**使用说明书**

洺信科技监控日志组件

目录.....................................................................................................................1

说明.....................................................................................................................2

1 使用方式.........................................................................................................3

* 1. 引入依赖...............................................................................................3
  2. 配置数据库...........................................................................................4
  3. 配置查询账号.......................................................................................5

2 组件原理.........................................................................................................6

2.1 采集和储存...........................................................................................7

2.1.1 全局过滤器.................................................................................8

2.1.2 异常拦截.....................................................................................9

2.1.3 异步存储

2.2 数据库...................................................................................................10

2.2.2 动态装载...................................................................................10

2.2.3 JDBC实现..................................................................................11

2.2.4 扩展性预留...............................................................................11

2.3 查询.......................................................................................................12

2.3.1 自定义路由.................................................................................12

2.3.2 访问控制.....................................................................................12

2.3.3 条件分页查询.............................................................................13

**说明**

为满足公司各项目的请求监控，并记录请求和响应相关的数据。开发了此请求监控组件，组件以拔插式、轻量级、低耦合、高扩展性的指标开发完成。业务项目只需要引入依赖并添加简单配置即可使用。组件通过maven私有仓库实现版本管理，不同的需求阶段更新不同的版本号，该组件已经内聚Mysql、ElasticSearch数据库和前端UI页面。项目添加完成后即可通过以下网址访问日志数据。http://项目网关/mxkj/index.html

本说明书基于目前版本1.0.4-RELEASE，后续版本如发生升级和变更，本文档可做参考处理。

**一、使用方式**

1. **引入依赖**

引入依赖前，需要在pom中先配置洺信科技的Maven私有仓库。

<repositories>  
 <repository>  
 <id>my-public</id>  
 <url>http://192.168.10.10:18081/repository/maven-public/</url>  
 <releases>  
 <enabled>true</enabled>  
 </releases>  
 <snapshots>  
 <enabled>true</enabled>  
 <updatePolicy>always</updatePolicy>  
 </snapshots>  
 </repository>

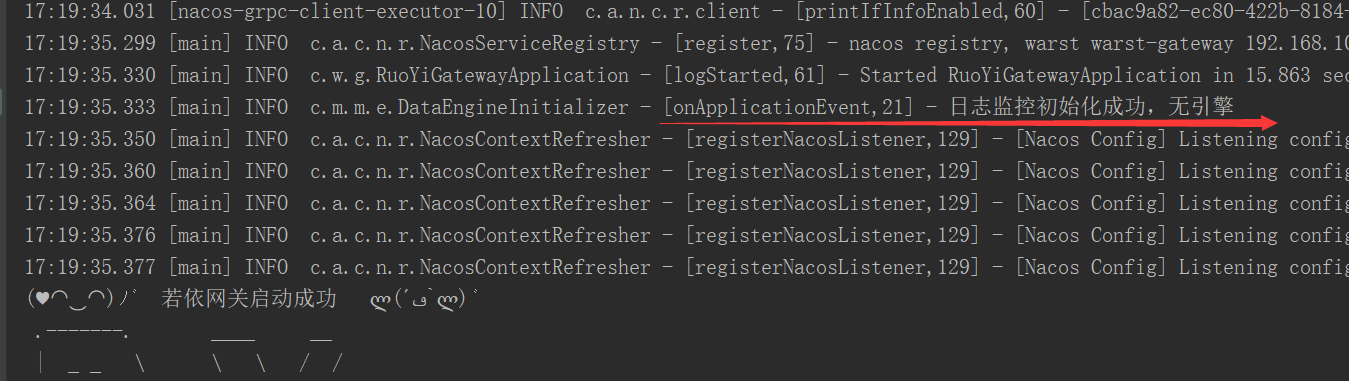
</repositories>

仓库配置完成后，在项目网关gateawy项目的pom中引入以下依赖

<dependency>  
 <groupId>com.mxkj</groupId>  
 <artifactId>monitor-log</artifactId>  
 <version>${最新版本}</version>  
</dependency>

