# 新版终端系统详细设计文档

詹小雷 2020/06/28

# 项目概述

新版终端系统致力于为公司官网首页提供7\*24小时实时财经金融快讯及最新的行业动态信息。涵盖1500多个政府，金融，经济，财经的国内外网站，通过获取监控网站最新发布的新闻消息，经过编译部门的快速精准筛选最终推送给官网首页及附加产品(app等)。

# 技术架构设计

新版终端系统的后端主要基于2种主流语言实现，网页爬虫任务专门由python做处理，爬虫任务的调度，分发，监控及消息组装推送都是用java实现的，前端基于web网页端实现。本文档主要详细描述java端部分的核心设计。

## 2.1 技术栈

爬虫终端系统主体技术栈如下，

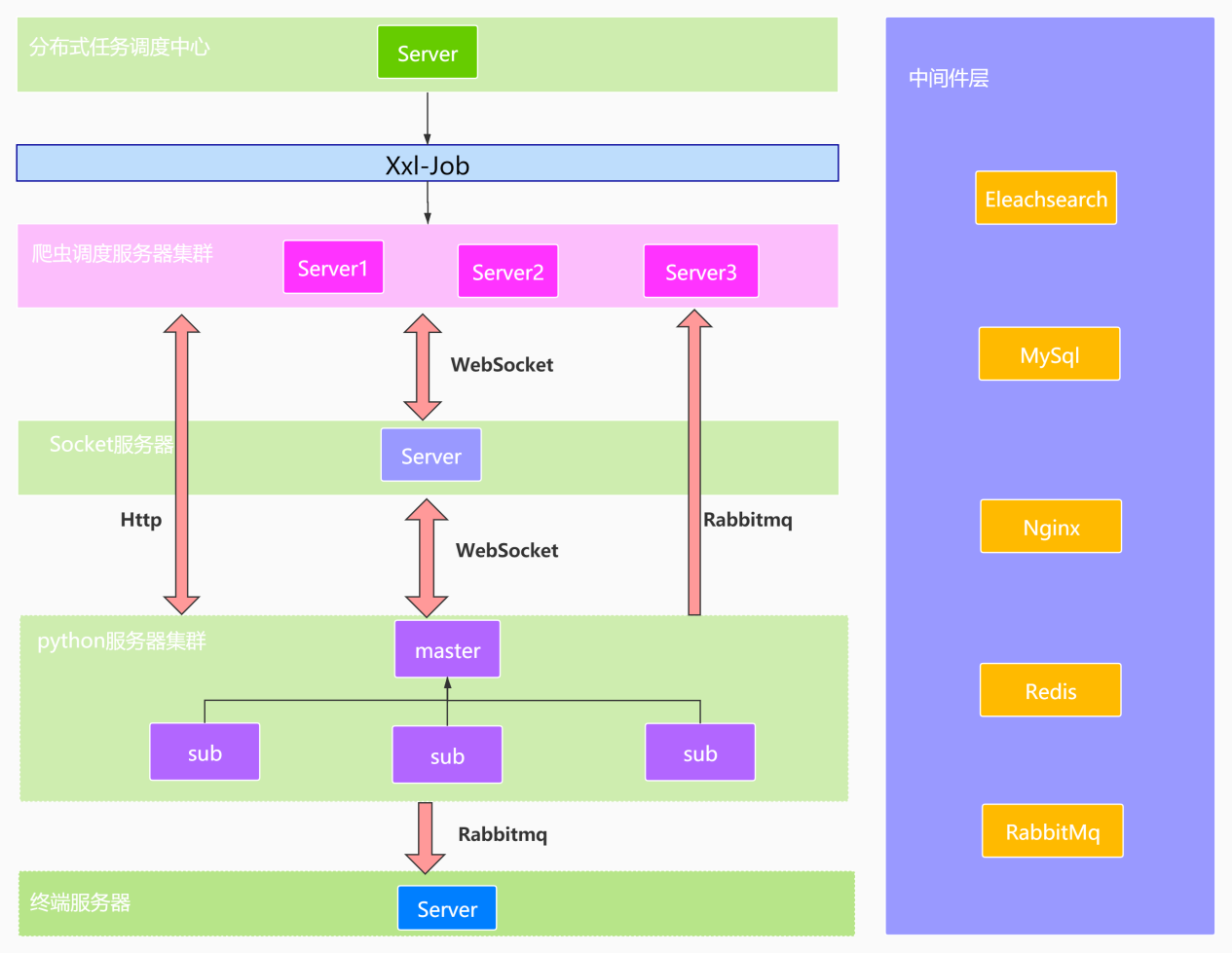
Springboot+mybatisplus+elasticsearch+redis+rabbitmq+mysql8+Netty+shardingJDBC+XXL-Job+springsecurity+docker+nginx

下面做具体说明,其中jdk基础版本为jdk8,项目中引用了很多jdk8的新特性，如lambda表达式，stream流式处理，Optional等实用快捷的功能。

核心框架为目前市场主流的springboot，版本为2.2.1.RELEASE，ORM框架采用mybatis增强版mybatisPlus开发。数据存储为mysql和elasticsearch，其中mysql存储系统关系型业务数据，elasticsearch作为全文检索的数据库，存储抓取的海量消息数据及爬虫日志数据。Redis用来作分布式缓存和计数器等功能，rabbitmq作为跨语言通信的消息中间件，作消息解耦。Netty实现websocket协议做服务器通信功能，shardingJDBC为mysql分表插件，xxl-job是一个分布式的定时任务调度中心，springsecurity作为权限校验的基本框架，项目基于容器化docker部署，nginx做请求转发，解决前端跨域问题。

## 2.2 技术架构

技术架构图



### 2.2.1 功能模块概述

#### 分布式任务调度中心（xxl-job）

分布式调度中心实质上是一个分布式定时任务框架，核心功能是对向爬虫服务器集群分布式的发送调度任务，根据配置的算法(一般为轮询算法)去调度对应的机器产生爬虫任务。本系统采用第三方开源框架XXL-JOB实现，有可视化的界面和灵活手动配置，使用比较方便。

#### **爬虫调度服务器集群（admin-web）**

爬虫调度服务器集群是将爬虫调度核心服务和监控服务部署到多台机器上，一方面产生接收分布式调度中心分发的调度任务，进行生产任务的组装，并转发给websocket服务器；另一方面接收python服务器集群返回的实时爬虫日志进行监控任务功能的处理；还会进行代理池监控，python服务器集群状态监控，中间件状态监控，调度监控等其他监控功能的工作。

#### **WebSocket服务器(spider-socket)**

websocket服务器实质上是爬虫调度服务器集群和python服务器集群通信的中间层，或者说桥梁，爬虫服务器集群将生产的调度任务通过websocket链路发送到websocket服务器上，python服务器集群也会连接到websocket服务器上，websocket服务器接收到生产任务后会根据优先级策略和拒绝策略发送给加入分组的并且满足条件的python服务器。

优先级策略是执行间隔时间最长的python服务器拥有最高优先级，首先加入的分组的ip也会有最高优先级。

拒绝策略是python服务器本身状态达到一定临界值时不能再接收或者处理爬虫任务时会返回一个拒绝状态或者即将不能处理状态，websocket服务器则会从分配列表中暂时剔除改python服务器。直到它重新恢复正常状态时通知对应状态再重新加入分配列表。

此外，websocket还会转发其他爬虫服务器集群的消息，比如实时爬虫日志，实时调度等消息。

#### **Python服务器集群**

Python服务器集群是由python搭建的一套服务器集群，集群结构为一个master搭载多台slave结构，master为中心结构，负责处理slave爬虫的日志，结果，去重，写快照等功能，还会监控每个slave的实时状态。Slave为各个独立的节点，专门负责执行爬虫任务。

Python服务器集群处理完成后将每次爬虫的日志通过rabbitmq发送给爬虫调度服务器集群进行处理，将去重后的结果通过rabbitmq发送给终端服务器。

此外，python服务器集群还会通过http接口添加和维护高质量代理。

#### **终端服务器(spider-server)**

终端服务器核心功能是接收python服务器集群的实时爬虫结果消息，分拣消息，推送消息的处理，以及权限控制的管理，终端上下线消息管理，标签类别管理等功能的处理。

终端服务器根据python服务器发送的实时爬虫结果消息通过websocket推送给前端进行渲染，并将数据持久化到mysql和es。如果觉得消息值得关注，可以从实时消息中进行筛选分拣，加入到分拣列表中，分拣操作是私有的。如果觉得分拣的消息可以直接发布了，就可以将分拣消息列表中的消息直接推送，推送操作是共享的。

终端服务器可以给账户分配对应的角色权限来控制不同的人展示不同的页面。

终端服务器处理一些其他的功能性管理，如上下线消息记录，标签类别管理等功能。

#### 上下线消息服务器(push-socket)

一个独立的socket项目，功能比较简单，专门处理终端上下线信息的es存储。

## 2.2.2 项目主线流程概述

1. 爬虫服务器集群中的服务器启动后初始化，从数据库中读取待运行的模板列表并加载到redis中，检测代理ip池。
2. 分布式调度中心XXL-JOB根据轮询机制向爬虫调度服务器集群发送调度任务；
3. 爬虫服务器集群中的服务器节点接收到调度任务后开始执行，如果有需要代理的模板则会从代理池中寻找对应的代理Ip，并将模板状态更新到redis中，组装完成生产任务后发送给websocket服务器。
4. websocket服务器接收到生产任务消息后，根据合适的策略分配给对应的python服务器，并更新模板状态到redis中。
5. python服务器集群接收到websocket的生成任务消息后开始执行任务，各个slave节点执行后统一返回给master处理结果，包括去重，写快照，写slave日志等。
6. python服务器将处理后的实时爬虫日志和爬虫结果发送给rabbitmq。
7. 爬虫调度服务器集群从rabbitmq中接收实时爬虫日志，持久化到es，并支持后面一系列的统计分析功能。终端服务器从rabbitmq中接收爬虫结果，实时转发给前端，并将数据持久化到mysql和es中。

# 数据库设计

1. **sys\_user 系统用户表 主要用来存储系统用户。**

CREATE TABLE `sys\_user` (

`user\_id` bigint(20) NOT NULL AUTO\_INCREMENT,

`username` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '用户名',

`password` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '密码',

`dept\_id` bigint(20) DEFAULT NULL COMMENT '部门ID',

`email` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '邮箱',

`phone` varchar(100) COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '手机号',

`status` tinyint(4) DEFAULT '1' COMMENT '状态 0：禁用 1：正常',

`maintainer` tinyint(1) DEFAULT '0' COMMENT '是否维护人员 0 否 1 是',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`user\_id`),

UNIQUE KEY `username` (`username`)

) ENGINE=InnoDB AUTO\_INCREMENT=1094 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='系统用户';

1. **sys\_user\_role 系统用户和用户角色关系映射表**

CREATE TABLE `sys\_user\_role` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`user\_id` bigint(20) DEFAULT NULL COMMENT '用户ID',

`role\_id` bigint(20) DEFAULT NULL COMMENT '角色ID',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=175 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='用户与角色对应关系';

1. **sys\_role 系统用户角色表**

CREATE TABLE `sys\_role` (

`role\_id` bigint(20) NOT NULL AUTO\_INCREMENT,

`role\_name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '角色名称',

`remark` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '备注',

`dept\_id` bigint(20) DEFAULT NULL COMMENT '部门ID',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

PRIMARY KEY (`role\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=1000 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='角色';

1. **sys\_role\_menu 用户角色和用户菜单关系映射表**

CREATE TABLE `sys\_role\_menu` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`role\_id` bigint(20) DEFAULT NULL COMMENT '角色ID',

`menu\_id` bigint(20) DEFAULT NULL COMMENT '菜单ID',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=171 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='角色与菜单对应关系';

1. **sys\_menu 系统菜单表**

CREATE TABLE `sys\_menu` (

`menu\_id` bigint(20) NOT NULL AUTO\_INCREMENT,

`parent\_id` bigint(20) DEFAULT NULL COMMENT '父菜单ID，一级菜单为0',

`name` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '菜单名称',

`url` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '菜单URL',

`perms` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '授权(多个用逗号分隔，如：user:list,user:create)',

`type` int(11) DEFAULT NULL COMMENT '类型 0：目录 1：菜单 2：按钮',

`icon` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '菜单图标',

`order\_num` int(11) DEFAULT NULL COMMENT '排序',

PRIMARY KEY (`menu\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=66 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='菜单管理';

1. **template 爬虫网页模板表 用来存储需要爬取的网页消息**

CREATE TABLE `template` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`page\_site` varchar(500) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '页面起始网址',

`domain\_name` varchar(250) NOT NULL DEFAULT '127.0.0.1',

`retry\_times` int(2) DEFAULT '15' COMMENT '重试失败次数',

`real\_site` text CHARACTER SET utf8 COLLATE utf8\_general\_ci COMMENT '抓取真正网址',

`title` varchar(500) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '页面标题，默认为网页标题',

`filter\_by` int(11) DEFAULT '1' COMMENT '去重方式 ：1 通过链接，2通过文字',

`status` tinyint(4) DEFAULT '1' COMMENT '当前页面监控状态 0停止 1运行中 2出错',

`interval\_time` int(10) DEFAULT '20000' COMMENT '页面爬取间隔，单位为毫秒，默认20秒钟爬取一次',

`channel` varchar(250) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '同一个标题可能有多个不同的频道',

`category` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '分类',

`encoding` varchar(20) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT 'UTF-8',

`create\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '添加监控页面时间',

`priority` int(11) DEFAULT '50' COMMENT '页面优先级',

`weforeign` tinyint(1) DEFAULT '0',

`dept\_id` bigint(20) DEFAULT NULL COMMENT '所属部门',

`update\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,

`special` tinyint(1) DEFAULT '0',

`random\_query` tinyint(1) DEFAULT '1' COMMENT '拼接URL随机参数',

`timeout` int(10) DEFAULT '7' COMMENT '超时时间 默认7s',

`detail\_encoding` varchar(20) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT 'UTF-8',

`remark` varchar(500) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '备注',

`deleted` int(11) DEFAULT '0',

`encrypt\_type` varchar(20) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '解密类型',

`spider\_temp\_id` bigint(20) DEFAULT NULL COMMENT '特殊爬虫模版的id',

`proxy\_level` int(11) DEFAULT '0',

`expire\_time` timestamp NULL DEFAULT NULL COMMENT '模版预计过期时间',

`create\_user` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '操作用户',

`update\_user` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '操作用户',

`delete\_user` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '操作用户',

`delete\_time` timestamp NULL DEFAULT NULL COMMENT '删除时间',

`stop\_reason` varchar(255) DEFAULT NULL COMMENT '停止原因',

`stop\_time` datetime DEFAULT NULL COMMENT '停止时间',

PRIMARY KEY (`id`),

KEY `page\_site\_idx` (`page\_site`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=1477 DEFAULT CHARSET=utf8 COMMENT='爬虫模版';

1. **template\_rule 网页模板对应的爬取规则 和template表的id关联**

CREATE TABLE `template\_rule` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`temp\_id` bigint(20) NOT NULL COMMENT '模版id',

`type` int(11) DEFAULT '1' COMMENT '内容类型，1文本，2json，3jsonp',

`headers` text CHARACTER SET utf8 COLLATE utf8\_general\_ci COMMENT '头信息',

`method` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '方法，post、get',

`params` text CHARACTER SET utf8 COLLATE utf8\_general\_ci COMMENT '附带参数',

`rules\_json` varchar(500) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '规则，json',

`is\_detail` tinyint(1) DEFAULT '0' COMMENT '是否为详情页面的规则',

`spider\_temp\_id` bigint(20) DEFAULT NULL COMMENT '特殊爬虫模版的id',

`encrypt\_type` varchar(20) CHARACTER SET utf8 COLLATE utf8\_general\_ci DEFAULT NULL COMMENT '解密类型',

PRIMARY KEY (`id`),

KEY `temp\_id\_idx` (`temp\_id`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=3791 DEFAULT CHARSET=utf8 COMMENT='爬虫模版规则';

1. **ip\_info 代理ip池 存储代理Ip信息**

CREATE TABLE `ip\_info` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`ip` varchar(100) DEFAULT NULL COMMENT 'ip:端口、113.120.61.166:22989',

`is\_foreign` tinyint(1) DEFAULT '0',

`area` varchar(50) DEFAULT '中国' COMMENT '地区',

`platform` varchar(50) NOT NULL COMMENT '代理平台',

`delay` int(11) DEFAULT NULL COMMENT 'ip 延迟 ms',

`check\_times` int(11) DEFAULT NULL COMMENT '检查次数',

`expire\_time` timestamp NULL DEFAULT NULL COMMENT '过期时间',

`check\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '检验时间',

`create\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',

`proxy\_level` int(11) DEFAULT '0',

`deleted` int(11) DEFAULT '0',

`is\_valid` tinyint(2) DEFAULT '1' COMMENT '是否有效',

`type` varchar(50) DEFAULT 'http' COMMENT '类型 http/ https/ socket',

`extra` varchar(500) DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `ip\_info\_ip\_uindex` (`ip`),

KEY `check\_time\_idx` (`check\_time`)

