

日志平台开发和说明文档

笔记本: mysql

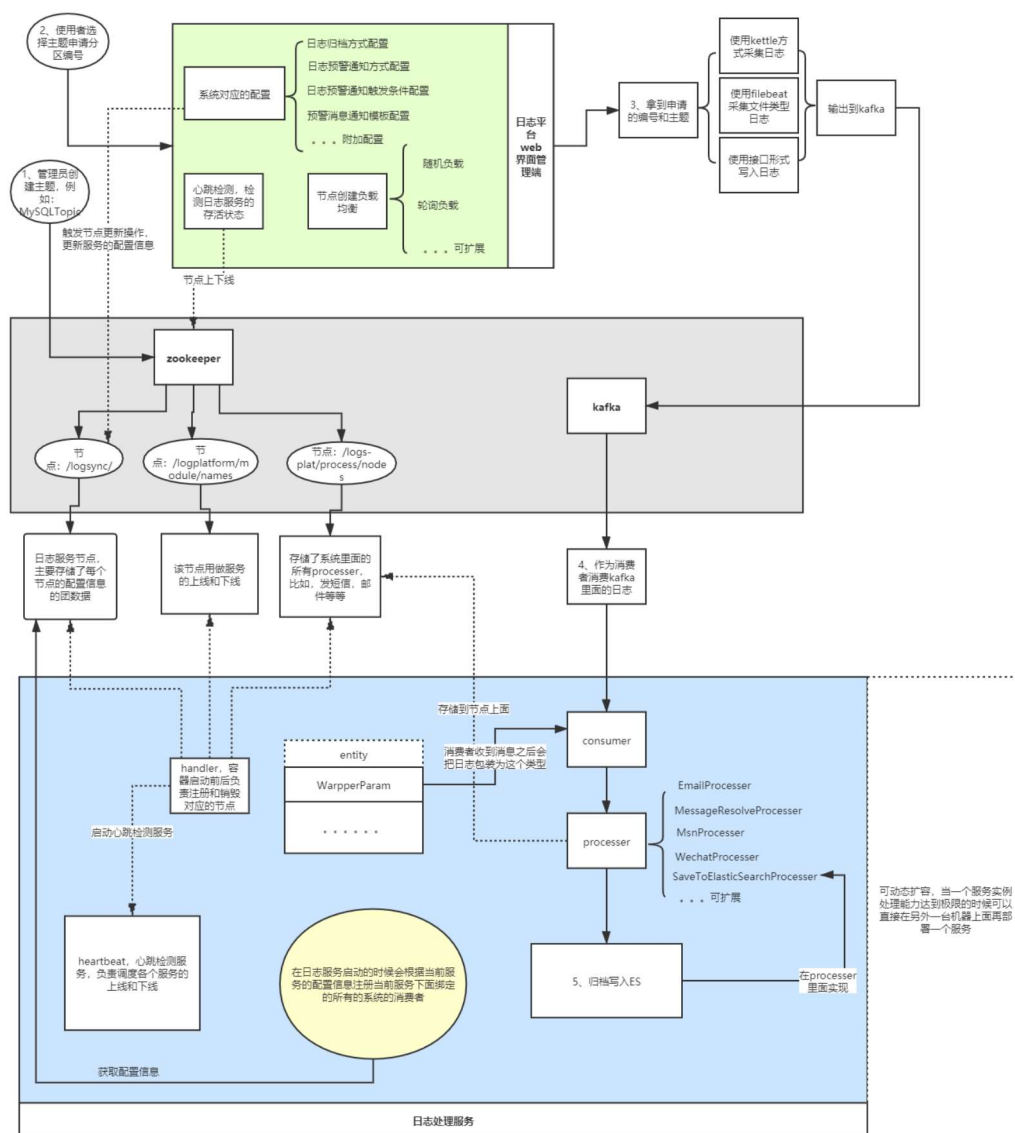
创建时间: 2020/6/8 9:40

更新时间: 2020/6/8 10:40

作者: jin.zhou@definesys.com