1. **配置数据库**

依赖引入后，默认不配置任何数据库。此时启动项目网关，将会打印如下日志：

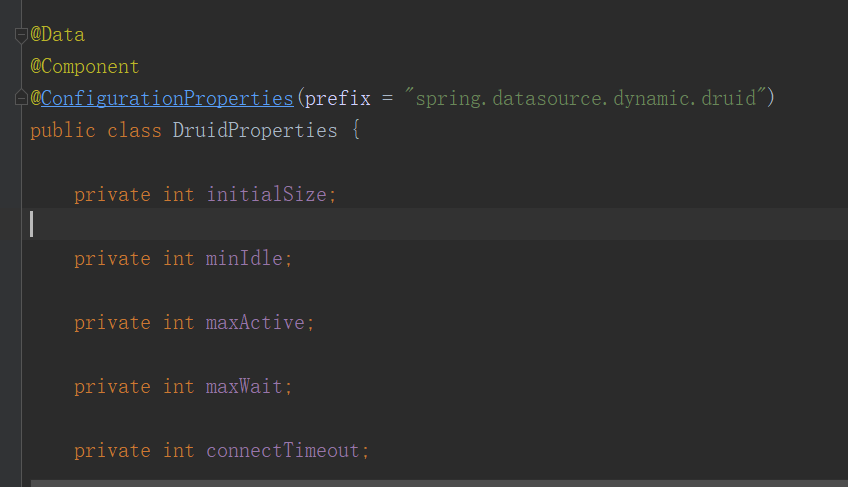


如果要配置数据库，目前可以选择Mysql和ElasticSearch两种，当Mysql和Elasticsearch都被配置时以Elasticsearch为主。

1. **配置Mysql**

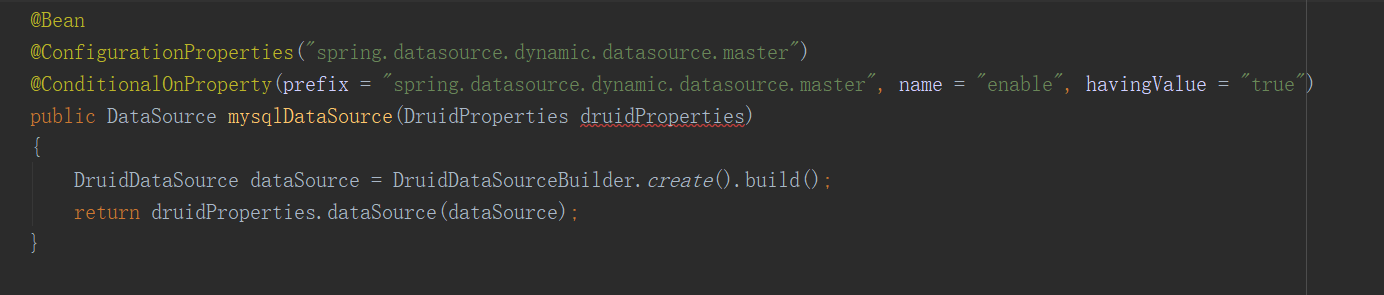
Mysql的配置主要有数据库驱动Driver和数据库连接池Druid.

连接池Druid的配置前缀为 spring.datasource.dynamic.druid

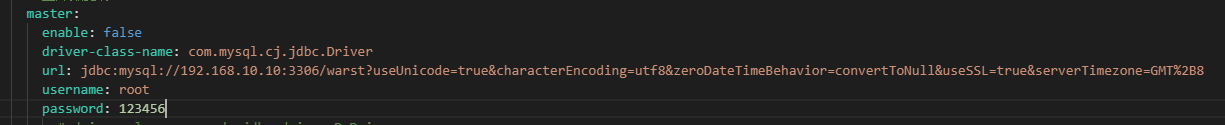


（以上相关的配置可缺省）

数据库的配置前缀为spring.datasource.dynamic.datasource.master, 其中enable用于控制是否启动。

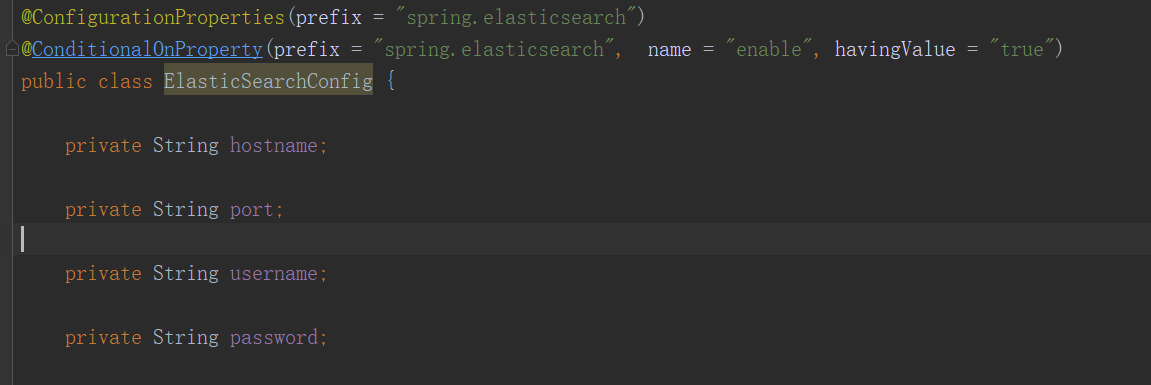


如下图：



1. **配置Elasticsearch**

数据库的配置前缀为spring.elasticsearch, 其中enable用于控制是否启动。



spring:

elasticsearch:

enable: false

hostname: localhost

port: 9200

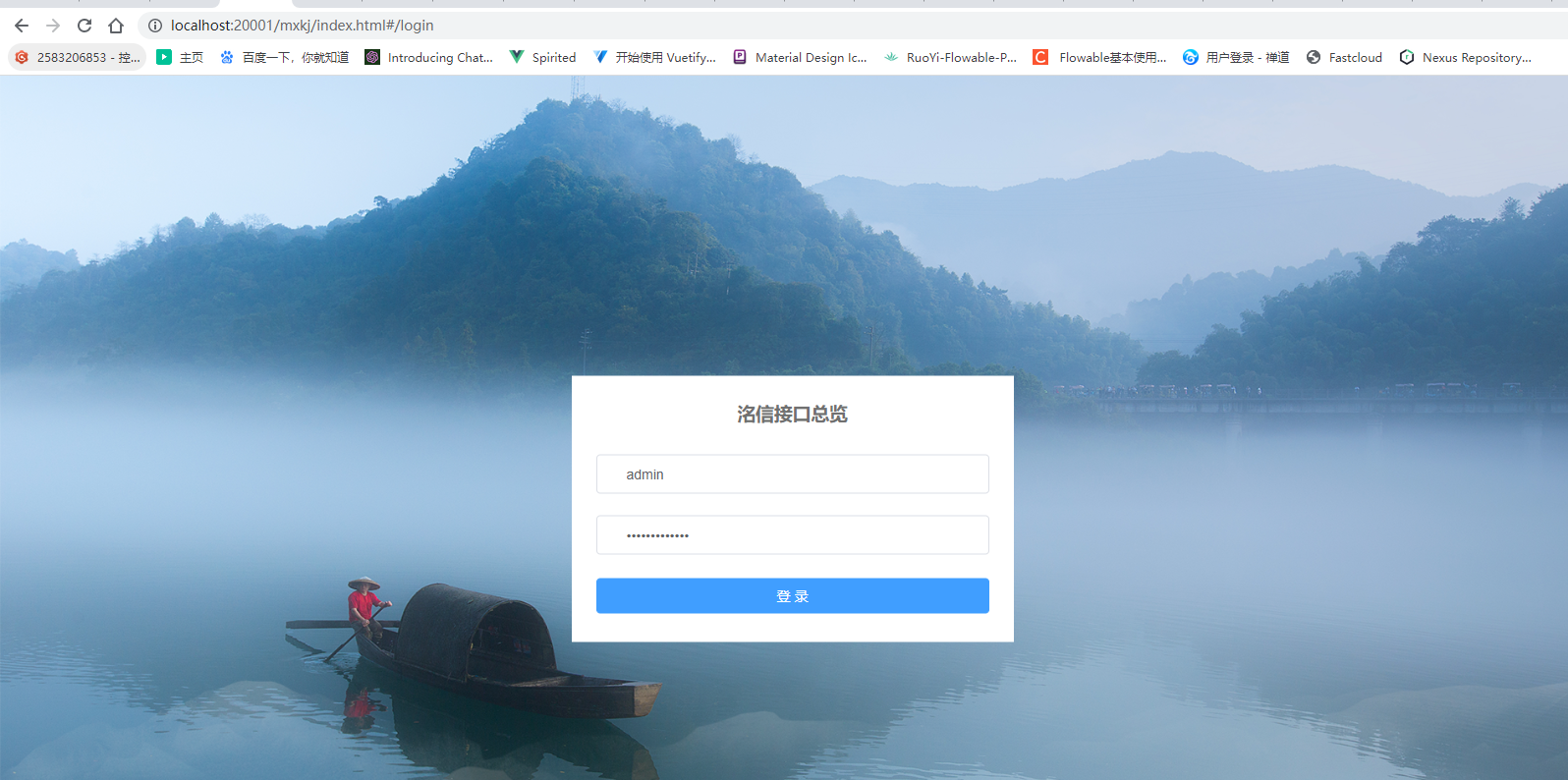
username: 1234

password: 1234

3 **. 配置查询账号**

日志存储在配置的数据库中，如果要查询日志，通过访问以下网址登录访问。

<http://项目网关/mxkj/index.html>



访问需要在启动网关时配置访问的账号密码：

Session是会话有效时间，如果不配置，则默认30分钟过期。

mxkj:

auth:

username: mxkj

password: mxkj123..