) ENGINE=InnoDB AUTO\_INCREMENT=1739635 DEFAULT CHARSET=utf8 COMMENT='代理ip 信息';

1. **invaded\_ip\_info 失效代理ip的历史记录**

CREATE TABLE `invaded\_ip\_info` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`ip` varchar(100) DEFAULT NULL COMMENT 'ip:端口、113.120.61.166:22989',

`whe\_foreign` tinyint(1) DEFAULT '0',

`area` varchar(50) DEFAULT NULL COMMENT '地区',

`platform` varchar(50) DEFAULT NULL COMMENT '代理平台',

`delay` int(11) DEFAULT NULL COMMENT 'ip 延迟 ms',

`check\_times` int(11) DEFAULT NULL COMMENT '检查次数',

`expire\_time` timestamp NULL DEFAULT NULL COMMENT '过期时间',

`check\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '检验时间',

`create\_time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',

`life\_time` bigint(10) DEFAULT NULL COMMENT '存活时长 单位s',

`proxy\_level` int(11) DEFAULT '0',

`deleted` int(11) DEFAULT '0',

`whe\_valid` tinyint(2) DEFAULT '0' COMMENT '是否有效 0-无效 1-有效',

`type` varchar(50) DEFAULT NULL COMMENT '类型 http/ https/ socket',

`extra` varchar(500) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `invaded\_ip\_info\_check\_time\_index` (`check\_time`),

KEY `invaded\_ip\_info\_ip\_index` (`ip`)

) ENGINE=InnoDB AUTO\_INCREMENT=1749337 DEFAULT CHARSET=utf8 COMMENT='代理ip 信息';

1. **spider\_category 爬虫消息类别**

CREATE TABLE `spider\_category` (

`cate\_id` bigint(10) NOT NULL AUTO\_INCREMENT COMMENT '消息类别管理',

`category\_name` varchar(40) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '类别名称',

`category\_color` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '类别颜色',

PRIMARY KEY (`cate\_id`),

UNIQUE KEY `spider\_category\_category\_name\_uindex` (`category\_name`)

) ENGINE=InnoDB AUTO\_INCREMENT=47 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='消息类别管理';

1. **spider\_count\_job 每小时爬虫执行任务统计**

CREATE TABLE `spider\_count\_job` (

`id` bigint(30) NOT NULL AUTO\_INCREMENT COMMENT '主键Id',

`ip` varchar(50) DEFAULT NULL COMMENT 'ip地址',

`all\_count` bigint(30) DEFAULT NULL COMMENT '总任务数量 ',

`success\_count` bigint(20) DEFAULT NULL COMMENT '成功数量',

`fail\_count` bigint(20) DEFAULT NULL COMMENT '失败数量',

`success\_rate` varchar(15) DEFAULT NULL COMMENT '成功率',

`temp` bigint(20) DEFAULT NULL COMMENT '模板数量',

`start\_time` datetime DEFAULT NULL COMMENT '开始统计时间',

`end\_time` datetime DEFAULT NULL COMMENT '结束时间',

`insert\_time` datetime DEFAULT CURRENT\_TIMESTAMP COMMENT '入库时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=15616 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci COMMENT='每小时执行任务统计';

1. **spider\_label 爬虫标签库**

CREATE TABLE `spider\_label` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`label\_name` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '标签名称',

`label\_color` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '标签颜色配置',

PRIMARY KEY (`id`),

UNIQUE KEY `spider\_label\_label\_name\_uindex` (`label\_name`)

) ENGINE=InnoDB AUTO\_INCREMENT=189 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='爬虫消息标签库';

1. **spider\_message\_2021\_0 爬虫消息表，用来存储抓取到的爬虫信息，只做存储备份作用，全文检索功能用es做的。 分表存储， 1年2张表，**

**如spider\_message\_2021\_0，spider\_message\_2021\_1这样的格式。**

CREATE TABLE `spider\_message\_2021\_0` (

`id` bigint(70) NOT NULL COMMENT '主键id',

`msg\_id` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL,

`task\_id` varchar(60) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '分发任务id',

`temp\_id` int(10) DEFAULT NULL COMMENT '模板id',

`dept\_id` int(10) DEFAULT NULL COMMENT '部门id',

`title` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci COMMENT '爬虫内容标题',

`url` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci COMMENT '爬虫内容对应的url',

`source` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT '未知来源' COMMENT '网站来源',

`category` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT '文章' COMMENT '消息类型 1.文章 2.快讯 3.楼市 等',

`time` datetime DEFAULT NULL COMMENT '爬虫爬取的日期 ',

`remark` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '备注',

`channel` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '同一个标题可能有多个不同的频道',

`insert\_time` datetime DEFAULT CURRENT\_TIMESTAMP COMMENT '入库时间',

PRIMARY KEY (`id`),

UNIQUE KEY `spider\_message\_msg\_id\_uindex` (`msg\_id`) USING BTREE,

KEY `spider\_message\_time\_index` (`time`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci ROW\_FORMAT=DYNAMIC;

1. **spider\_message\_abnormal 消息异常记录表**

CREATE TABLE `spider\_message\_abnormal` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`ex\_type` int(1) DEFAULT NULL COMMENT '消息异常类型 1.category为空 2.category非法类型',

`ex\_remark` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '消息异常描述',

`spider\_message` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci COMMENT '不正常消息',

`insert\_time` datetime DEFAULT CURRENT\_TIMESTAMP COMMENT '入库时间',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5269 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='异常消息记录';

1. **spider\_message\_monitor 消息最后更新时间记录表**

CREATE TABLE `spider\_message\_monitor` (

`id` bigint(10) NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`source` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '消息来源',

`category` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '消息类型',

`channel` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '消息频道',

`dispatch` int(1) DEFAULT NULL COMMENT '是否调度 0.是 1.否',

`auto\_push` int(1) DEFAULT '1' COMMENT '是否自动推送， 0-是 1-否',

`site` text COLLATE utf8mb4\_unicode\_ci COMMENT '对应链接网址',

`update\_time` datetime DEFAULT NULL COMMENT '最后更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `spider\_message\_monitor\_channel\_source\_category\_uindex` (`channel`,`source`,`category`)

) ENGINE=InnoDB AUTO\_INCREMENT=16829 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='消息来源监控';

**16.spider\_message\_push 推送消息记录表**

CREATE TABLE `spider\_message\_push` (

`id` bigint(70) NOT NULL AUTO\_INCREMENT COMMENT '主键id',

`msg\_id` varchar(255) COLLATE utf8mb4\_unicode\_ci NOT NULL,

`task\_id` varchar(60) COLLATE utf8mb4\_unicode\_ci NOT NULL COMMENT '分发任务id',

`temp\_id` int(10) DEFAULT NULL COMMENT '模板id',

`dept\_id` int(10) DEFAULT NULL COMMENT '部门id',

`title` text COLLATE utf8mb4\_unicode\_ci COMMENT '爬虫内容标题',

`url` text COLLATE utf8mb4\_unicode\_ci COMMENT '爬虫内容对应的url',

`source` varchar(50) COLLATE utf8mb4\_unicode\_ci DEFAULT '未知来源' COMMENT '网站来源',

`category` varchar(20) COLLATE utf8mb4\_unicode\_ci DEFAULT '文章' COMMENT '消息类型 1.文章 2.快讯 3.楼市 等',

`time` datetime DEFAULT NULL COMMENT '爬虫爬取的日期',

`remark` varchar(100) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '备注',

`channel` varchar(30) COLLATE utf8mb4\_unicode\_ci DEFAULT NULL COMMENT '同一个标题可能有多个不同的频道',

`env` varchar(30) COLLATE utf8mb4\_unicode\_ci NOT NULL DEFAULT 'prod' COMMENT '环境变量 prod 生产环境 dev 开发人员推送',

`insert\_time` datetime DEFAULT CURRENT\_TIMESTAMP COMMENT '入库时间',

PRIMARY KEY (`id`),

UNIQUE KEY `spider\_message\_push\_pk` (`msg\_id`,`env`)

) ENGINE=InnoDB AUTO\_INCREMENT=50635 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci COMMENT='推送爬虫消息列表';

1. xxl\_job开头的表均为xxl\_job所需的表。具体可看官网说明。

<https://www.xuxueli.com/xxl-job/>

# 功能模块详细设计

### 4.1 spider-common项目(基础依赖项目)

Spider-common主要是admin-web项目，spider-socker项目以及spider-server项目的基础依赖项目，将基础或者公有的工具类做了统一封装。比如netty服务端和客户端的实现，redis工具类的封装，共用的实体类等。

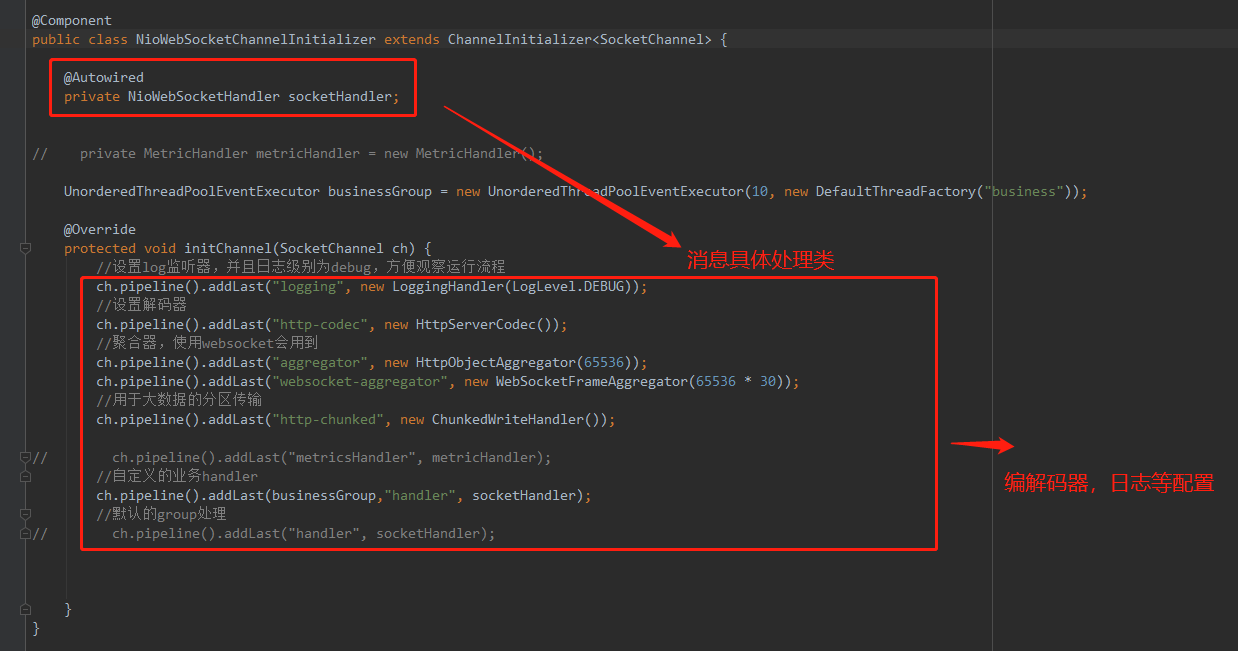
#### 4.1.1 netty实现websocket服务器核心代码

1.主入口类 NioWebSocketServer.java



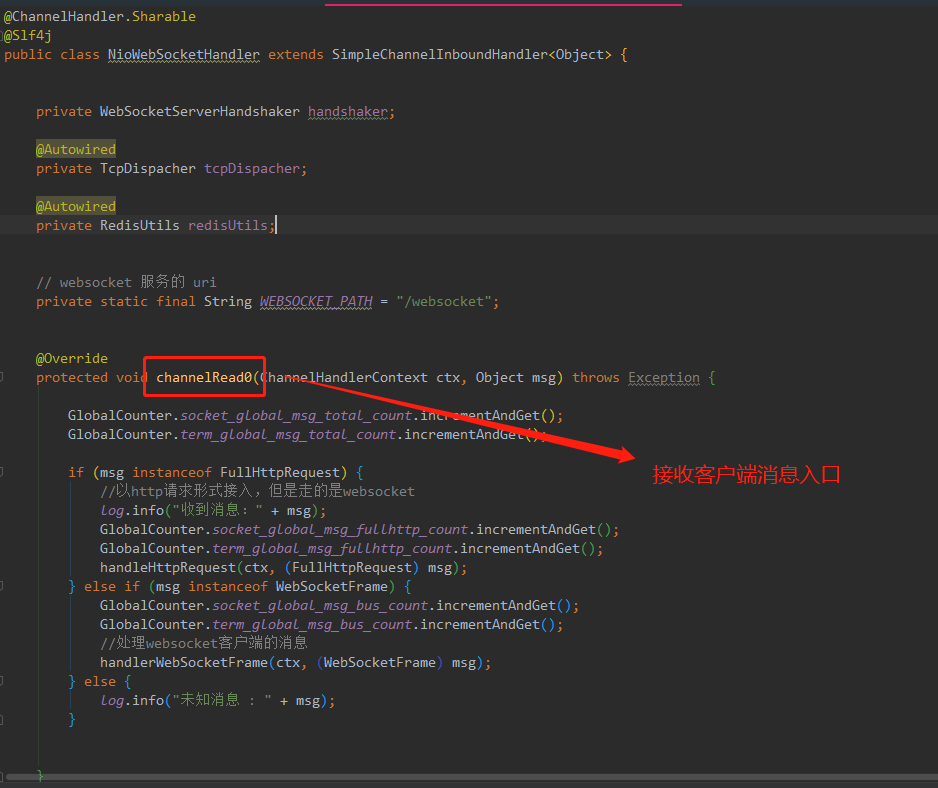
该类主要是绑定netty的bossGroup和workGroup，并加载启动对应的配置类。

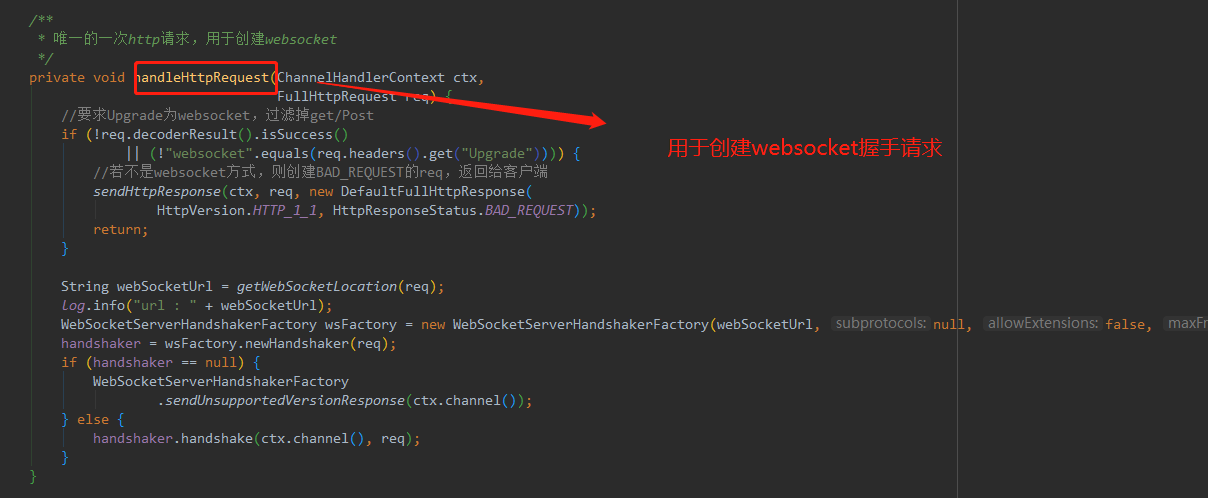
2.服务器及handler配置类 NioWebSocketChannelInitializer.java