结构划分

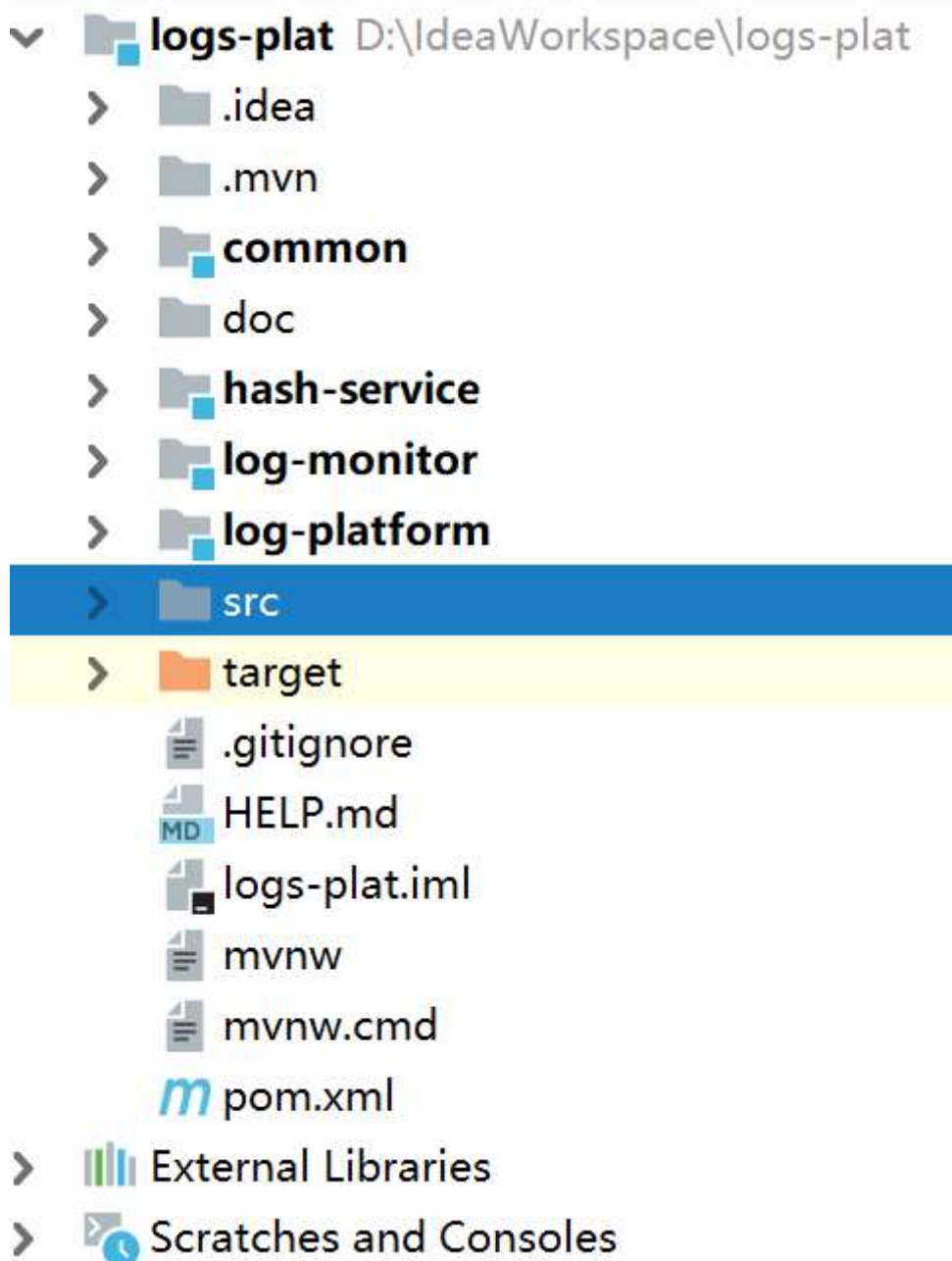
整个日志平台分为服务端和客户端，
服务端提供日志拦截，预警的功能，
客户端提供日志预警规则配置，日志服务的自动上线下线功能以及日志服务使用时申请分区的功能
整体的功能逻辑大致如下图所示