session: 30

二、**组件原理**

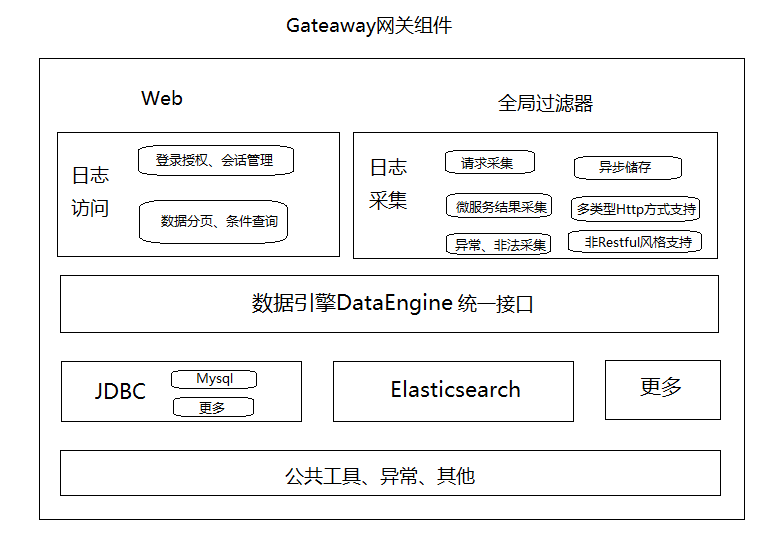
日志组件主要分为三大部分，收集、储存和查询。

收集过程中除了收集各个微服务项目的请求响应结果，还包括对系统的未知请求、访问受限和非法请求，如404、401和500以上请求等。

数据捕获后，通过配置的数据库实现存储和更新等操作。为防止阻塞和造成延迟，数据捕获后采用异步方式储存数据。

数据储存后，可以通过组件内聚的UI界面实现授权登录和访问。数据查询配置会话有效期保证访问的安全性。

下面将会从源码角度解析功能的设计原理。



组件架构图

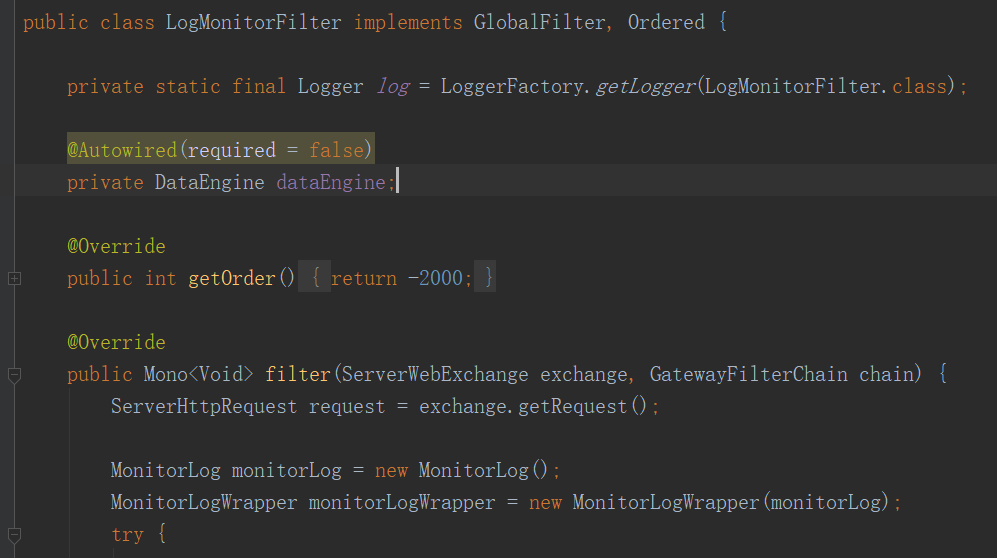
日志采集

1. 全局过滤器