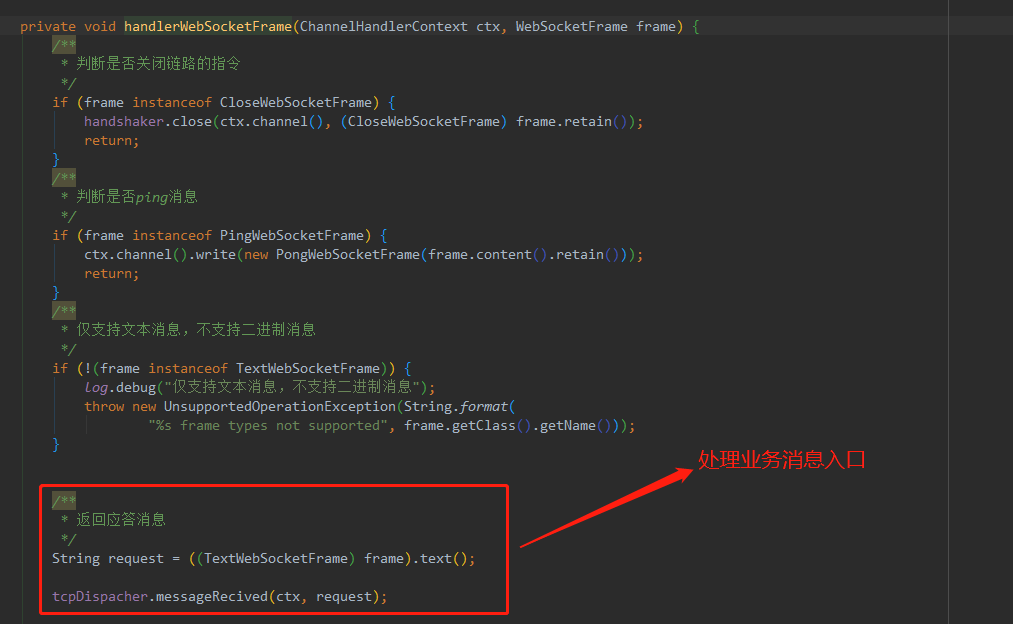


主要配置消息日志，编辑码器，自定义handler等handler的配置项

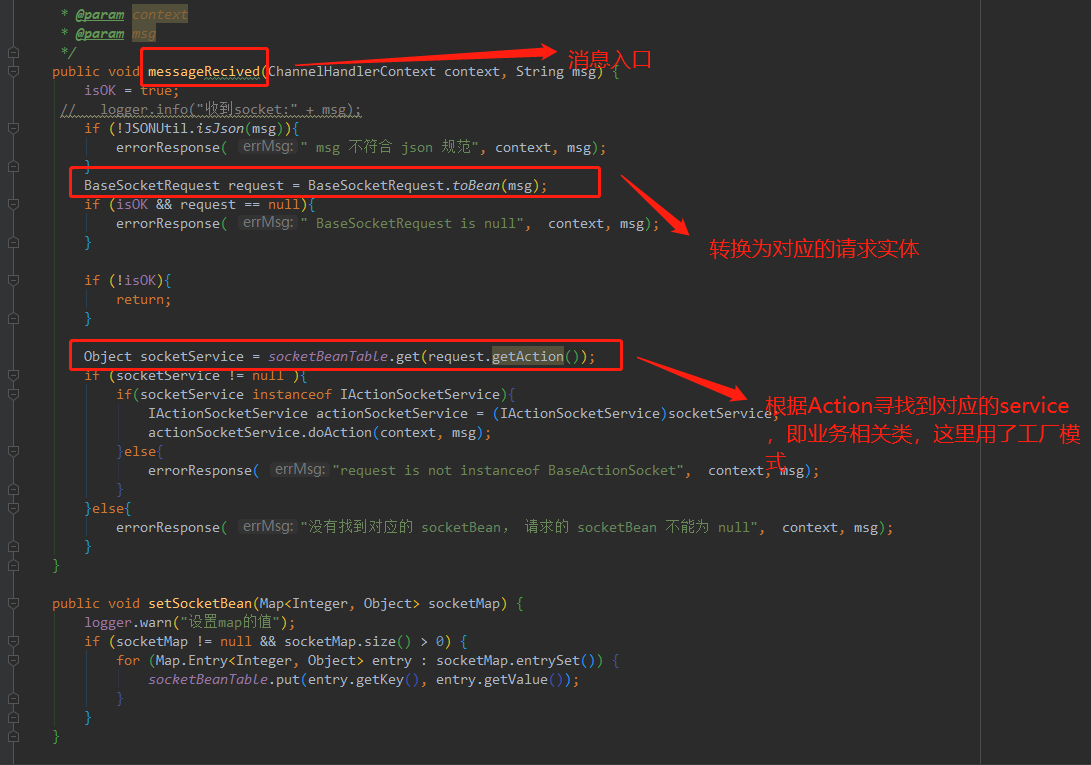
3.具体消息处理逻辑类 NioWebSocketHandler.java







1. 业务消息处理入口 TcpDispacher.java

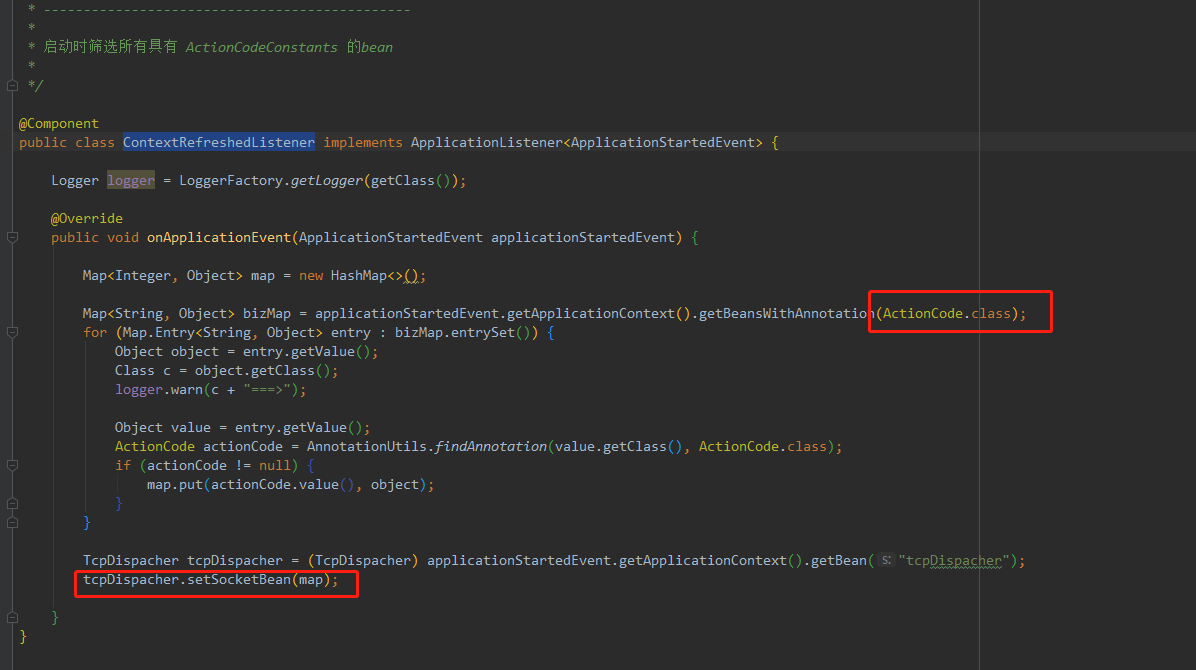


1. 业务相关类的封装过程

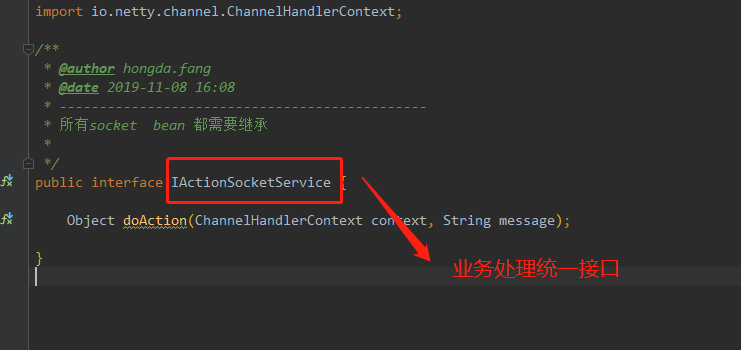
①注解ActionCode



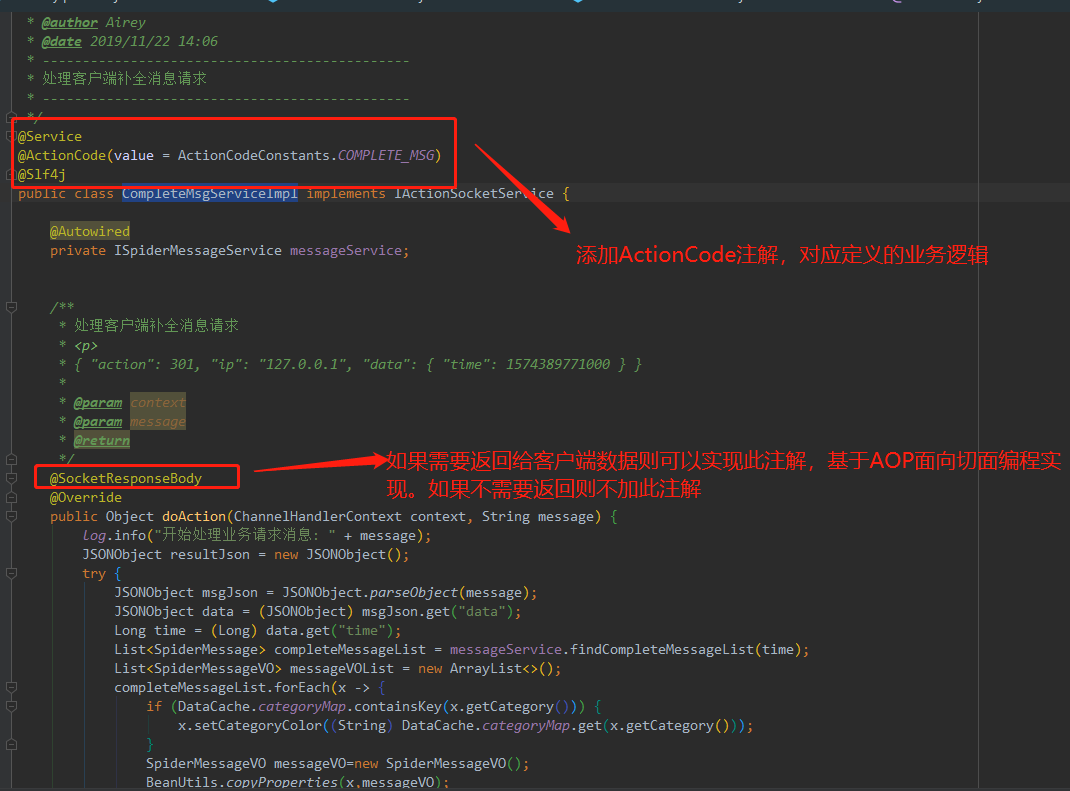
②ContextRefreshedListener.java 项目初始化时扫描添加了注解ActionCode的类，并加入工厂中。



