Spring boot admin 的监控使用说明

# 1 前言

Spring Boot Admin 通过Srping-boot-stater-actuator提供的rest接口实现了图形化的监控页面，包括应用的配置信息，Bean信息，环境属性和线程信息、JVM状况等。

Spring Boot Admin分为服务端和客户端。客户端通过HTTP向服务端提供自身信息，服务端收集这些信息并以图形化界面的方式呈现。下面，Spring Boot Admin客户端简称为SBA客户端，Spring Boot Admin服务端简称为SBA服务端，Spring Boot Admin 就是将 Spring Boot Actuator中提供的endpoint信息可视化表示，并且可以通过邮件、Telegram，Hipchat等发送告警信息。

# 2 集成

注意一定要版本对应，否则会出现意想不到的问题，建议使用Srping Boot Admin 2.0+以上版本，可以多语言切换。

## 2.1 父项目

<modules>

<module>admin-server</module>

<module>admin-client</module>

</modules>

## 2.2 监控服务端

pom.xml 引入：

<artifactId>admin-server</artifactId>

<dependencies>

<dependency>

<groupId>de.codecentric</groupId>

<artifactId>spring-boot-admin-starter-server</artifactId>

<version>2.2.1</version>

</dependency>

<!--登录认证-->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<!--掉线发送邮件通知-->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-mail</artifactId>

</dependency>

</dependencies>

## 2.3 application.properties配置文件

server.port=9000

spring.application.name=SpringBootAdmin

spring.security.user.name=admin

spring.security.user.password=admin

spring.boot.admin.monitor.status-interval = 10000

spring.boot.admin.monitor.info-interval = 10000

spring.mail.host = smtp.163.com

spring.mail.username = 12345678@163.com

spring.mail.password = 123456

spring.boot.admin.notify.mail.from = 12345678@163.com

spring.boot.admin.notify.mail.to = 12345678@qq.com

## 2.4 启动类

@Configuration

@EnableAutoConfiguration

@EnableAdminServer

public class Application {

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

@Configuration

public static class SecuritySecureConfig extends WebSecurityConfigurerAdapter {

private final AdminServerProperties adminServer;

public SecuritySecureConfig(AdminServerProperties adminServer) {

this.adminServer = adminServer;

}

@Override

protected void configure(HttpSecurity http) throws Exception {

SavedRequestAwareAuthenticationSuccessHandler successHandler = new SavedRequestAwareAuthenticationSuccessHandler();

successHandler.setTargetUrlParameter("redirectTo");

successHandler.setDefaultTargetUrl(this.adminServer.path("/"));

http.authorizeRequests()

.antMatchers(this.adminServer.path("/assets/\*\*")).permitAll()

.antMatchers(this.adminServer.path("/login")).permitAll()

.anyRequest().authenticated()

.and()

.formLogin().loginPage(this.adminServer.path("/login")).successHandler(successHandler).and()

.logout().logoutUrl(this.adminServer.path("/logout")).and()

.httpBasic().and()

.csrf()

.csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse())

.ignoringRequestMatchers(

new AntPathRequestMatcher(this.adminServer.path("/instances"), HttpMethod.POST.toString()),

new AntPathRequestMatcher(this.adminServer.path("/instances/\*"), HttpMethod.DELETE.toString()),

new AntPathRequestMatcher(this.adminServer.path("/actuator/\*\*"))

)

.and()

.rememberMe().key(UUID.randomUUID().toString()).tokenValiditySeconds(1209600);