日志采集是整个组件的第一步，采集目标为各种请求类型、地址、参数以及对应的正常结果或异常响应。

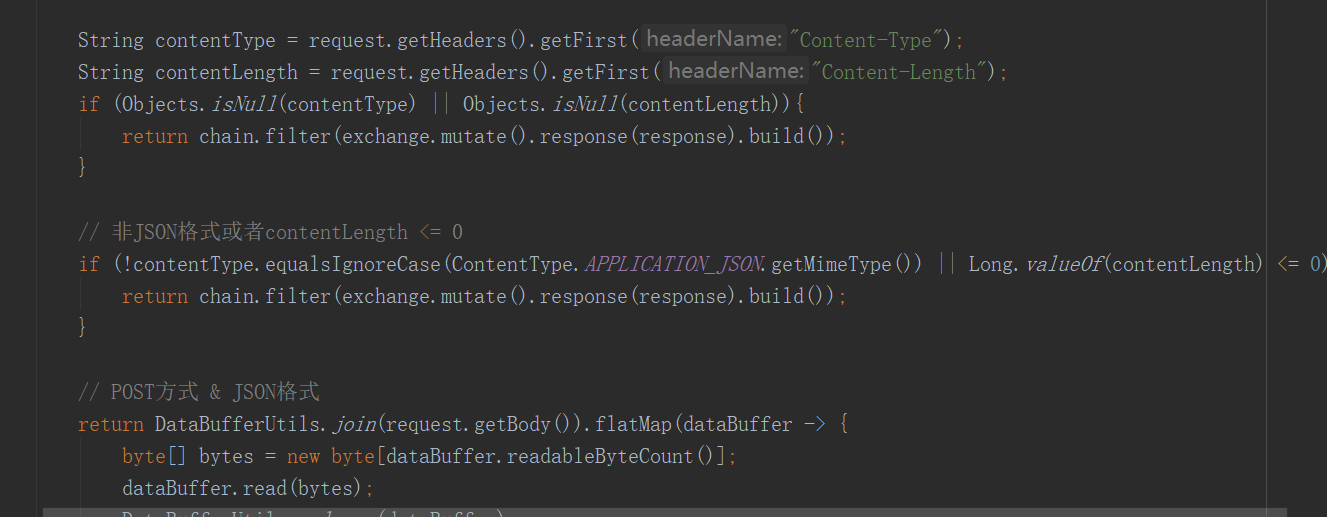
为了拦截全局请求，采用向Gateway网关注入高优先级的全局过滤器方式拦截进入网关的所有请求。 通过重写响应处理和注入高优先级的异常处理器ExceptionHandler捕获正常和异常响应。

全局过滤器目前优先级别默认为2000。



在该过滤器中，对于请求捕获，分为带请求体和不带请求体两种方式的请求。

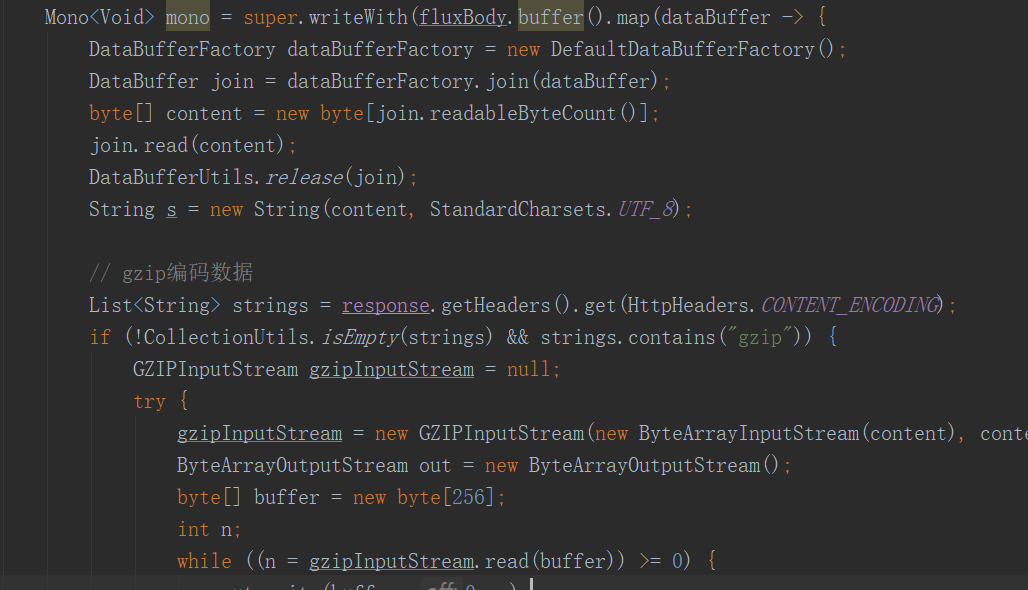
在目前的版本中，仅对POST方式且数据格式为JSON的请求实现请求体的参数采集。对于其他请求暂不对请求体的参数进行采集。