项目结构

项目采用maven父子工程结构，一共有四个模块

- **common**
公用模块，其中主要是一些公用的工具方法以及公用的实体类
- **hash-service**
该模块可作为一个接口服务部署到任意地方，主要是在做分区的hash分发时候使用的，为了确保消费者能够消费到正确的分区的数据
- **log-monitor**
日志预警服务，该模块是一个后台服务，会对它所监听的所有日志进行一个拦截处理，根据不同处理规则进行处理
- **log-platform**
日志平台，该平台是为日志预警服务提供的，在这个平台上面使用者可以申请自己系统对应的日志分区，配置自己系统的元数据，包括通知方式，归档模式，消息模板等等。同时还提供了服务的心跳检测和上线下线的功能



log-monitor的开发

在这个模块里面核心功能主要是以下三个模块：

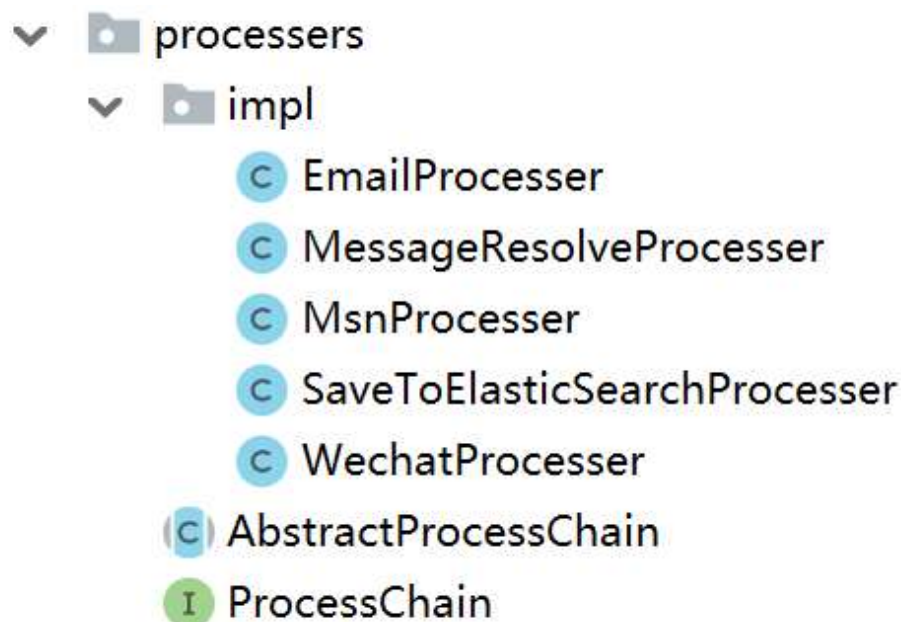
1、消费者模块

在消费者里面我们主要做的事情是 解析节点下面的所有系统，为每个系统分配一个消费者线程并且消费指定的分区，消费者收到消息之后对消息进行包装，把消息发送给默认的processor去处理。后续开发过程中对这一块改动应该是不大



2、processor模块

在processor模块里面我们使用了一个责任链的思想对消息做一个链式处理。在下面的实现类里面可以通过注解的方式对这些实现类进行一个排序，目前已经实现了EmailProcessor, MessageResolveProcessor和SaveToElasticSearchProcessor, 生效MSNProcessor和WeChatProcessor需要去实现，如果后面需要添加新的功能可以直接添加实现类就好，注意注解里面的名称不要重复就好了



