# Admin功能介绍

1. 自动收集每个微服务产生的HTTP请求日志(不管是用@Restcontroller还是zuul)
2. 自动收集每个微服务访问数据库的sql日志(hibernate目前暂不支持获取执行时间功能)
3. 可获取发现服务中的每个微服务的服务器信息(JVM,内存,GC,JDK,堆栈)
4. 每个微服务的HTTP接口日志统计分析功能（每个uri的成功 失败 总次数 请求平均时间，单次请求最长时间）
5. 每个微服务的SQL日志统计分析功能(可以筛选出执行时间大于1.5秒的sql语句)
6. 分布式链路追踪(追踪一个请求的整个生命周期)
7. 预警邮件发送(未完成)

# Admin架构介绍

## 涉及技术

### 后端：

Websocket

Elastisearch

hibernate

Spring-boot

Spring-boot-actutor

Spring-boot-task

Spring-boot-aop

Spring-boot-orm

Spring-cloud-sleuth

Spring-cloud-eureka

Spring-cloud-zuul

### 前端:

Vue

Bootstarp

Jquery

Jquery.jqgrid

Layer

Echart

# Admin服务器

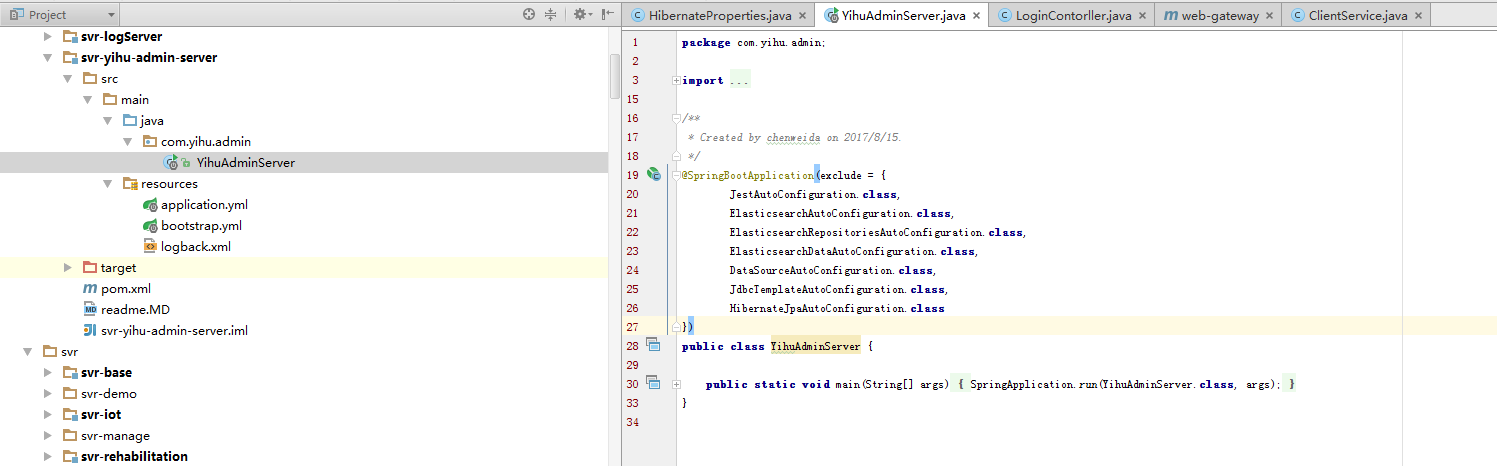
## 新建springboot的工程

版本1.5.9

## 添加maven依赖

<**properties**>  
 <**version.yihu.admin**>1.2.0</**version.yihu.admin**>  
</**properties**>  
<**dependencies**>  
 <**dependency**>  
 <**groupId**>com.yihu</**groupId**>  
 <**artifactId**>admin-server-starter</**artifactId**>  
 <**version**>${version.yihu.admin}</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>com.yihu</**groupId**>  
 <**artifactId**>admin-server-ui-starter</**artifactId**>  
 <**version**>${version.yihu.admin}</**version**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-tomcat</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-starter-eureka</**artifactId**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.springframework.boot</**groupId**>  
 <**artifactId**>spring-boot-starter-actuator</**artifactId**>  
 </**dependency**>