在对请求体参数采集时，由于网络流式对象的不可重复读取的特性，采集时必须先读取数据并缓存到内存中，然后将数据记录同时向后续传递缓存的参数。这里通过缓存数据重新构建ServerHttpRequest对象，写入body参数，传入到后续的过滤链中。

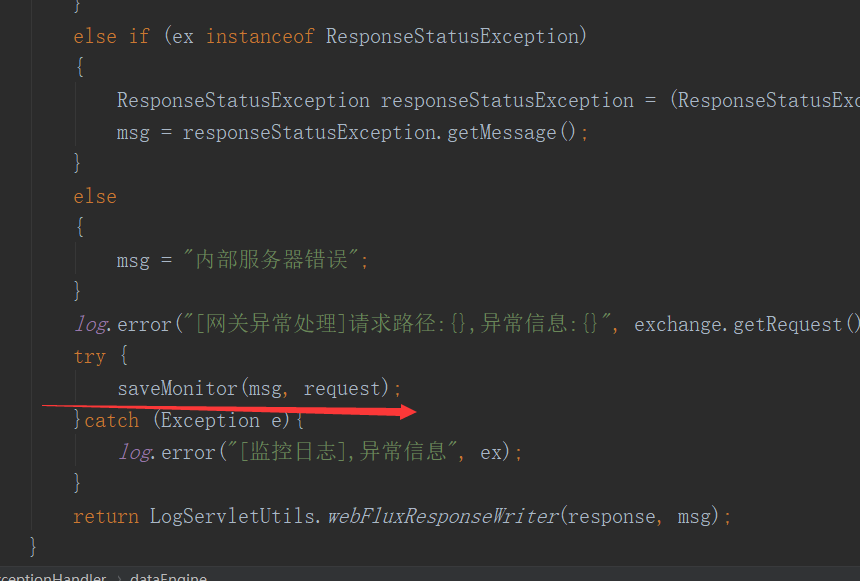


在对响应体进行采集时，同样需要考虑该问题。同时支持对gzip编码数据块的响应支持。



1. 异常拦截

如果在请求过程中出现异常，如对非注册路由实现请求、未鉴权访问请求、系统异常等通过实现高优先级的ExceptionHandler实现捕获，

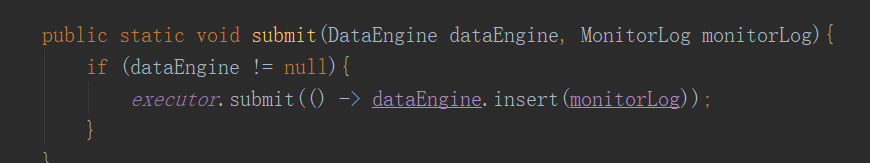


需要注意的是，如果优先级小于系统网关原有的其他异常处理器，该处理器的异常拦截将会失效。目前默认-100。



1. 异步储存

在类FilterCatchMap中，配置ThreadPoolExecuter线程池，实现日志的异步存储，防止主业务的加载阻塞或者异常。

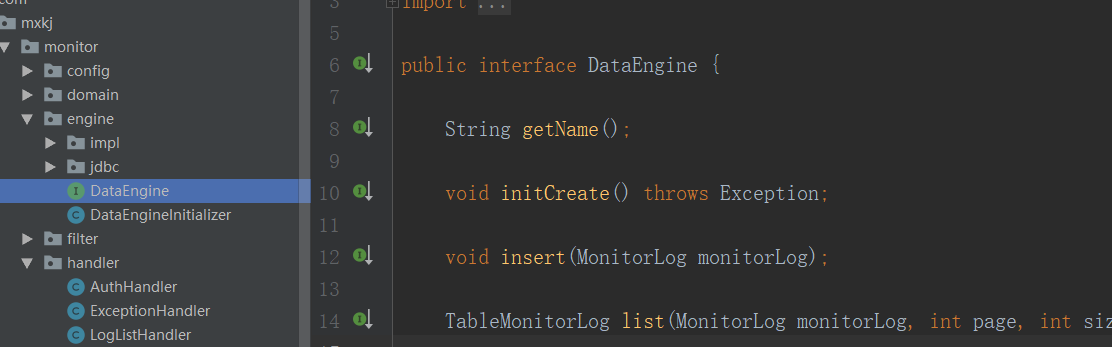


日志储存

1. 动态装载

组件通过封装数据库支撑特性，抽象统一访问接口类DataEngine.