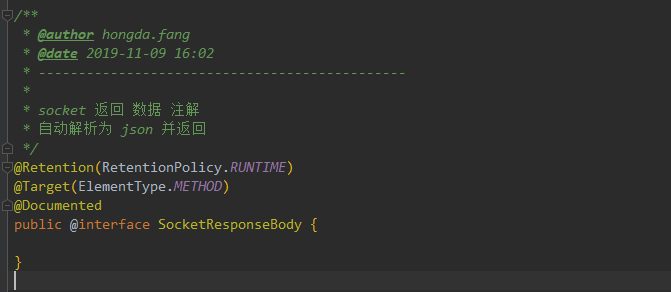
③ 业务处理统一接口 IActionSocketService.java



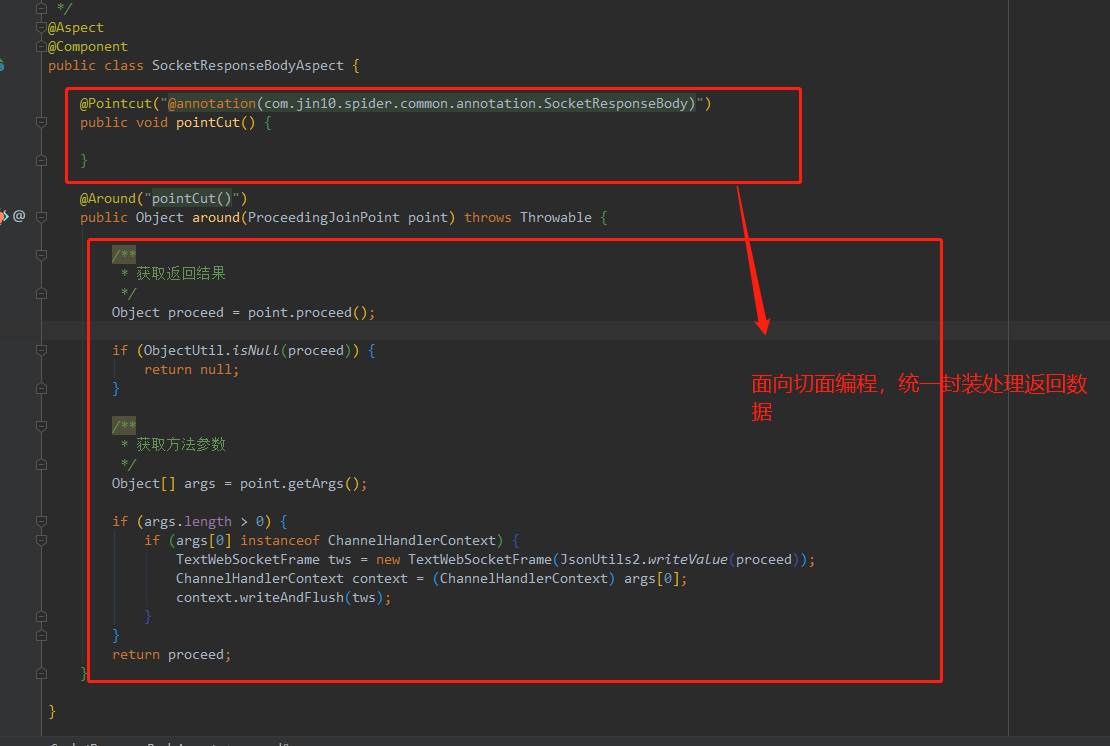
示例实现类 CompleteMsgServiceImpl.java



④ SocketResponseBody注解，用于统一封装返回给客户端的数据



⑤ SocketResponseBodyAspect.java

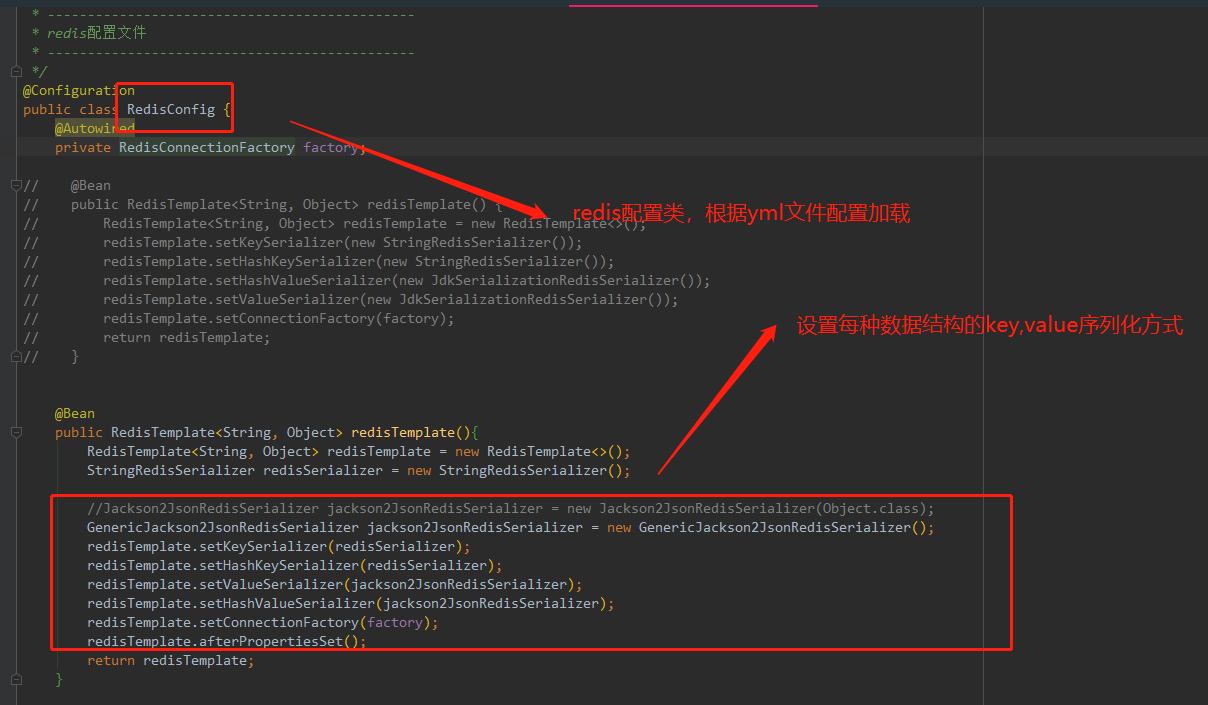


⑥ ChannelSupervise.java

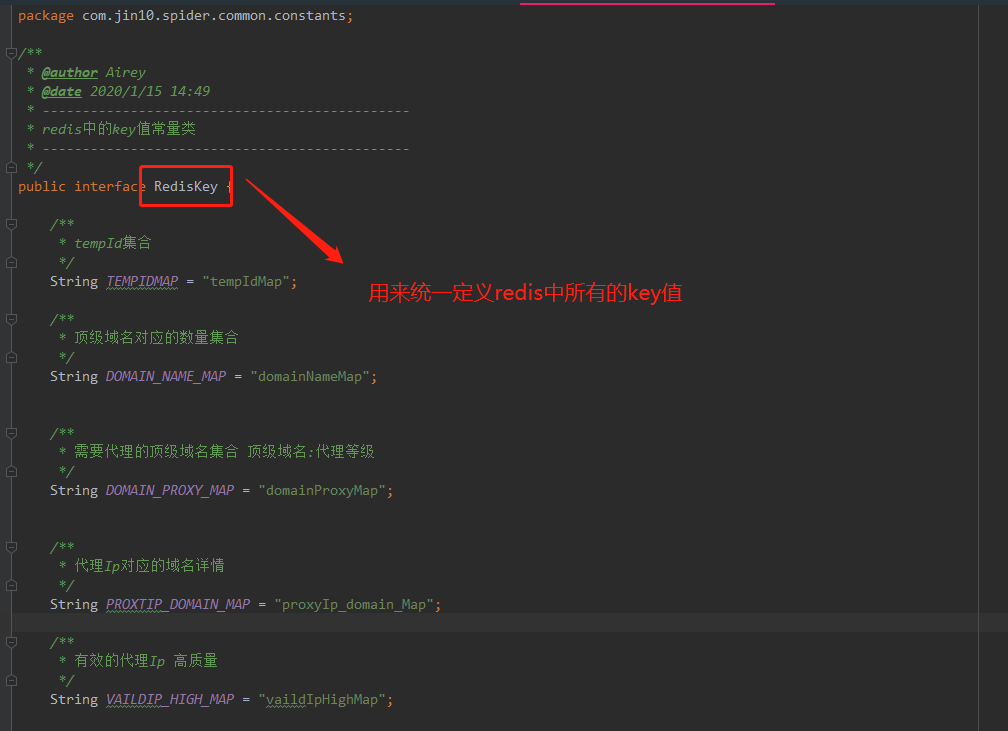


#### 4.1.2 redis配置及工具类

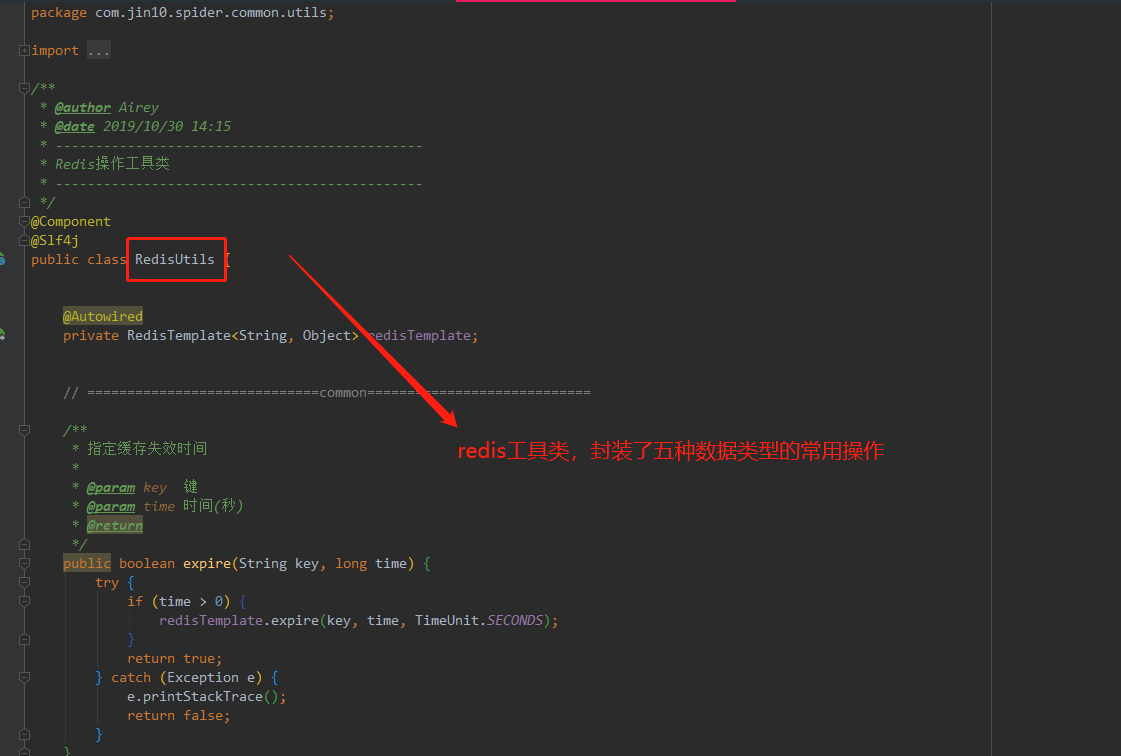
1. RedisConfig.java redis配置类



2.RedisKey.java redis中key值的统一定义入口



3.RedisUtils redis工具类的封装

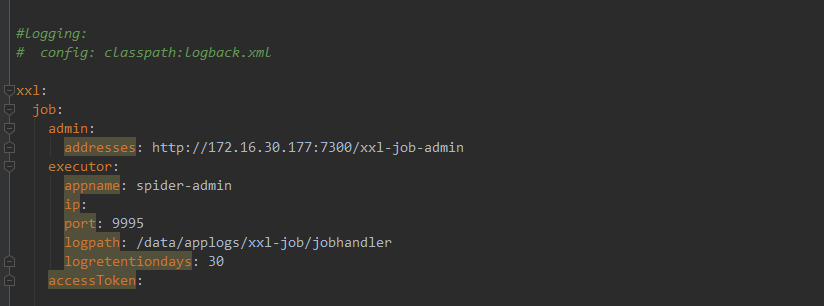


### 4.2. admin-web项目(爬虫调度核心)

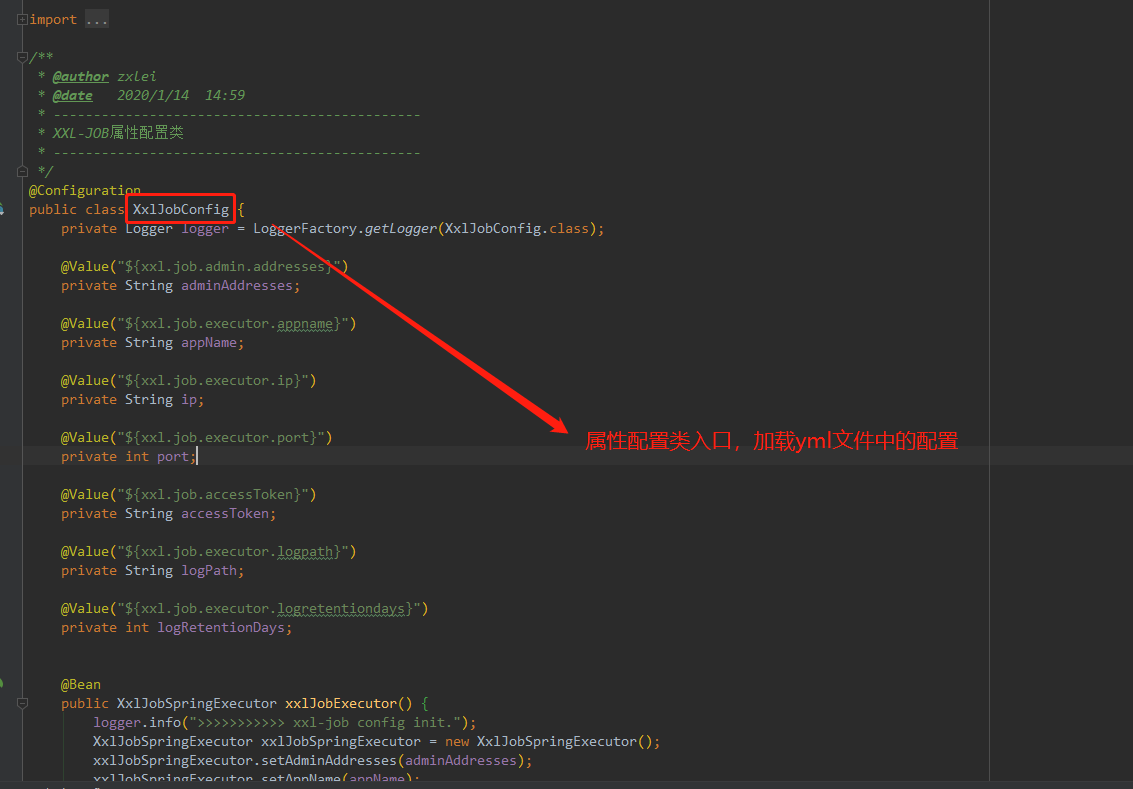
#### 4.2.1 爬虫调度核心逻辑

**一．Xxl-job的配置**

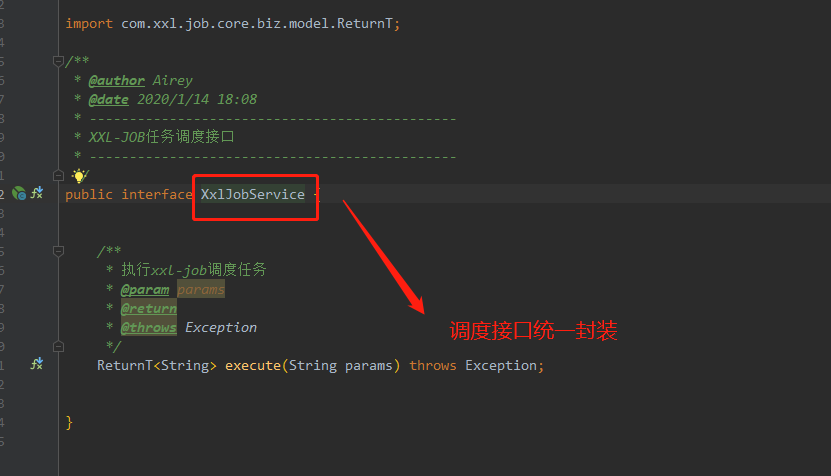
1.yml文件配置内容



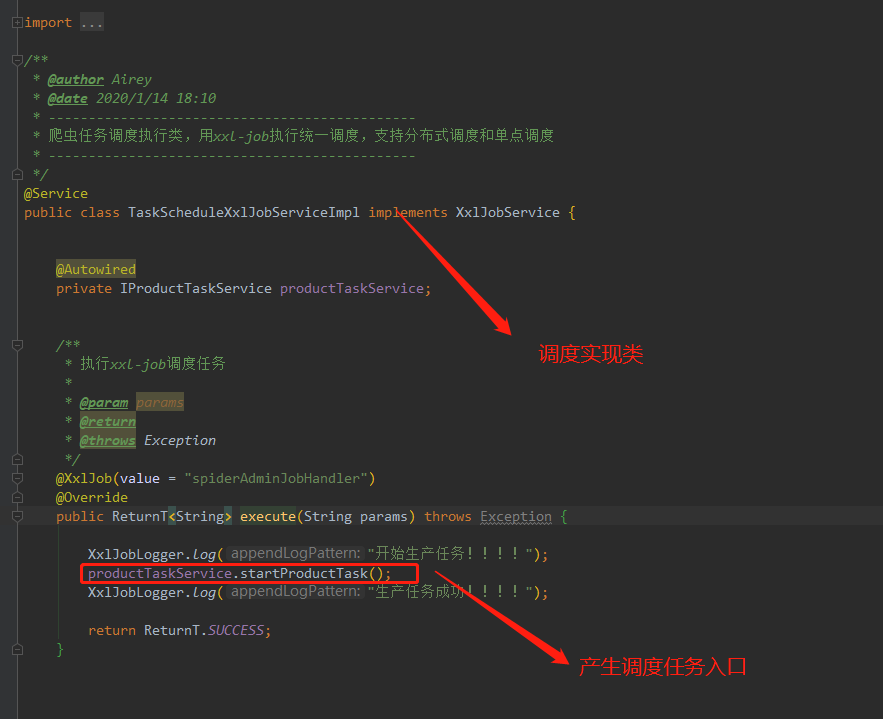
2.XxlJobConfig.java xxl-job的属性配置加载



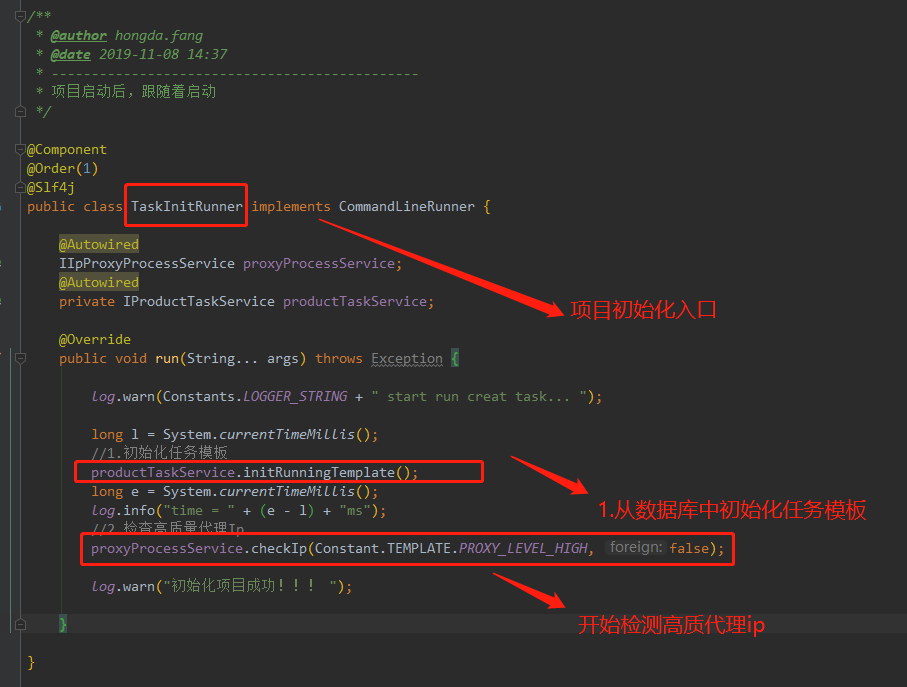
1. XxlJobService.java 任务调度接口统一封装



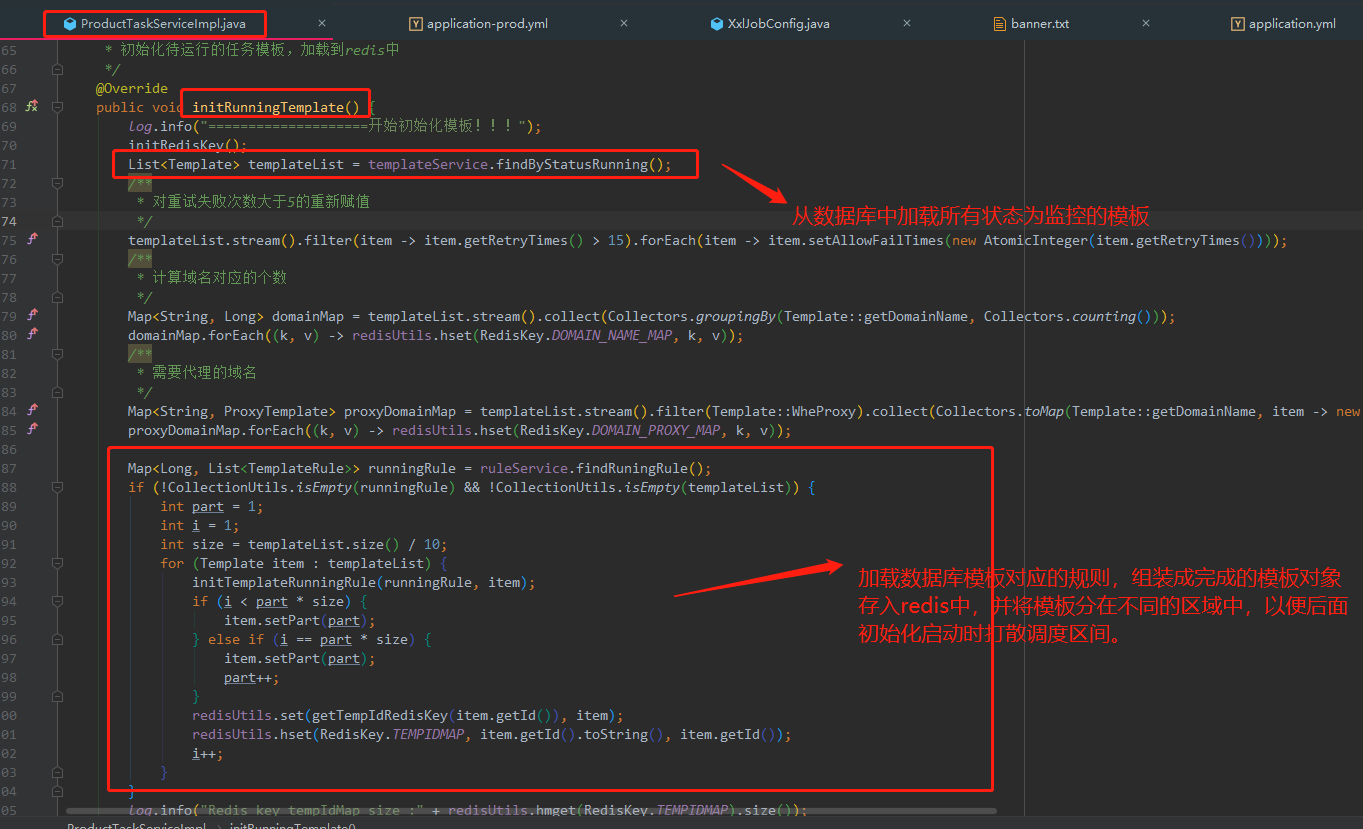
1. TaskScheduleXxlJobServiceImpl.java 调度实现类