3、handler模块

这个模块主要的功能是在服务启动的时候和服务停止的时候做一些必要的处理，目前在启动时主要做的事情是 启动消费者，注册processor，注册系统服务名称（服务上

▼  hander

© StopAddDataListener

log-platform

▼ src

▼ java

- balanceloaders


- RandomBalanceLoader

I BalanceLoader

▼ controller

c TimeJobsController

> hander

-  repositories

-  utils

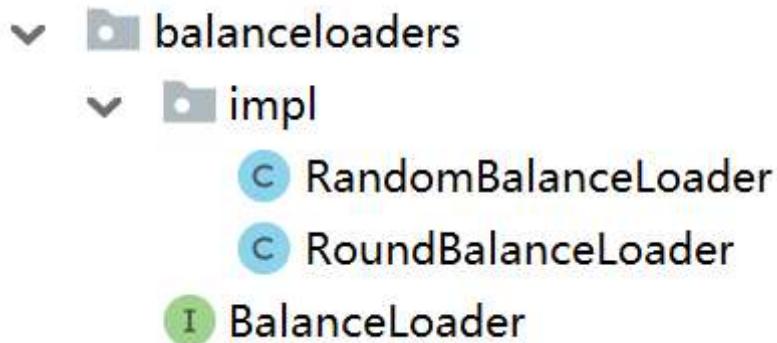
resources

target

其中主要的部分有如下几个

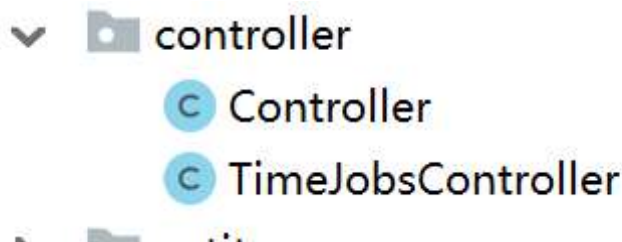
balanceloaders 负载均衡

该模块是一个负载均衡策略模块，在申请分区的时候如果服务是用集群模式部署的，那么我们需要使用一个策略来选择申请哪个服务上的分区，下图中负载均衡策略实现了随机和轮询两种策略，当然也可以扩展，比如权重随机，一致性hash等等，可根据实际情况自行实现并选择对应的策略，注意，可在实现类上面加上@Primary注解来激活当前的负载均衡策略