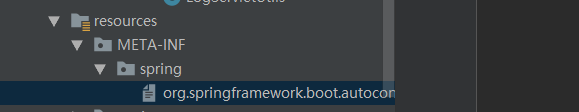
包含获取数据库名称、数据库初始化构建、向数据库插入数据、条件分页查询等行为方式。



其实现类目前支持Mysql和Elasticsearch。

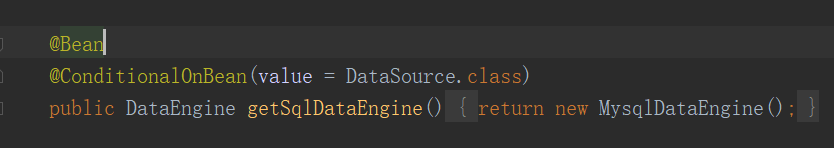
当组件加载时，将通过配置实现数据库引擎的动态装载。动态装载的特性依赖与springboot的IOC容器管理。

首先通过springboot自动装载特性在resources/META-INF/spring/org.springframework

.boot.autoconfigure.AutoConfiguration.imports中注入需要被加载的对象。

扫描时，会扫描到config包下的MysqlConfig和ElasticSearchConfig。

该包中都存在getSqlDataEngine()方法，实现向IOC中注入DataEngine实现类。



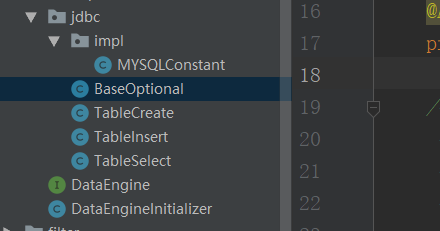
最后，DataEngineInitializer实现了ApplicationListener<ApplicationStartedEvent>接口，在

IOC容器加载完后后，通过实现该接口的onApplicationEvent方法，完成对应数据库的初始化。



1. JDBC实现

在engine.mysql包中，实现了基于JDBC通信协议的数据库操作的轻量级ORM框架，可以通过JDBC操作Mysql数据库。对应增删改查分别实现了Create、Insert、Select操作。以及下划线与小驼峰的命名映射方式等。方便基于数据对象与数据库字段的属性映射。