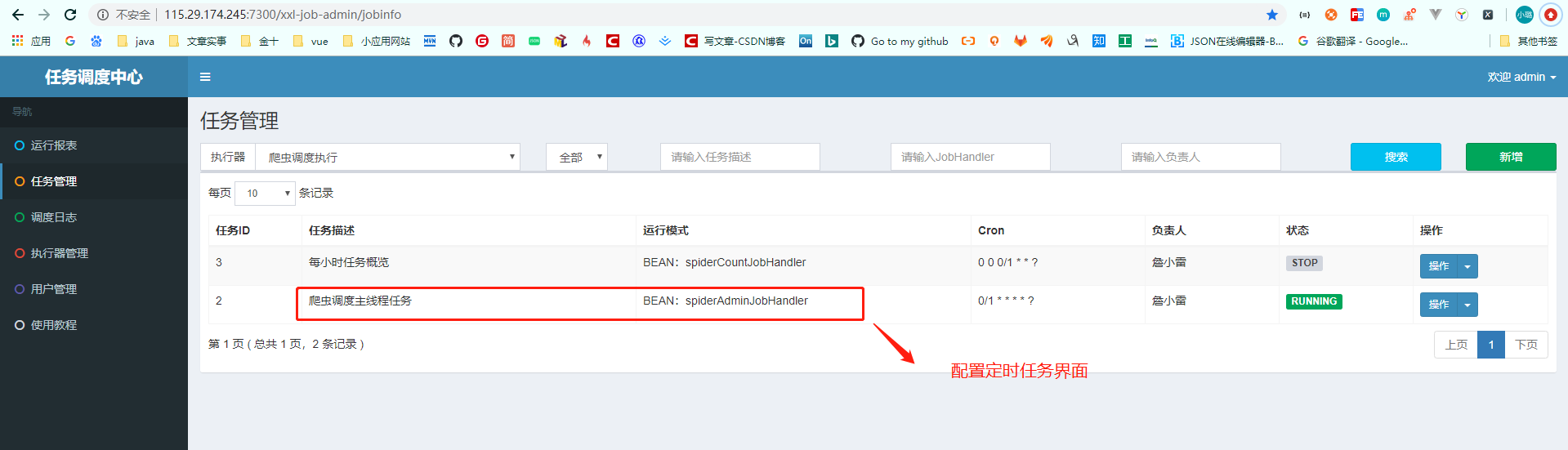
1. **admin-web项目初始化过程**
2. TaskInitRunner.java 项目初始化入口