controller

目前系统的接口不多，所以全部集成在Controller里面



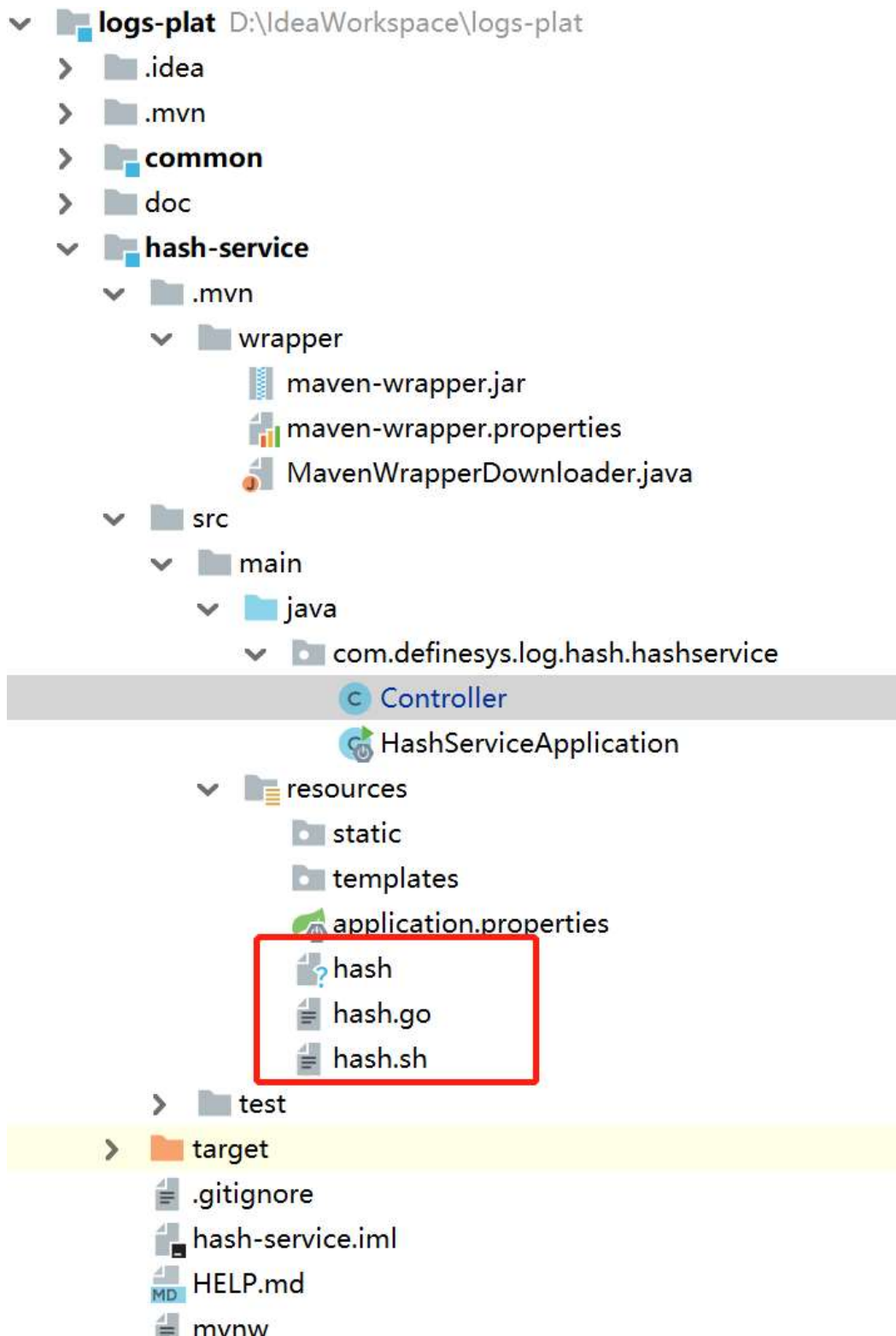
heartbeat

心跳检测模块，主要是定时来检查服务是否存活，使用netty4 开发

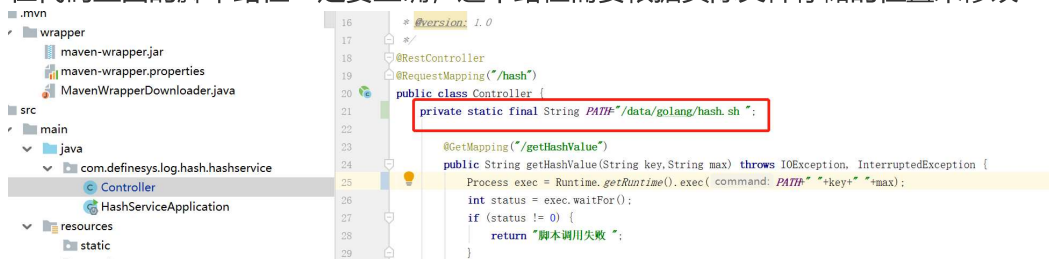


如何部署hash-service

hash-service部署方式很简单，直接使用mnv package 打包，在服务器上面运行 `java -jar` 就可以运行。不过需要注意的是服务器需要安装好go语言的环境，然后图中三个文件一定要保存好。



在代码里面的脚本路径一定要正确，这个路径需要根据实际文件存储的位置来修改



如何部署日志服务 (log-monitor)

部署日志服务也不难，在代码确认无误之后使用maven吧项目打包，直接在服务器上面使用java -jar就可以运行了，但需要注意的是需要在启动命令里面加上节点名称的参数

例如：`java -jar log-monitor-service.jar -DnodeName=node1`

如何部署log-platform

log-platform是一个web应用，可以当做普通的rest服务来部署，可以部署在中台上或者直接使用java -jar运行