}

}

}

## 2.5 监控客户端

Pom.xml 引入

<artifactId>admin-client</artifactId>

<dependencies>

<dependency>

<groupId>de.codecentric</groupId>

<artifactId>spring-boot-admin-starter-client</artifactId>

<version>2.2.1</version>

</dependency>

</dependencies>

application.properties文件引入

spring.boot.admin.client.instance.name = 007

spring.boot.admin.client.url= http://localhost:9000

management.endpoints.web.exposure.include=\*

spring.boot.admin.client.username = admin

spring.boot.admin.client.password = admin

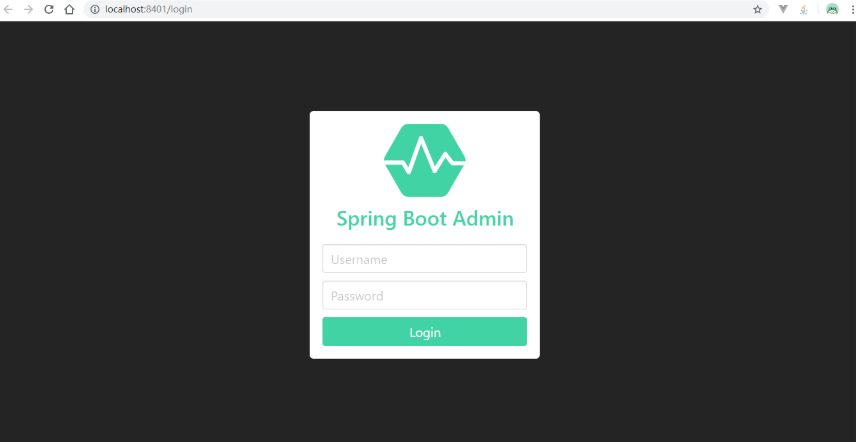
spring.boot.admin.client.period = 10000

spring.boot.admin.client.connect-timeout = 5000

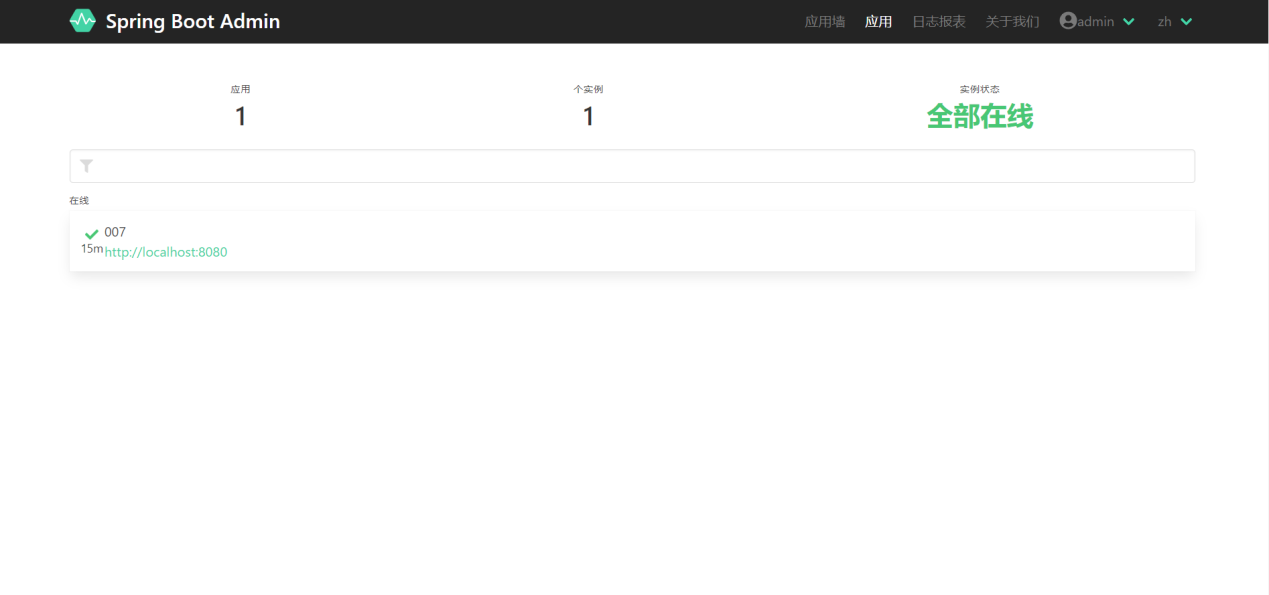
spring.boot.admin.client.read-timeout = 5000