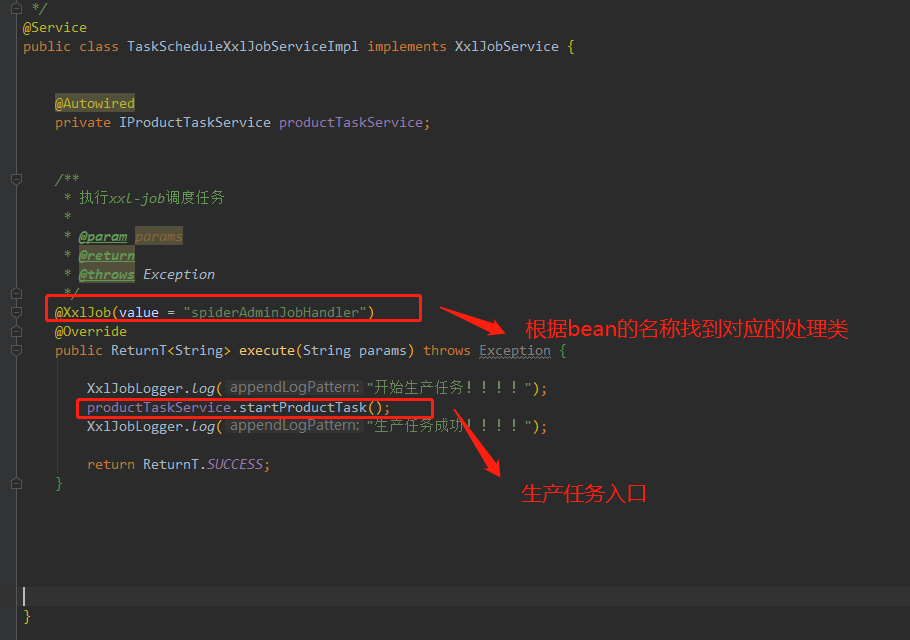


1. ProductTaskServiceImpl.java 生产任务逻辑处理主类

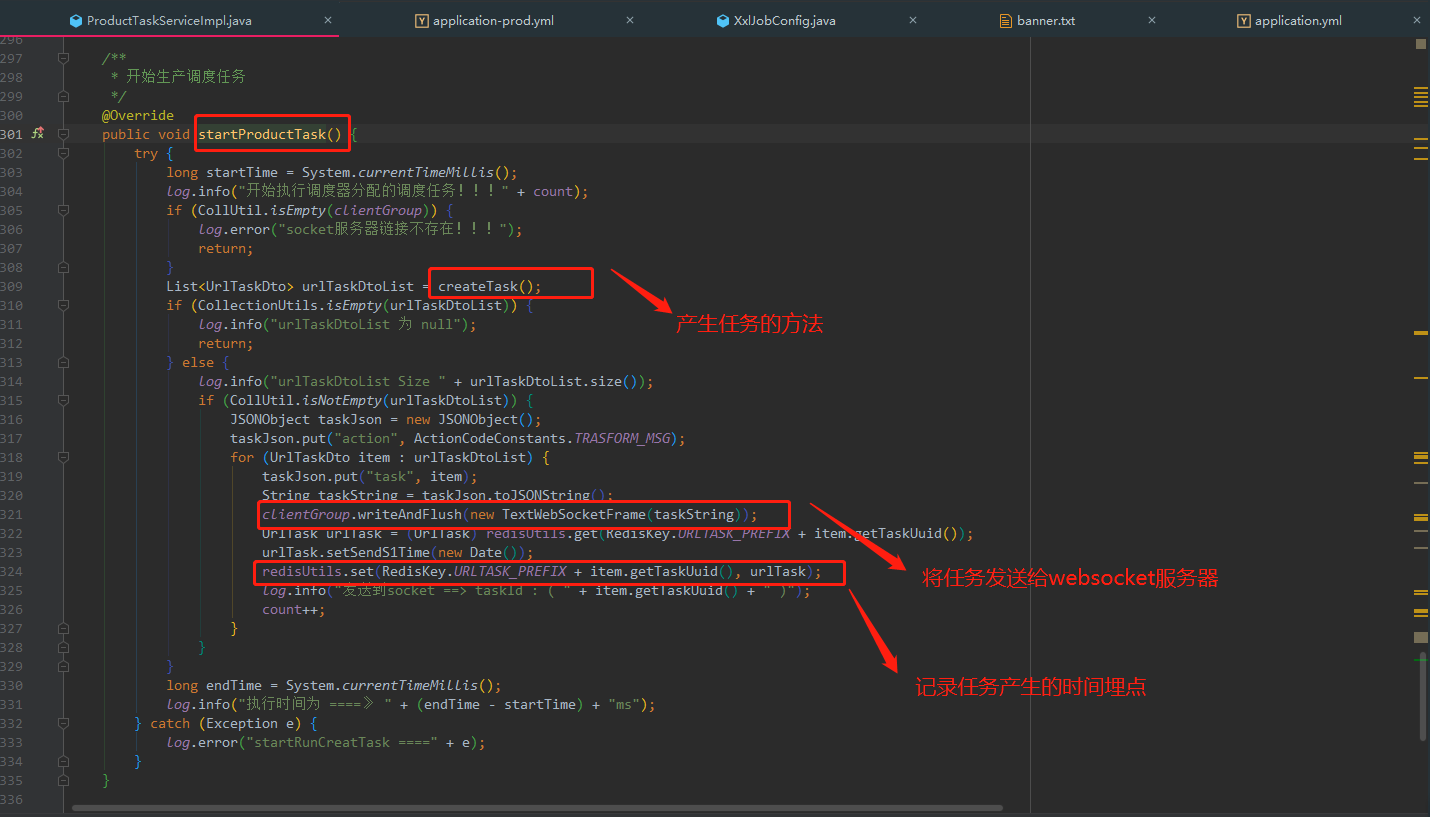


1. **任务调度，产生任务过程**





ProductTaskServiceImpl.java

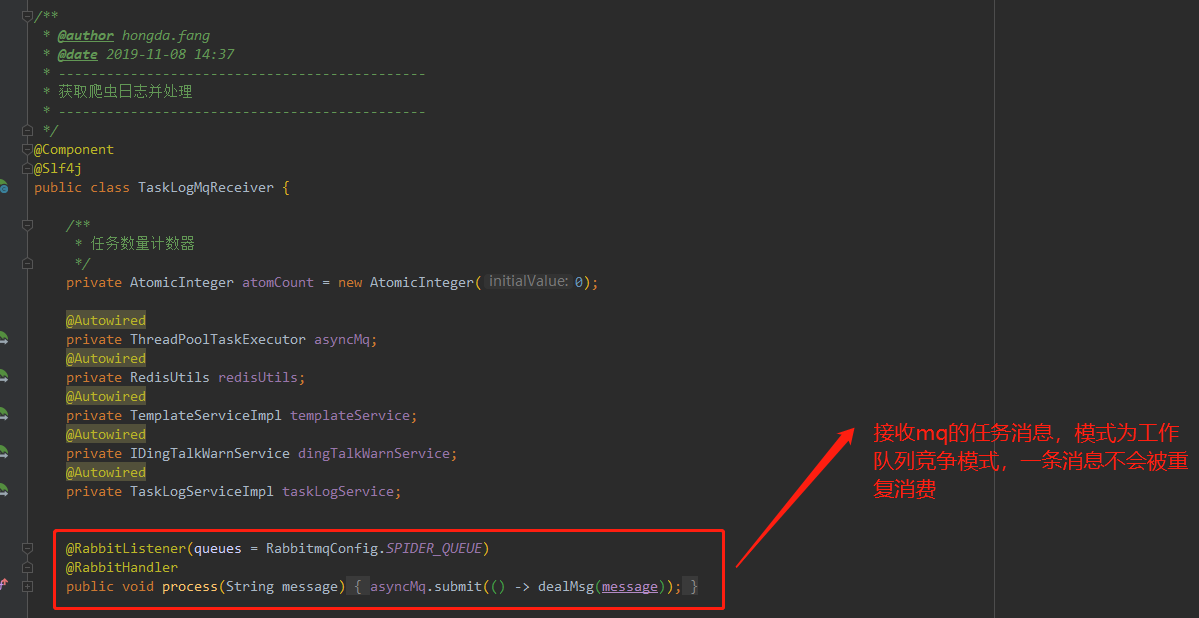






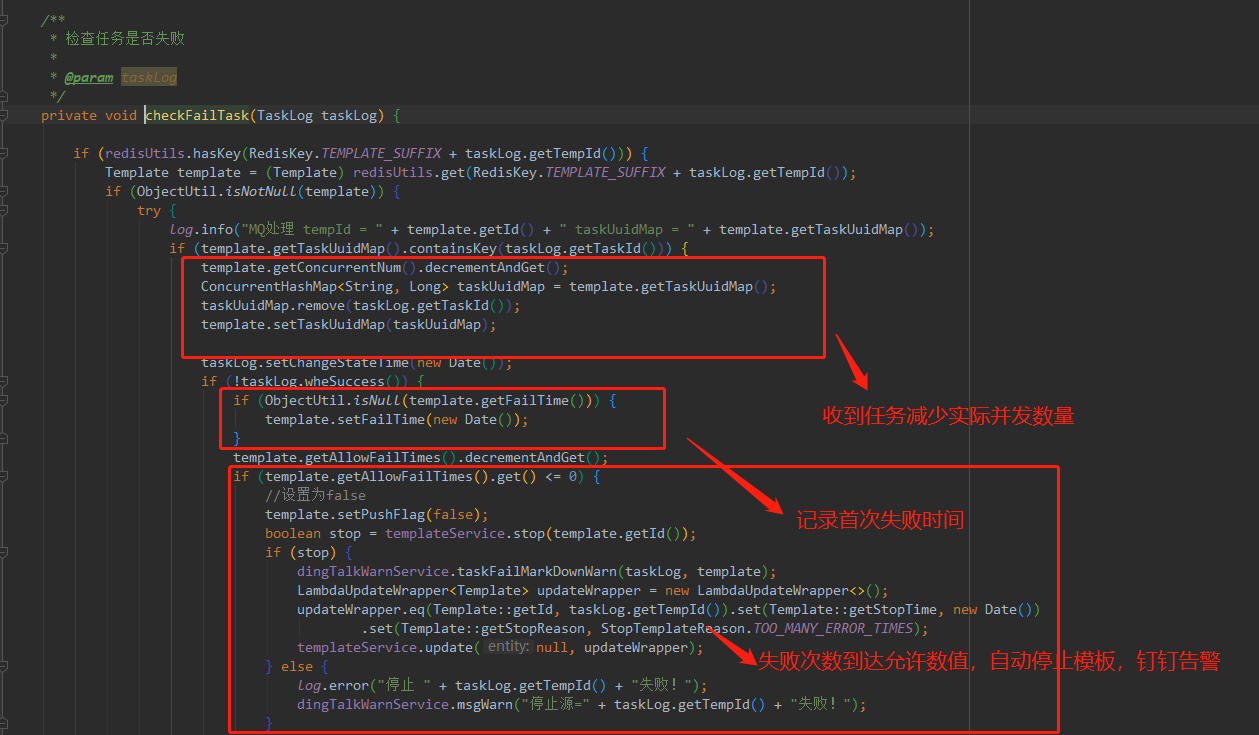
#### 4.2.2 爬虫日志处理核心逻辑

核心处理类 TaskLogMqReceiver.java







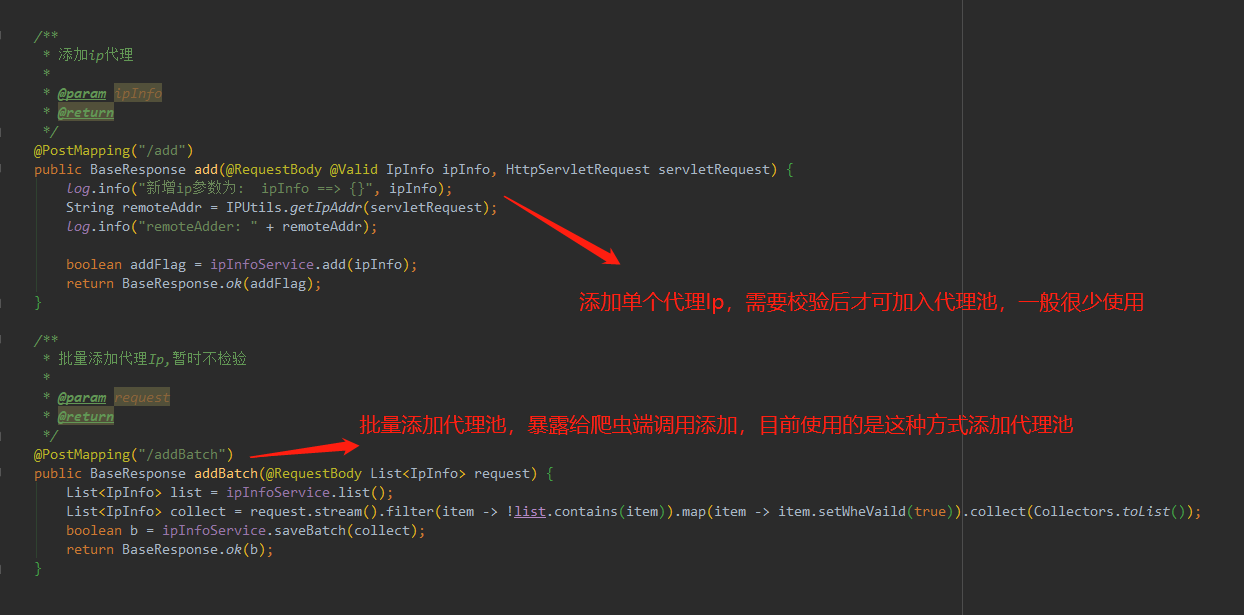






#### 4.2.3 代理池管理及校验逻辑

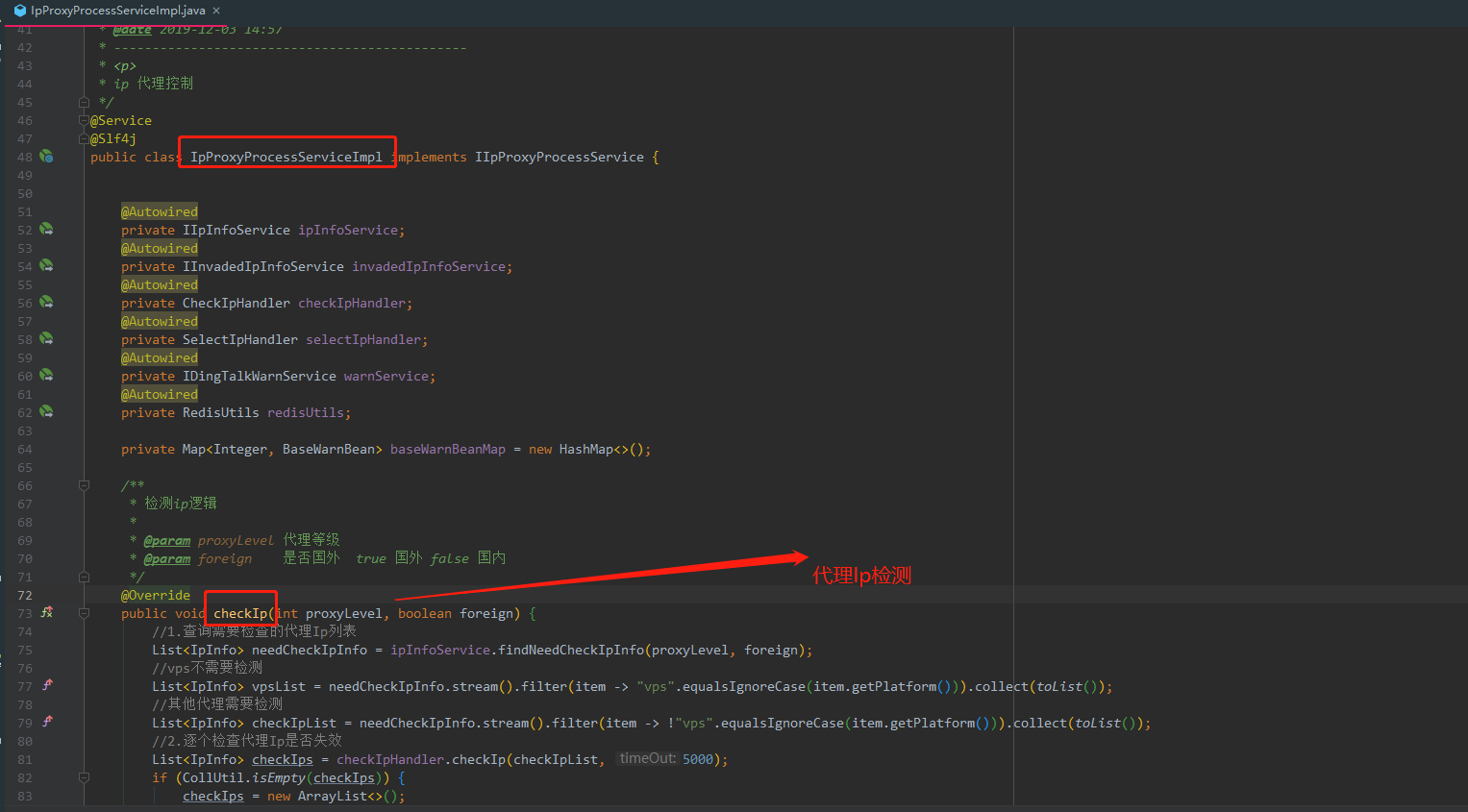
IpInfoController.java 代理ip管理信息

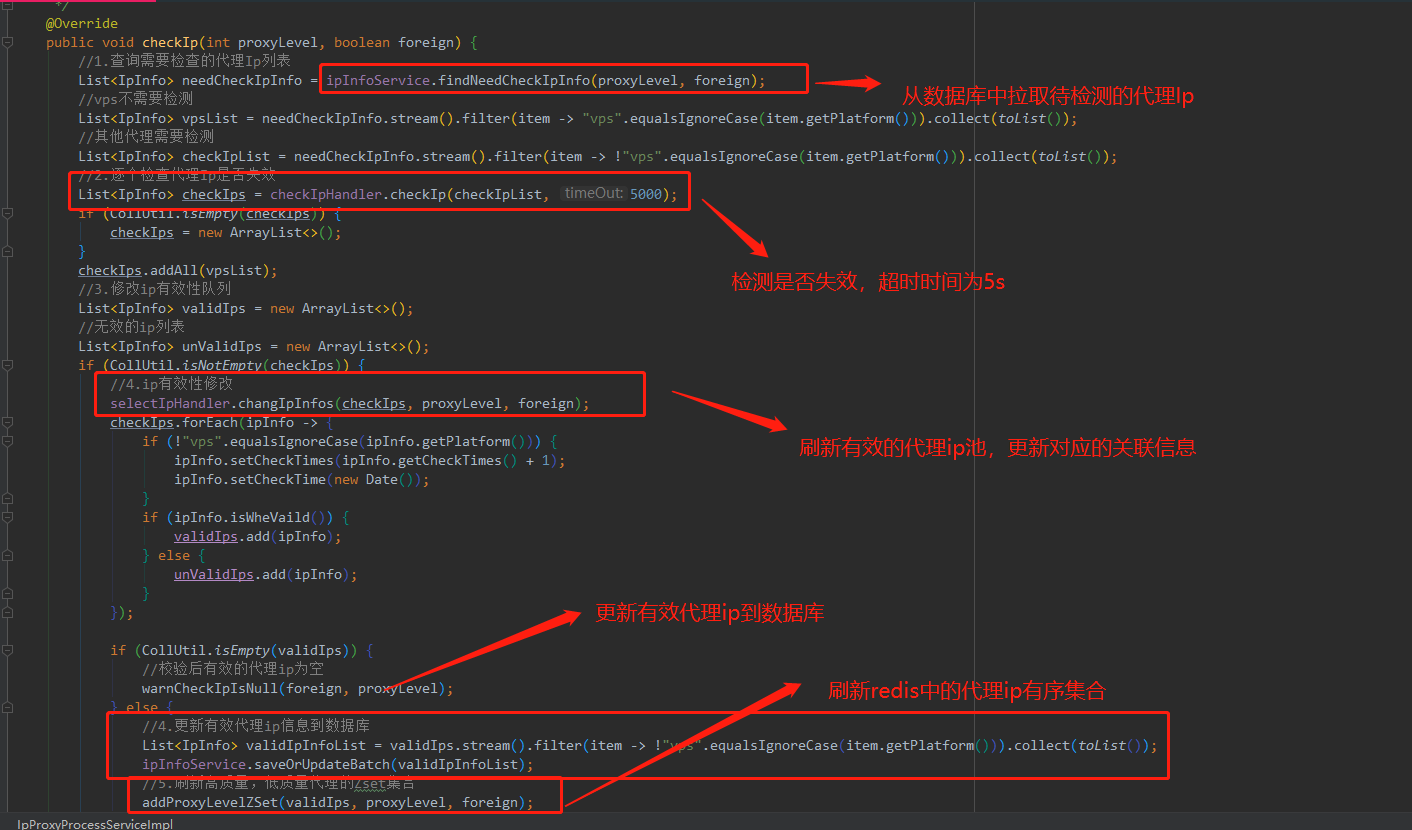


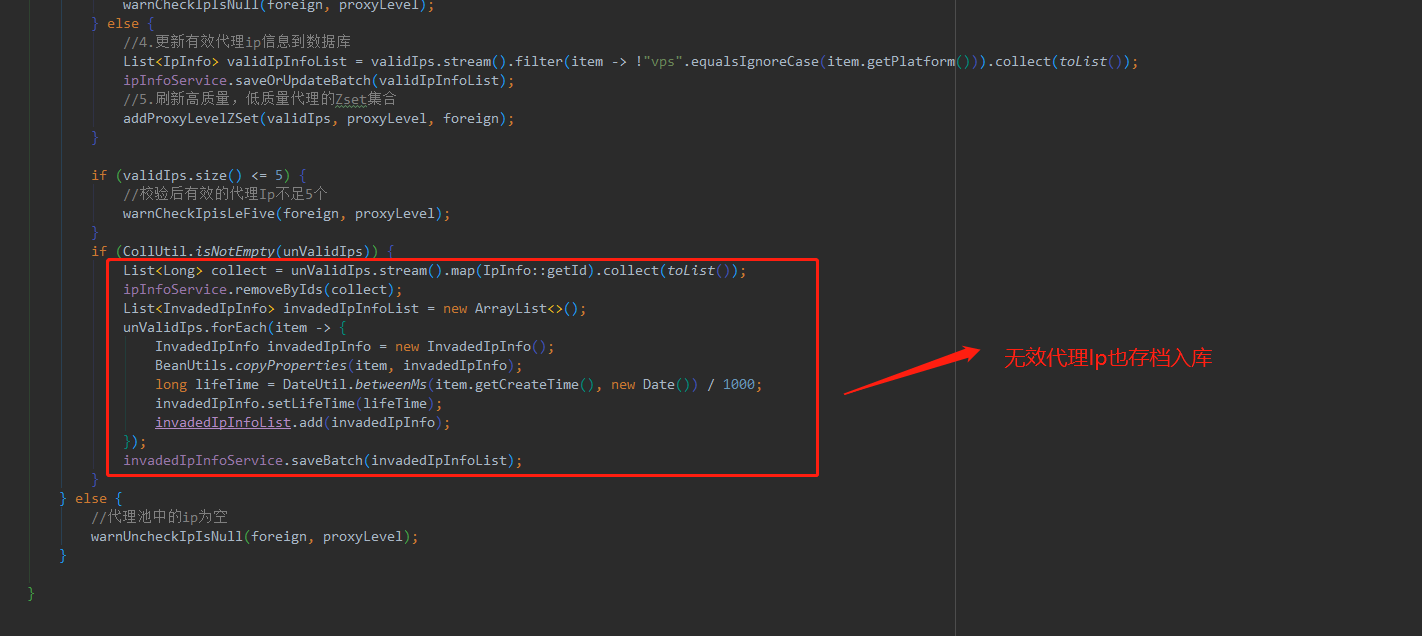
ScheduledTask.java 定时任务类



IpProxyProcessServiceImpl.java 代理Ip检测逻辑



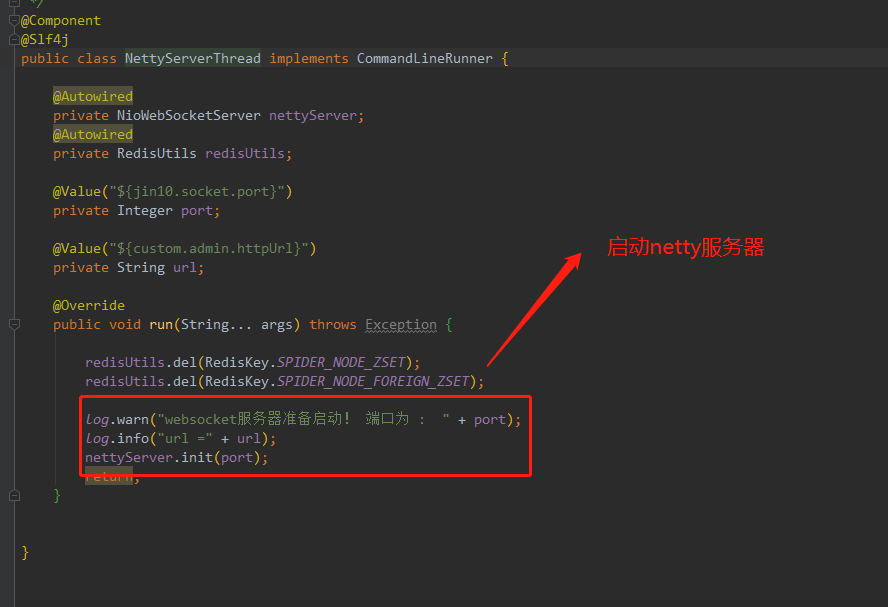




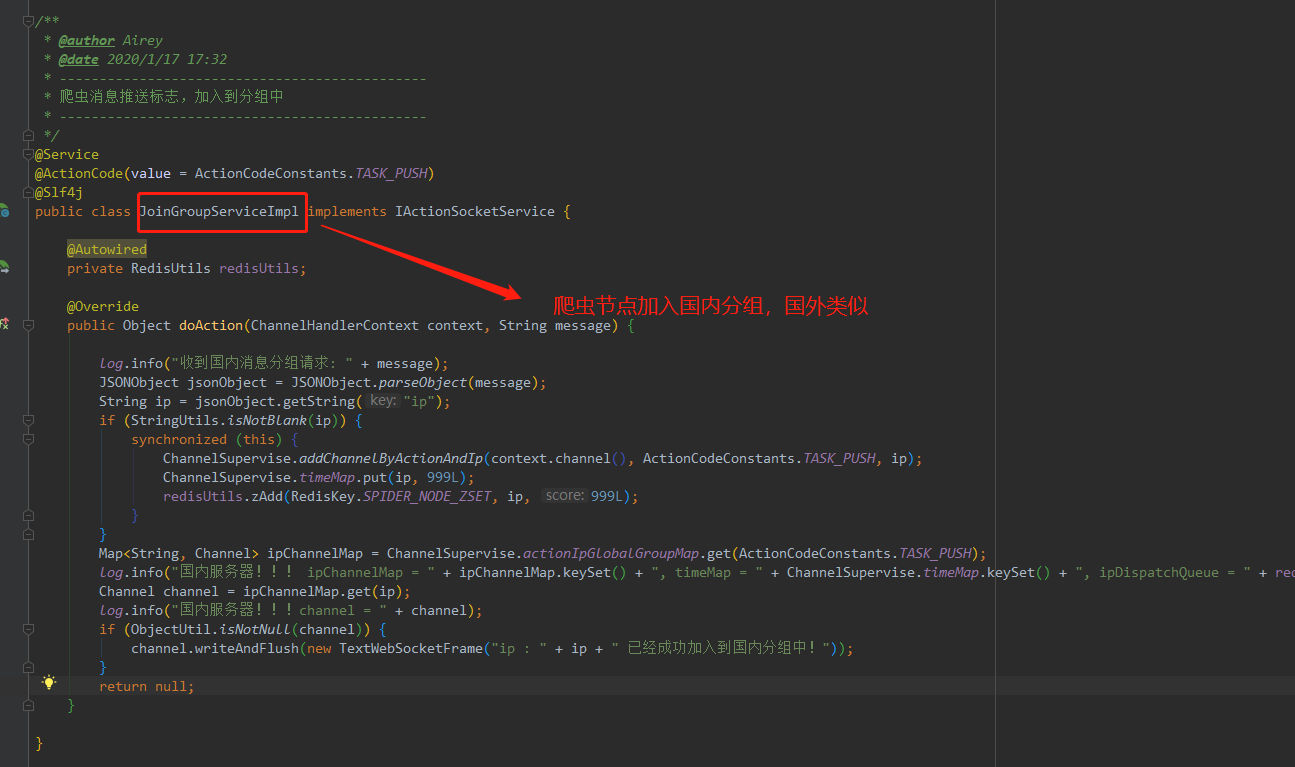
### 4.3 spider-socket项目(websocket服务器)