## 新增启动类



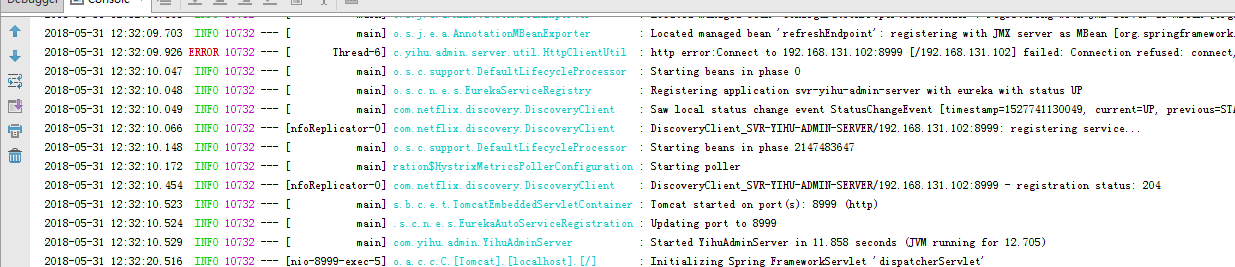
@SpringBootApplication(exclude = {  
 JestAutoConfiguration.**class**,  
 ElasticsearchAutoConfiguration.**class**,  
 ElasticsearchRepositoriesAutoConfiguration.**class**,  
 ElasticsearchDataAutoConfiguration.**class**,  
 DataSourceAutoConfiguration.**class**,  
 JdbcTemplateAutoConfiguration.**class**,  
 HibernateJpaAutoConfiguration.**class**})  
**public class** YihuAdminServer {  
  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(YihuAdminServer.**class**, args);  
 }  
}

## 4添加yml配置

**server:  
 port:** 8999 *##端口修改 前端工程也需要改造***spring:  
 application:  
 name:** svr-yihu-admin-server  
 **data:  
 elasticsearch:  
 cluster-name:** jkzl *#es集群的名字* **cluster-nodes:** 172.19.103.68:9300 *#多个逗号分割* **cluster-nodes-jest:** http://172.19.103.68:9200 *#多个逗号分割* **jest:  
 uris:** ${**spring.data.elasticsearch.cluster-nodes-jest**}  
 **discovery-enabled:** false *#开启嗅探* **conn-timeout:** 60 *# 连接时间单位是秒* **read-timeout:** 60 *# 读取时间单位是秒* **multith-readed:** true *#开启多线程* **max-total-connection:** 50 *##最大连接* **max-connection-idle-time:** 50 *#最大等待时间* **elasticsearch:  
 jest:  
 read-timeout:** ${**spring.data.elasticsearch.jest.read-timeout**}  
 **multith-readed:** ${**spring.data.elasticsearch.jest.multith-readed**}  
  
  
**logging:  
 file:** ${**logging.path**}svr-yihu-admin-server.log  
 **path:** G:\idea\_workspace\jkzl\_git\jkzl-start\logs\ ##这边路径需要改成实际的路径  
 **level:  
 root:** INFO  
  
  
  
  
  
**management:  
 health:  
 db:  
 enabled:** false

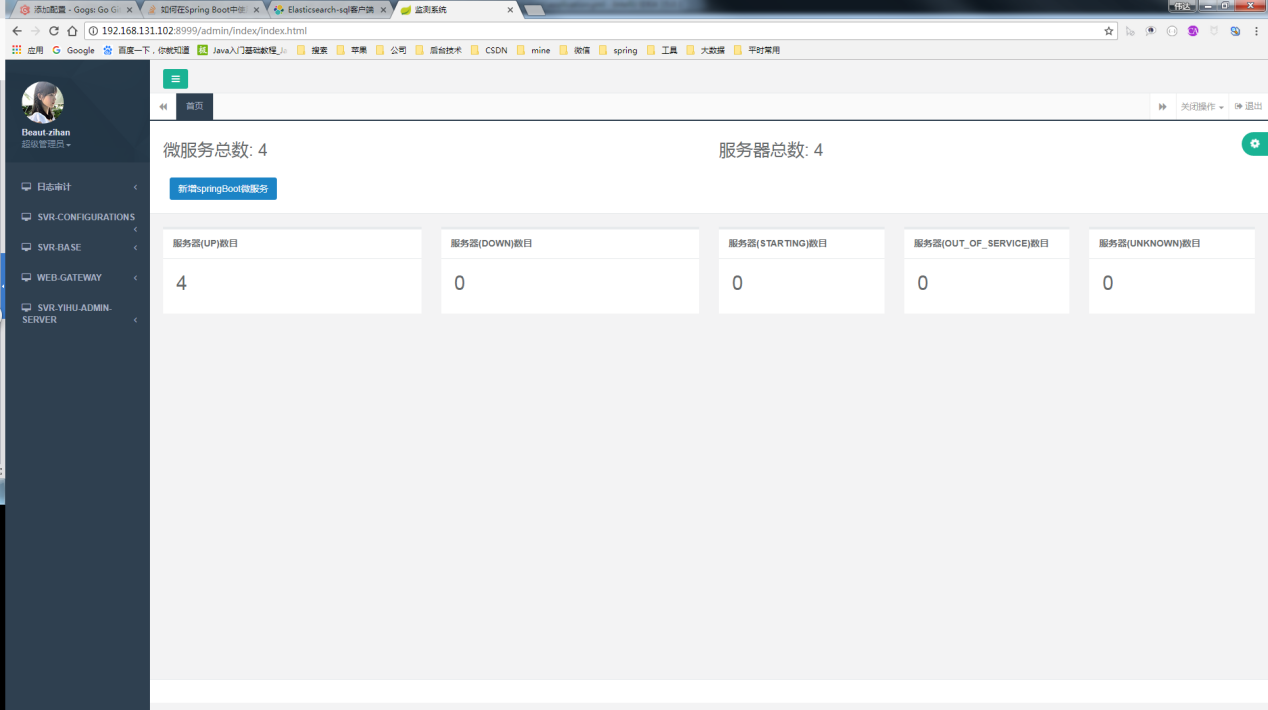
**yihu:  
 admin:  
 server:  
 defaultIndexName:** admin\_log *##默认admin\_log* **defaultTypeName:** admin\_log *##默认admin\_log* **partition:** month *##默认是week day week month year*