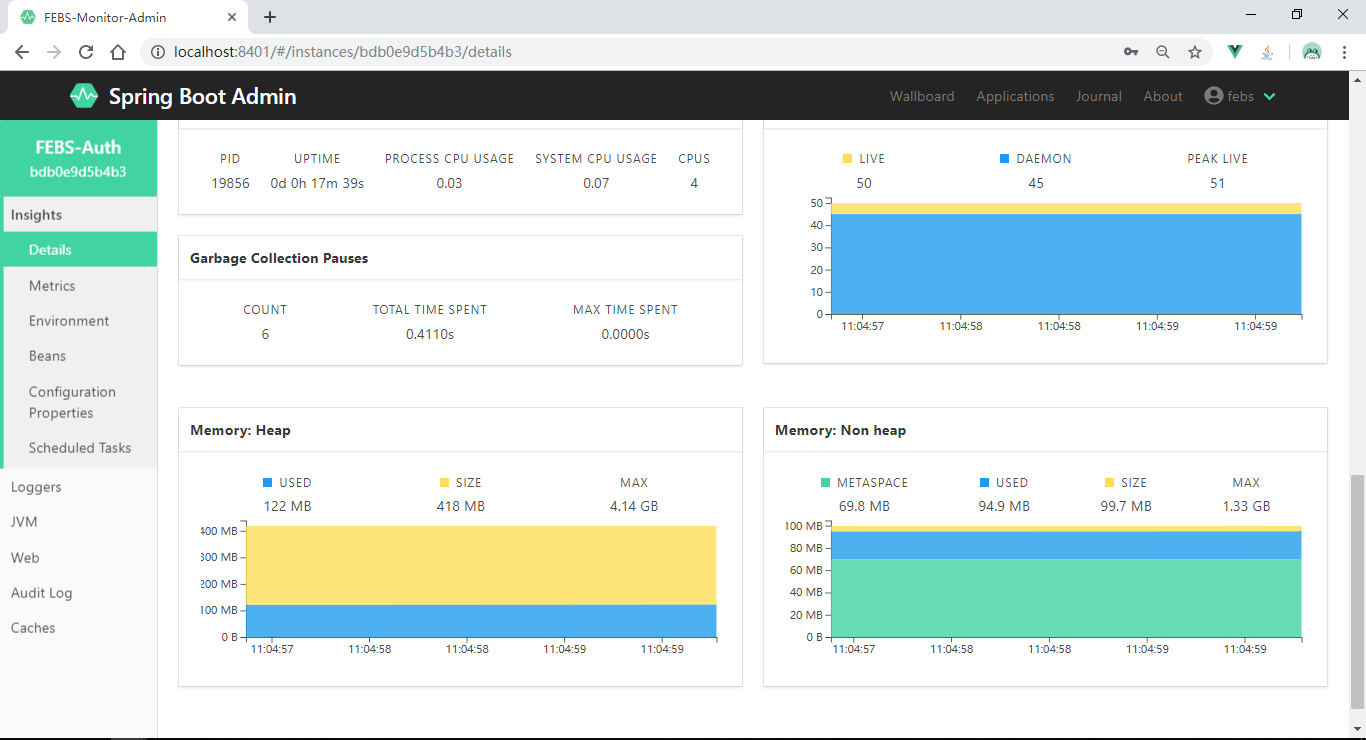
spring.boot.admin.client.instance.service-url = http://localhost:8080