#### 4.3.1 爬虫任务推送核心逻辑

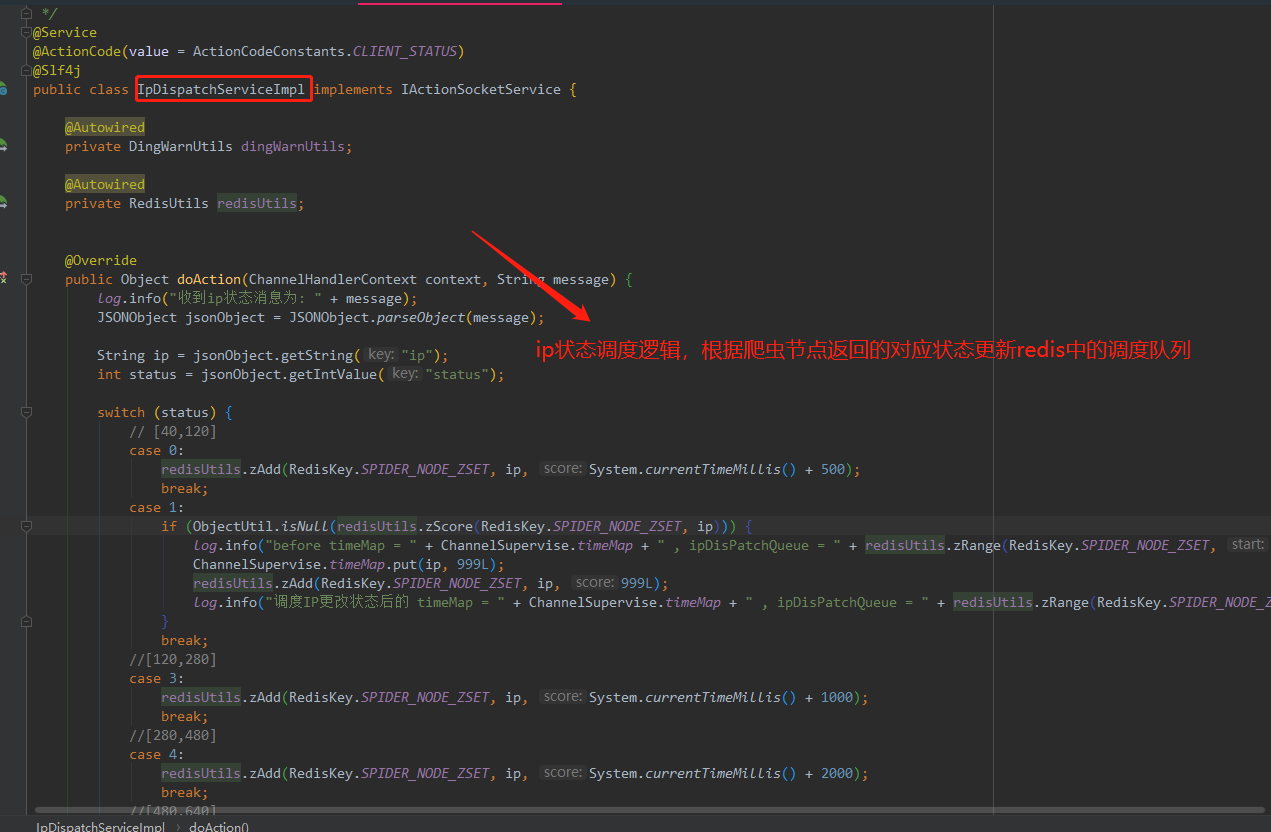
1.NettyServerThread.java



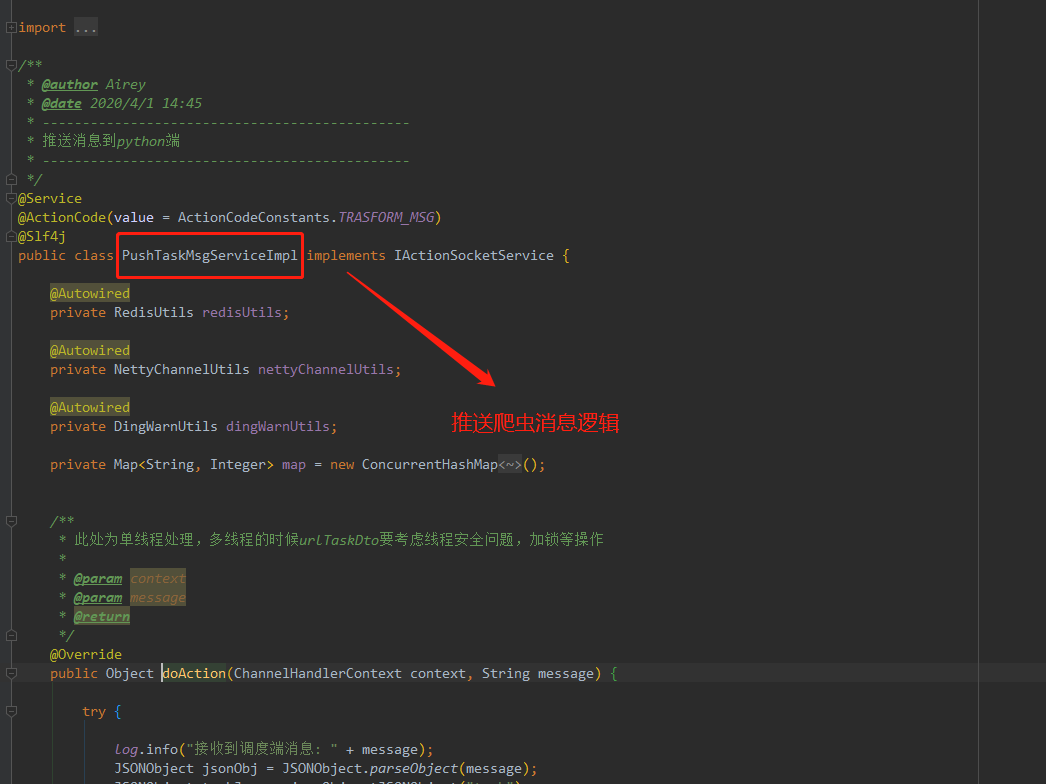
2.JoinGroupServiceImpl.java

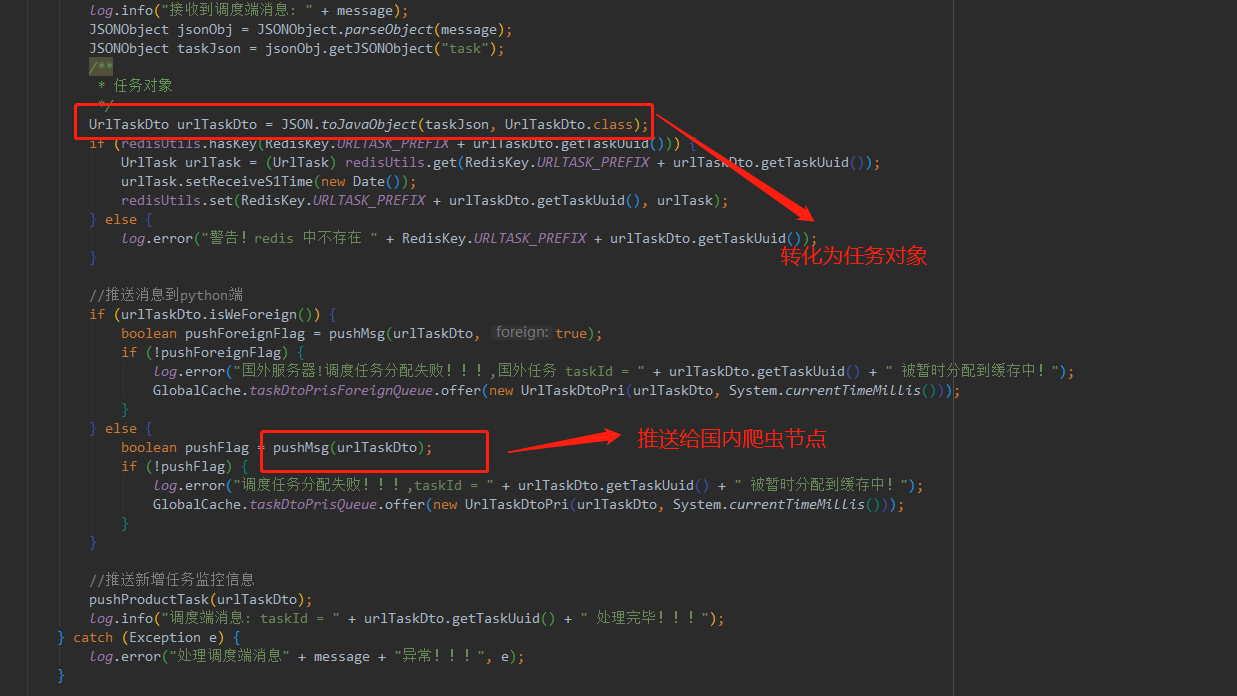


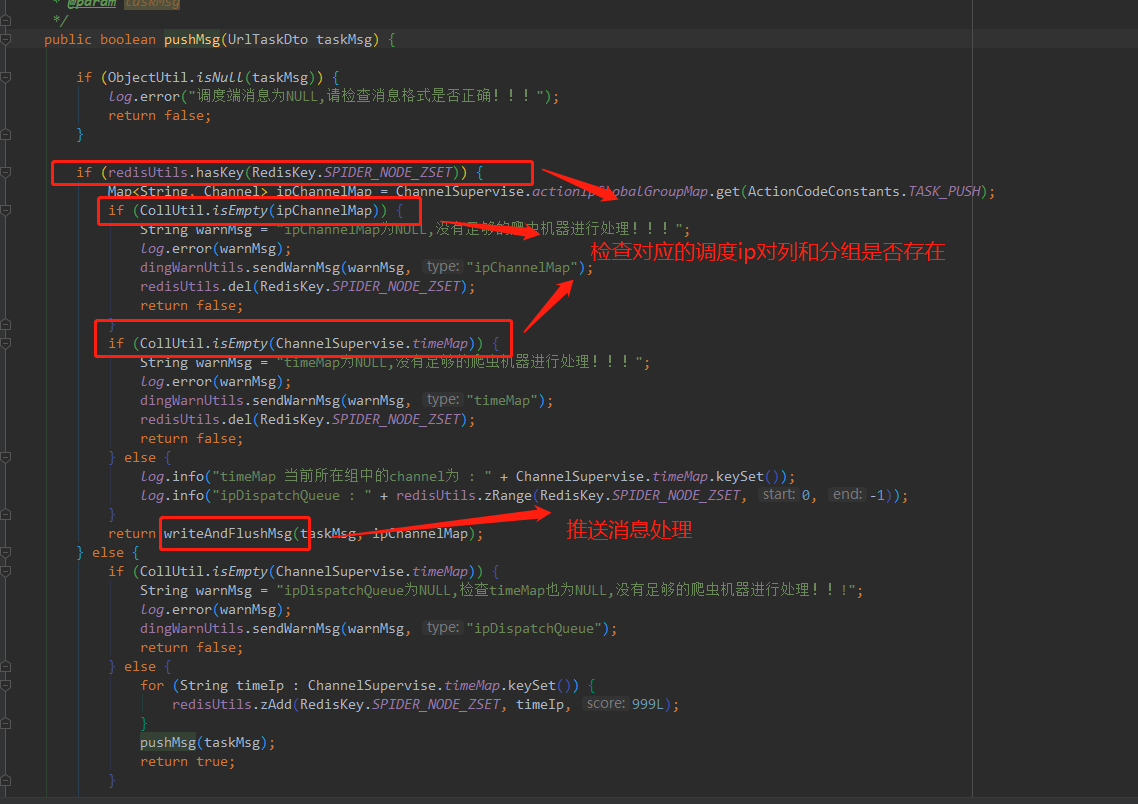
3.IpDispatchServiceImpl.java

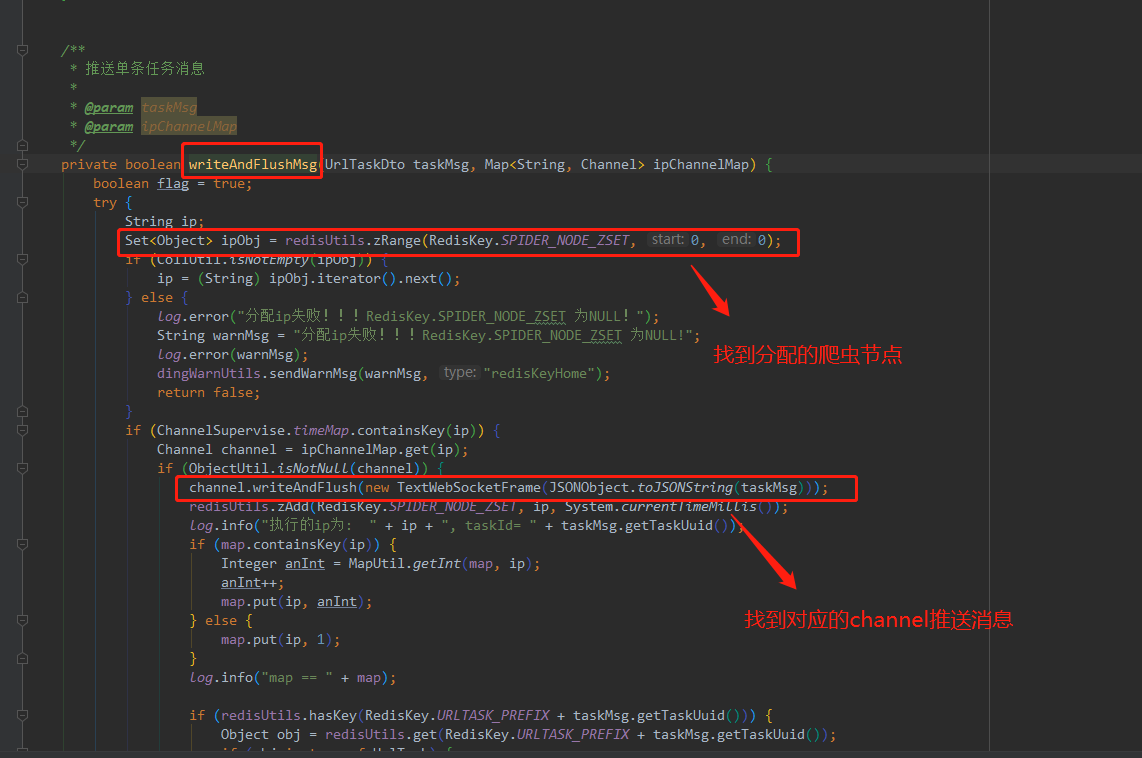


4.PushTaskMsgServiceImpl.java





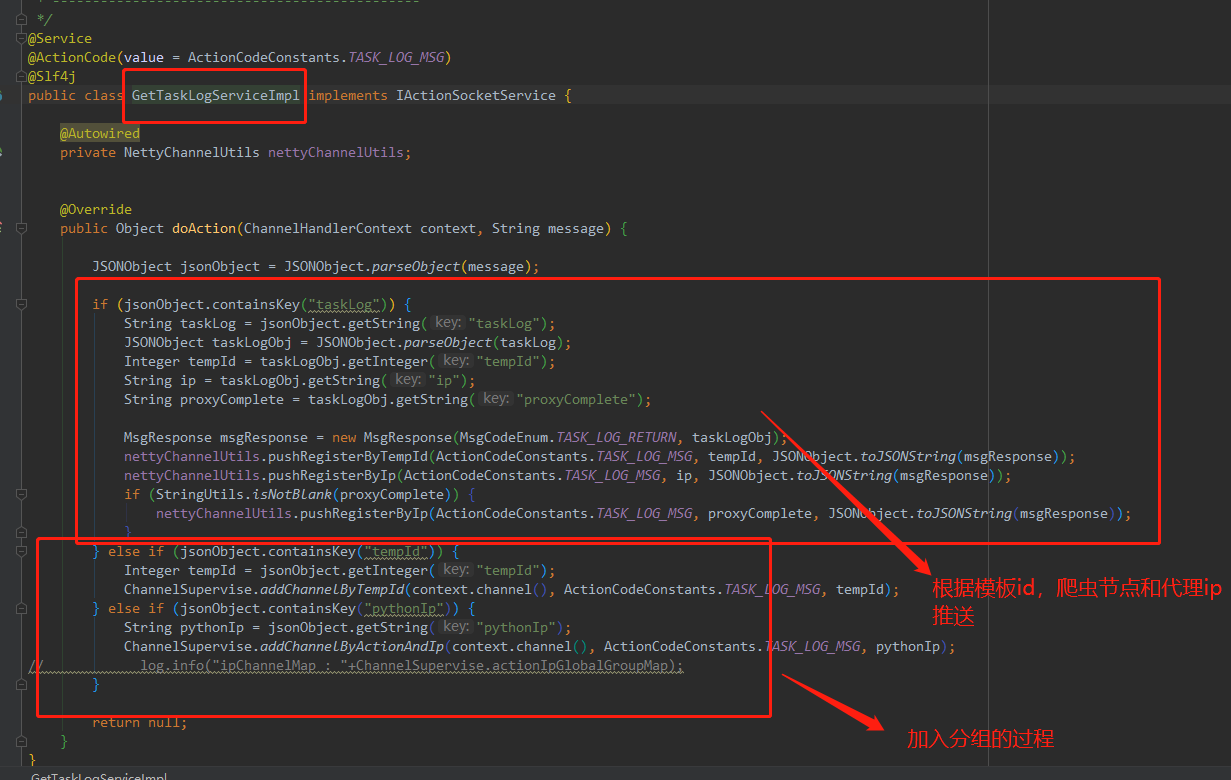






#### 4.3.2 推送爬虫实时日志

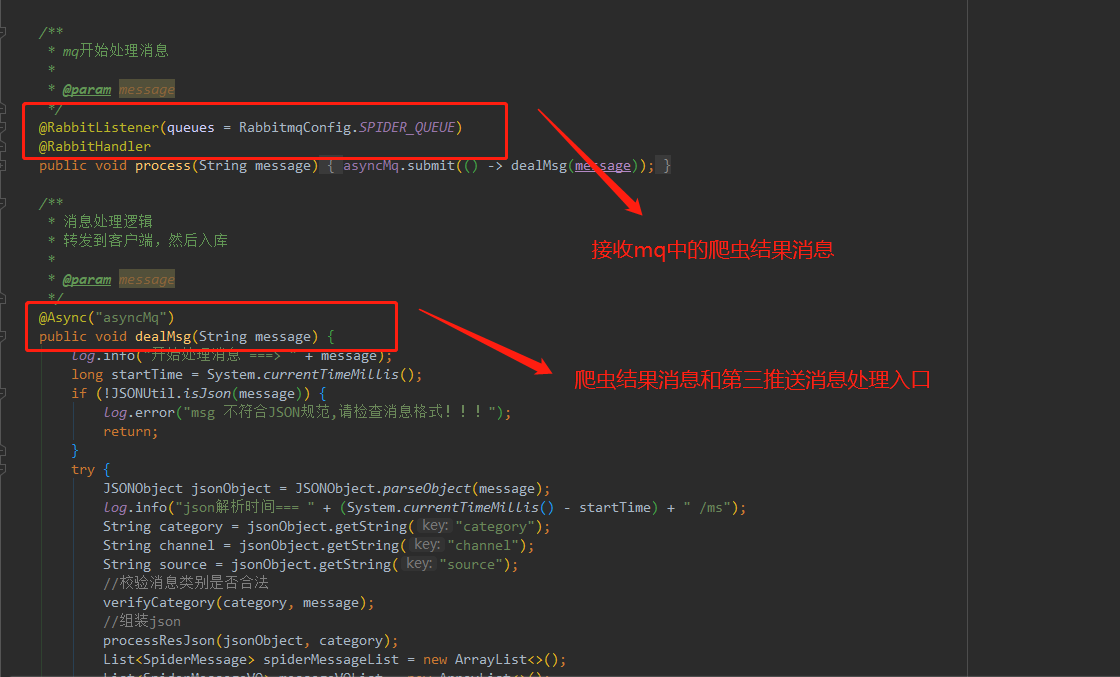