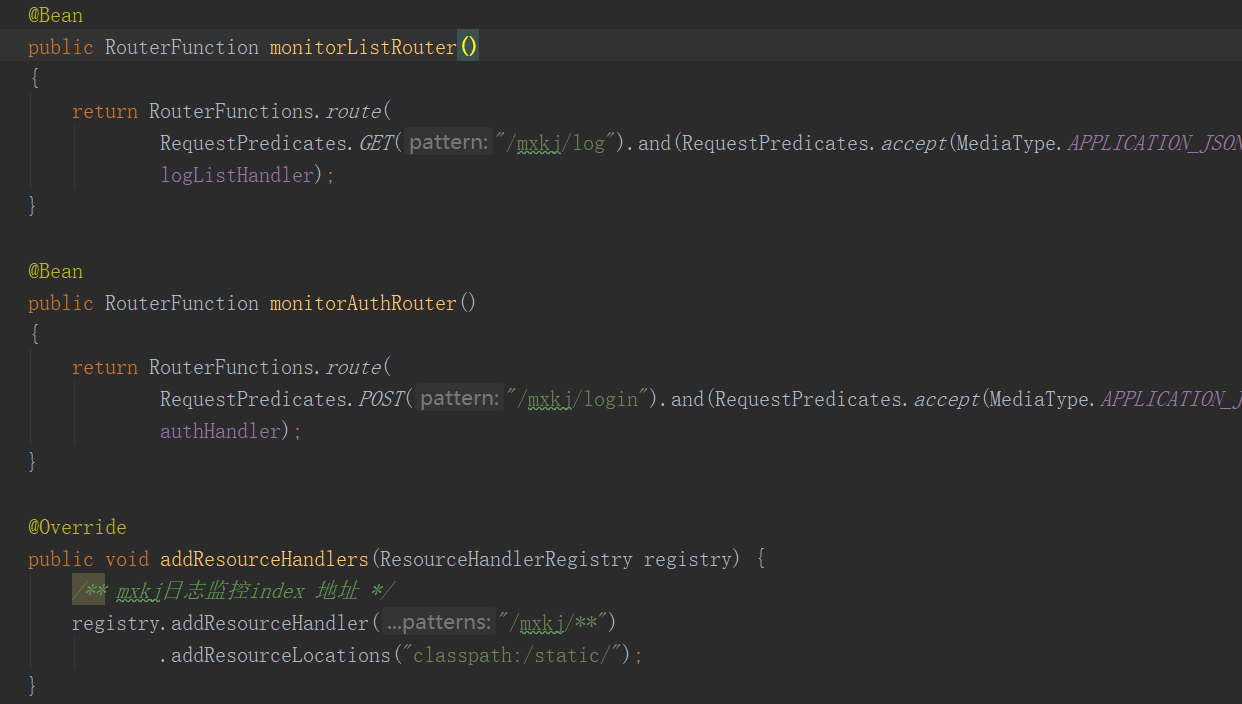
1. 扩展性预留

在该架构模式下，可以通过实现DataEngine接口，扩展更多的数据库支持。如其他DBMS规范的数据库，ORABLE、达梦、SqlServer等，组件中实现了基于JDBC的轻量级ORM框架，可以直接实现扩展。也可以支持其他非关系型数据库Redis、MangoDB等等。

日志查询

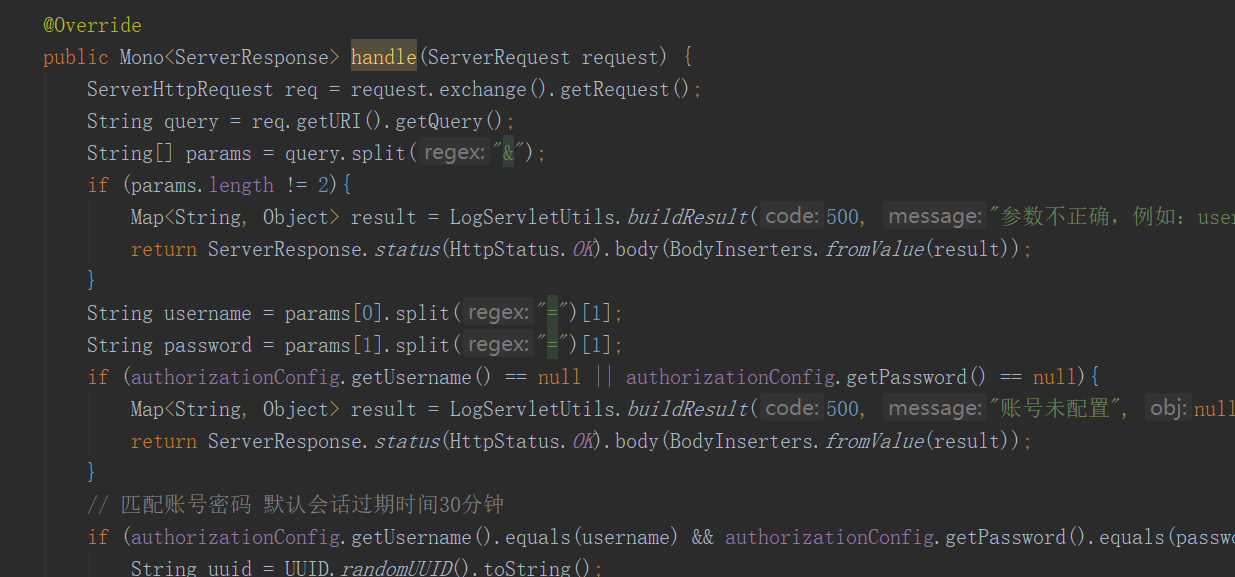
1. 自定义路由

Gateaway的底层依赖于Webflux，Webflux底层依赖于Netty， WebFlux是一个类似wevMvc的路由访问控制器，除了在yml文件中配置路由。组件通过配之类MonitorRouteConfiguration实现路由对象注入。对于日志查询的访问接口，在这个配置类向Gateway网关中注入几个路由对象实现外部web访问，同时配置静态资源映射，实现静态页面的资源访问。



1. 访问控制

在handler包中，实现了AuthHandler，当请求访问到mxkj/login接口，会通过配置的AuthHandler类来处理请求。该类中实现了基于账号密码的访问控制，实现了基于内存的会话有效期控制等。



1. 条件分页查询

条件分页查询同理，通过在handler包中实现LogListHandler类，该类会通过DataEngin访问底层数据库，并实现封装。