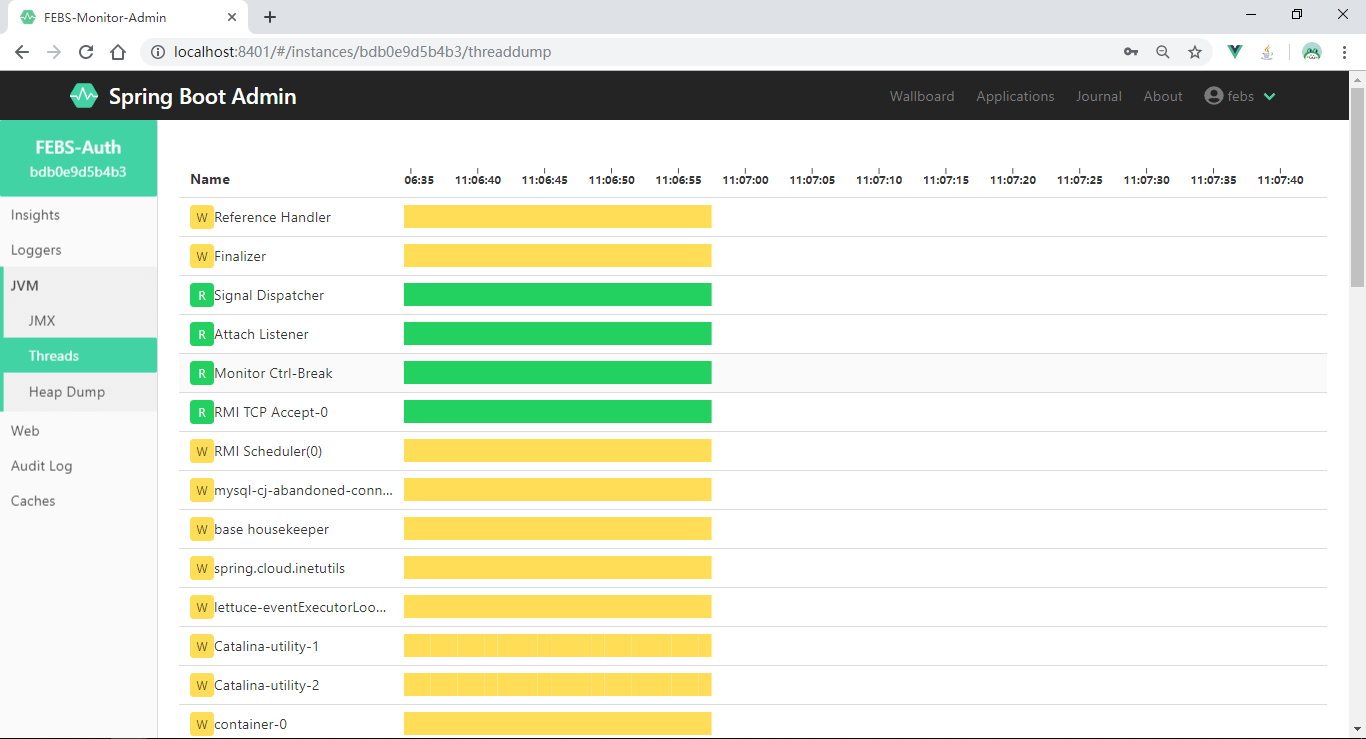
## 2.6监控页面

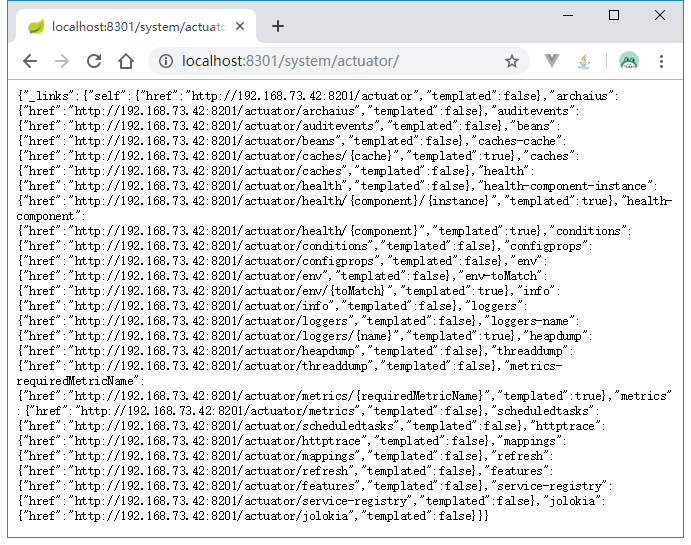












# 小结：

2.x 版本的功能还是比较齐全，界面也比较友好。