## 5启动



UI路径

http://192.168.131.102:8999/admin/index/index.html



# Admin客户端

## 1新建springboot工程

版本1.5.9

## 添加maven依赖

<**dependency**>  
 <**groupId**>com.yihu</**groupId**>  
 <**artifactId**>admin-client-starter</**artifactId**>  
 <**version**>${version.yihu.admin}</**version**>  
</**dependency**>

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

## 配置yml

**eureka:  
 client:  
 serviceUrl:  
 defaultZone:** http://jw:jkzl@127.0.0.1:8761/eureka/  
  
  
 **instance:** *#eurika使用IP不使用host* **prefer-ip-address:** true  
 *#定制化在eurika中显示的名称  
 ##instance-id: ${spring.application.name}:${spring.application.instance\_id:${server.port}}* **instance-id:** ${**spring.cloud.client.ipAddress**}:${server.port}  
 **leaseRenewalIntervalInSeconds:** 30 *##心跳时间 默认30秒 一般不修改***management:  
 security:  
 enabled:** false  
  
  
**endpoints:  
 logfile:  
 enabled:** true  
  
  
**logging:  
 file:** ${**logging.path**}demo-socket-client.log  
 **path:** G:\idea\_workspace\jkzl\_git\jkzl-start\logs\ ##这边路径需要改成实际的路径  
 **level:  
 root:** INFO

**yihu:  
 admin:  
 client:  
 advice:  
 http:** true *#默认true* **sql:  
 hibernate:** true *##默认false* **springJDBC:** true *##默认false* **zuul:  
 pre:** true *#默认false* **post:** true *#默认false* **server:  
 ip:** 192.168.131.102  
 **port:** 8082

## 4 如需要使用hibernate 需要在hibernate管理添加拦截器

**import** org.hibernate.EmptyInterceptor;  
**import** org.hibernate.Interceptor;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.beans.factory.annotation.Value;  
**import** org.springframework.cloud.context.config.annotation.RefreshScope;  
**import** org.springframework.stereotype.Component;  
  
**import** java.util.List;  
**import** java.util.Properties;  
  
*/\*\*  
 \* Created by chenweida on 2017/4/6.  
 \*/*@Component  
*//@RefreshScope***public class** HibernateProperties {  
 @Value(**"${hibernate.dialect}"**)  
 **private** String **dialect**;  
 @Value(**"${hibernate.show\_sql}"**)  
 **private** String **show\_sql**;  
*// @Value("${hibernate.ejb.naming\_strategy}")  
// private String naming\_strategy;* @Value(**"${hibernate.physical\_naming\_strategy}"**)  
 **private** String **physical\_naming\_strategy**;  
 @Value(**"${hibernate.implicit\_naming\_strategy}"**)  
 **private** String **implicit\_naming\_strategy**;  
 @Autowired  
 **private** List<Interceptor> **interceptors**;  
  
 **public** Properties hibProperties() {  
 Properties properties = **new** Properties();  
 properties.put(**"hibernate.dialect"**, **dialect**);  
 properties.put(**"hibernate.show\_sql"**, **show\_sql**);  
 properties.put(**"hibernate.physical\_naming\_strategy"**, **physical\_naming\_strategy**);  
 properties.put(**"hibernate.implicit\_naming\_strategy"**, **implicit\_naming\_strategy**);  
 properties.put(**"hibernate.ejb.interceptor"**, **interceptors**.get(0)); ##添加这行  
 **return** properties;  
 }  
}

## 5启动即可