main(String[] args) {
        SpringApplication.run(AdminClientApplication.class, args);
    }
}
```

2.2.4 启动访问

启动 client后重新访问 server,可以在里面看到我们刚才启动的程序,可以查看 detail



3 监听微服务(eureka 中拉取)

开发中,我们的微服务都是 springboot 项目,我们想查看这些程序的话可以通过让 server 去 eureka 上面拉取我们所有的程序即可,此处略过 eureka配置,需要配置 eureka server

3.1 配置 server

我们只需要让 server注册到 eureka 即可

3.1.1 pom.xml

```

<dependencies>

    <!-- https://mvnrepository.com/artifact/de.codecentric/spring-boot-
admin-server -->
    <dependency>
        <groupId>de.codecentric</groupId>
        <artifactId>spring-boot-admin-server</artifactId>
        <version>1.5.7</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/de.codecentric/spring-boot-
admin-server-ui -->
    <dependency>
        <groupId>de.codecentric</groupId>
        <artifactId>spring-boot-admin-server-ui</artifactId>
        <version>1.5.7</version>
    </dependency>
    <dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-netflix-eureka-
client</artifactId>
    </dependency>

</dependencies>

```

3.1.2 application.yml

```

server:
  port: 30010
eureka:
  client:
    service-url:
      defaultZone: http://user:123@localhost:10000/eureka #配置 eureka
spring:
  application:
    name: springbootadminserver

```

3.1.3 主程序

```

@SpringBootApplication
@EnableAdminServer
@EnableEurekaClient #注册到 eureka,用于从 eureka 中拉取程序列表
public class AdminStarter {
    public static void main (String[] args){
        SpringApplication.run(AdminStarter.class,args);
    }
}

```

3.1.4 启动 server

启动后配置 client 即可

3.2 配置 client

我们将我们所有需要被 springboot admin 查看的程序也注册到 eureka 即可

3.2.1 pom 文件

在 pom 中添加 eureka

```

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-
client</artifactId>
</dependency>

```

3.2.2 配置 application.yml

在这里,需要把 eureka 地址配置,不需要在像单体应用那样配置 admin server 的地址

```

server:
  port: 30011
eureka:
  client:
    service-url:
      defaultZone: http://user:123@localhost:10000/eureka #配置 eureka 地址,需要提
前搭建
spring:
  application:
    name: springbootadminclient

```

3.2.3 主程序

```

@SpringBootApplication
@EnableEurekaClient //注册到 eureka,可以被 admin server 发现
public class AdminClientStarter {
    public static void main (String[] args){
        SpringApplication.run(AdminClientStarter.class,args);
    }
}

```

3.2.4 启动

启动后访问 server ui,发现可以看到在 eureka 中所有的服务,包括 admin server 这个程序

The screenshot shows the Spring Eureka web interface at localhost:10000. The top navigation bar includes 'HOME' and 'LAST 1000 SINCE STARTUP'. The main content area is divided into two sections: 'System Status' and 'DS Replicas'.

System Status

Environment	test	Current time	2019-01-08T22:58:40 +0800
Data center	default	Uptime	00:01
		Lease expiration enabled	false
		Renews threshold	6
		Renews (last min)	2

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
SPRINGBOOTADMINCLIENT	n/a (1)	(1)	UP (1) - 192.168.3.87:springbootadminclient:30011
SPRINGBOOTADMINSERVER	n/a (1)	(1)	UP (1) - 192.168.3.87:springbootadminserver:30010

General Info

Name	Value
total-avail-memory	479mb
environment	test
num-of-cpus	8
current-memory-usage	108mb (22%)
server-uptime	00:01
registered-replicas	
unavailable-replicas	
available-replicas	

Eureka 中的服务列表

The screenshot shows the Spring Boot Admin web interface at localhost:30010/#/. The top navigation bar includes 'APPLICATIONS', 'JOURNAL', and 'ABOUT'. The main content area is titled 'Spring Boot® applications' and features a filter input field.

Spring Boot® applications

Application / URL	Version	Info	Status
SPRINGBOOTADMINCLIENT (2cb2727e) http://192.168.3.87:30011/			UP Details
SPRINGBOOTADMINSERVER (d00e346d) http://192.168.3.87:30010/			UP Details

Admin server 中的服务

Reference Guide - Sources - Code licensed under Apache License 2.0