GetTaskLogServiceImpl.java

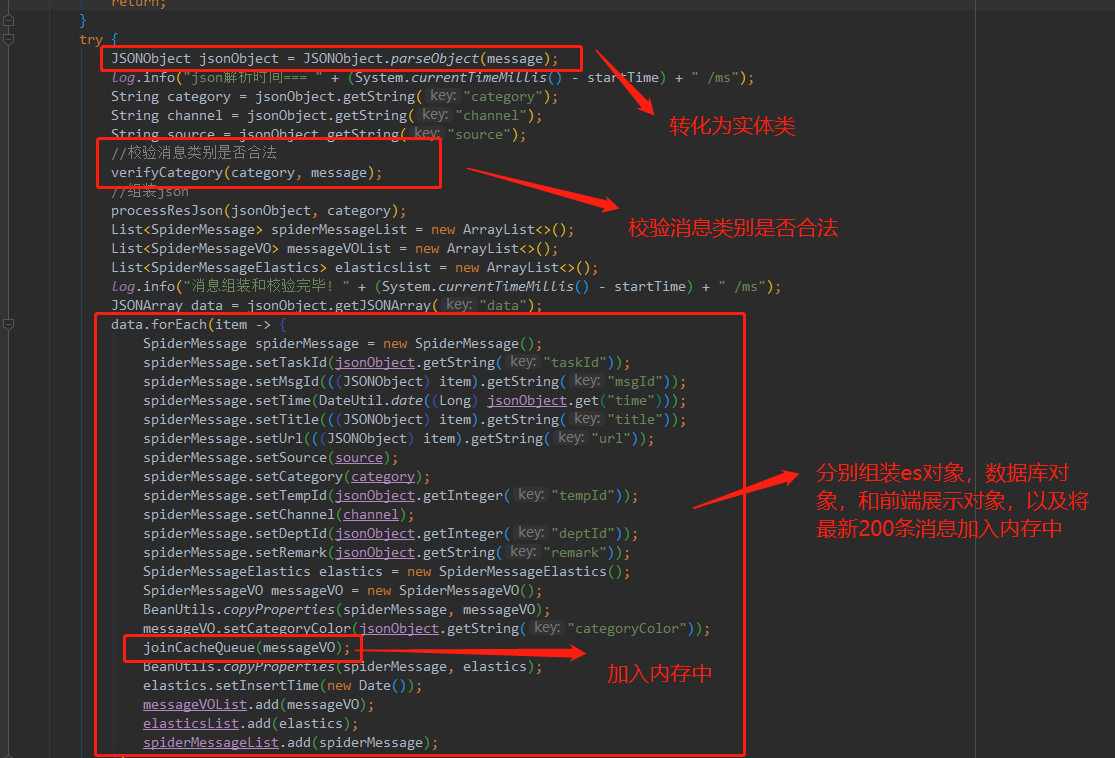


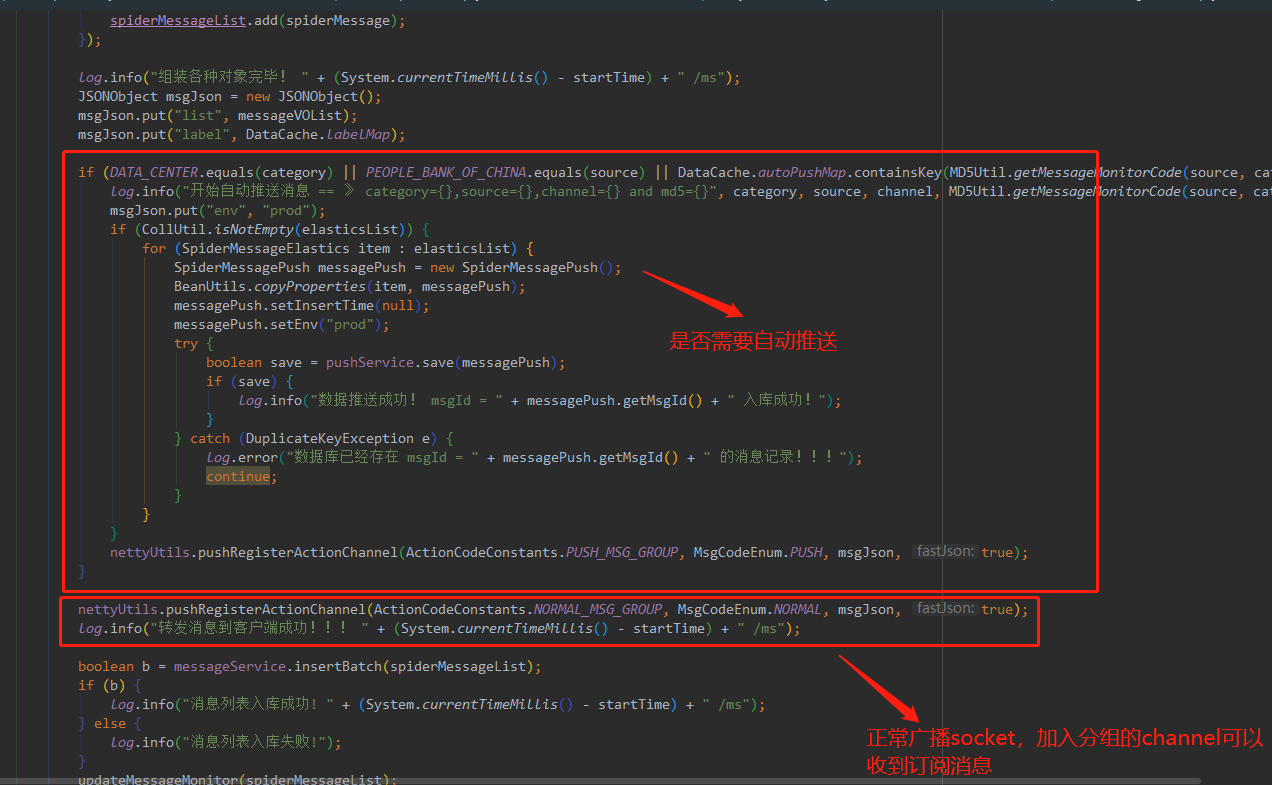
### 4.4 spider-server项目(爬虫终端及权限)

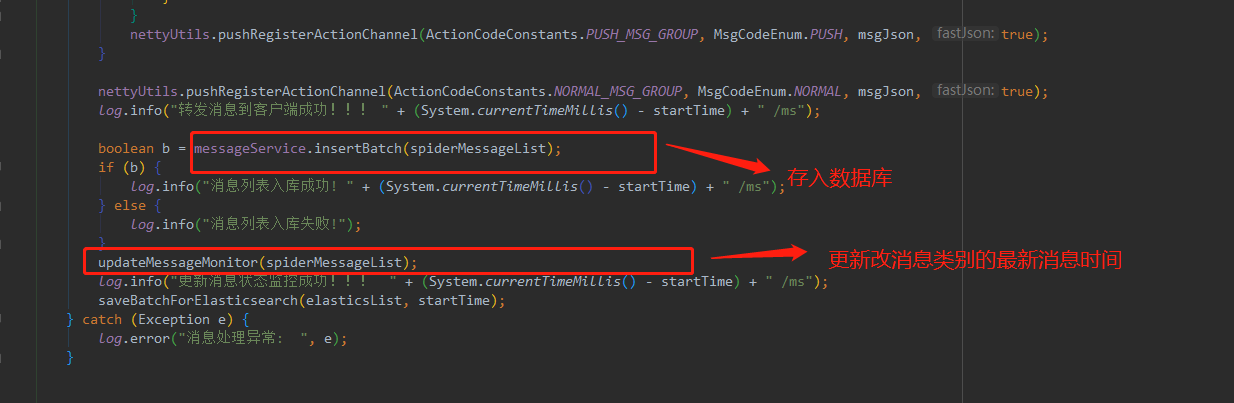
#### 4.4.1 爬虫消息处理核心逻辑

MqReceiver.java





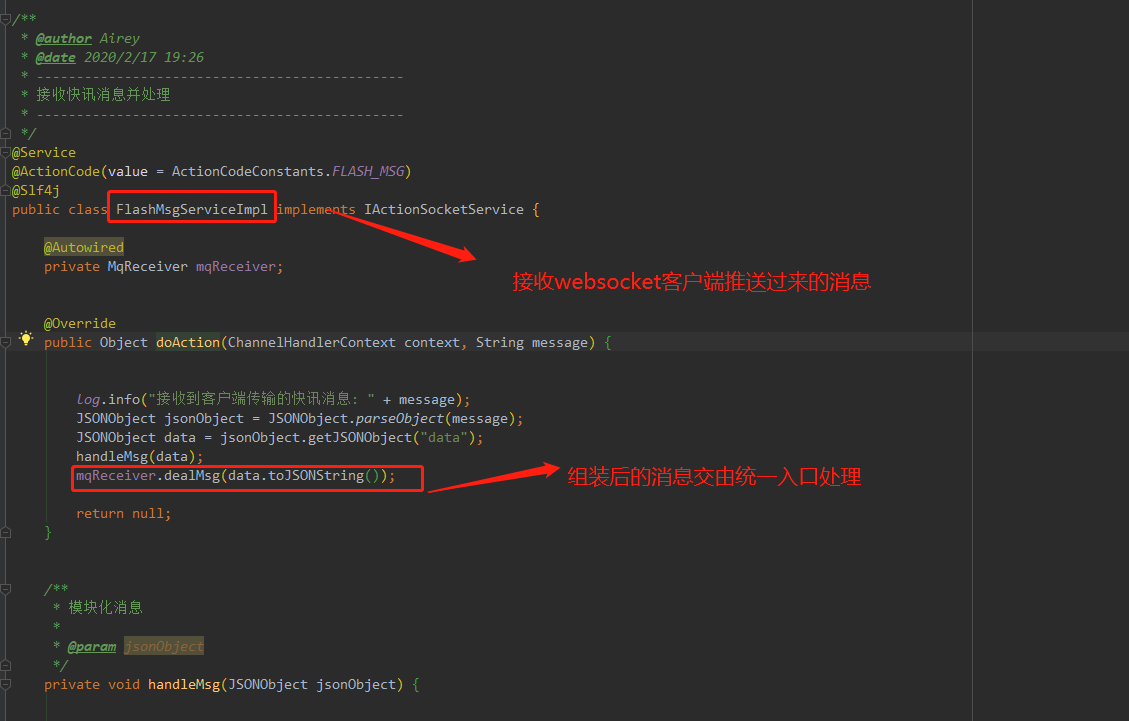




#### 4.4.2 第三方推送消息

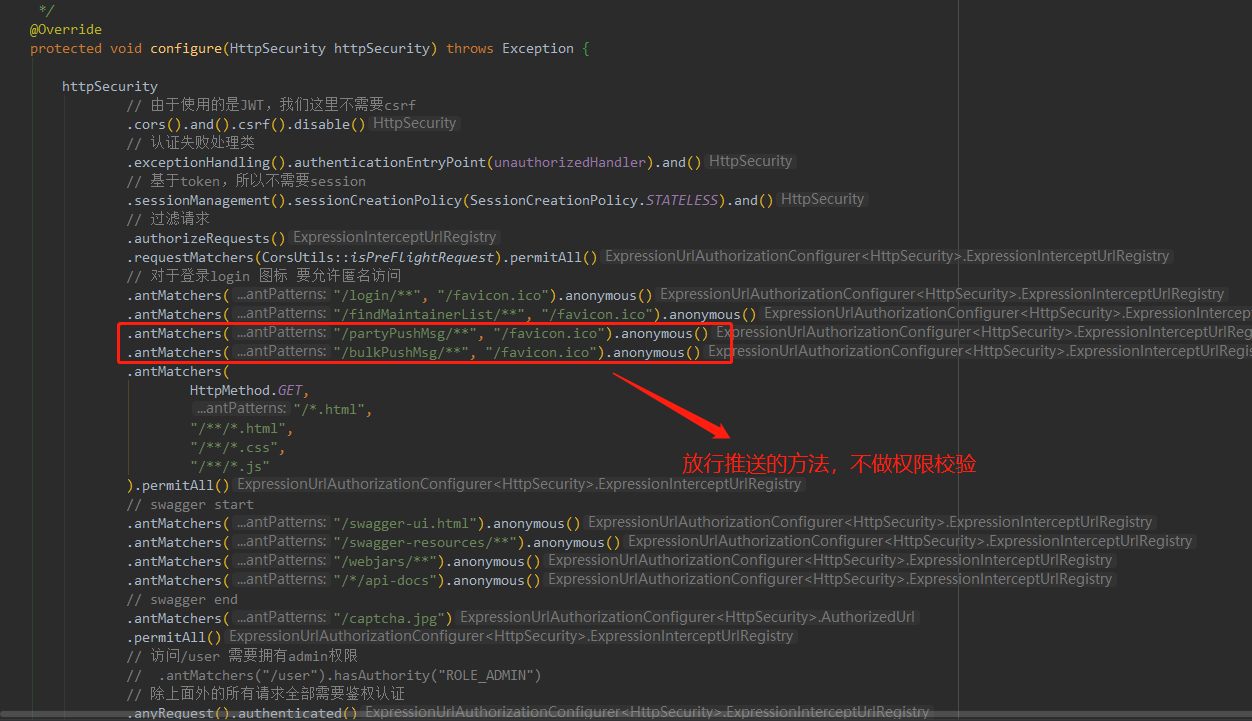
1. websocket推送消息

FlashMsgServiceImpl.java

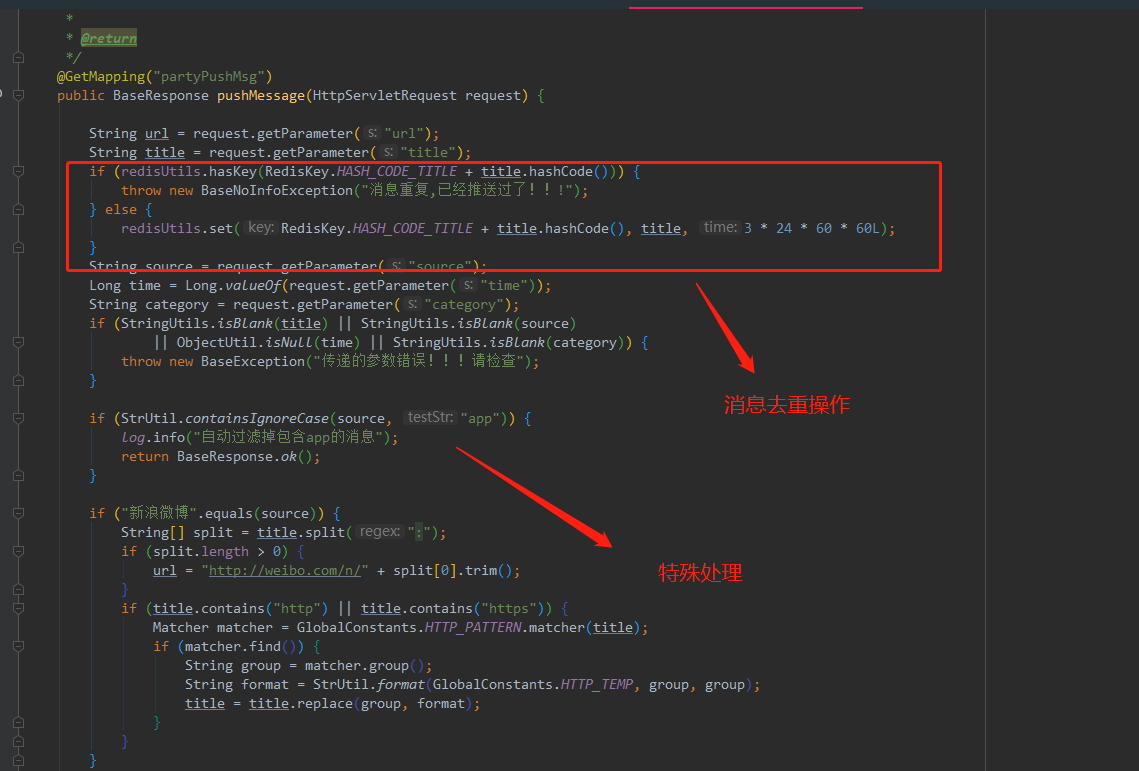


2.第三方快讯 http方式推送

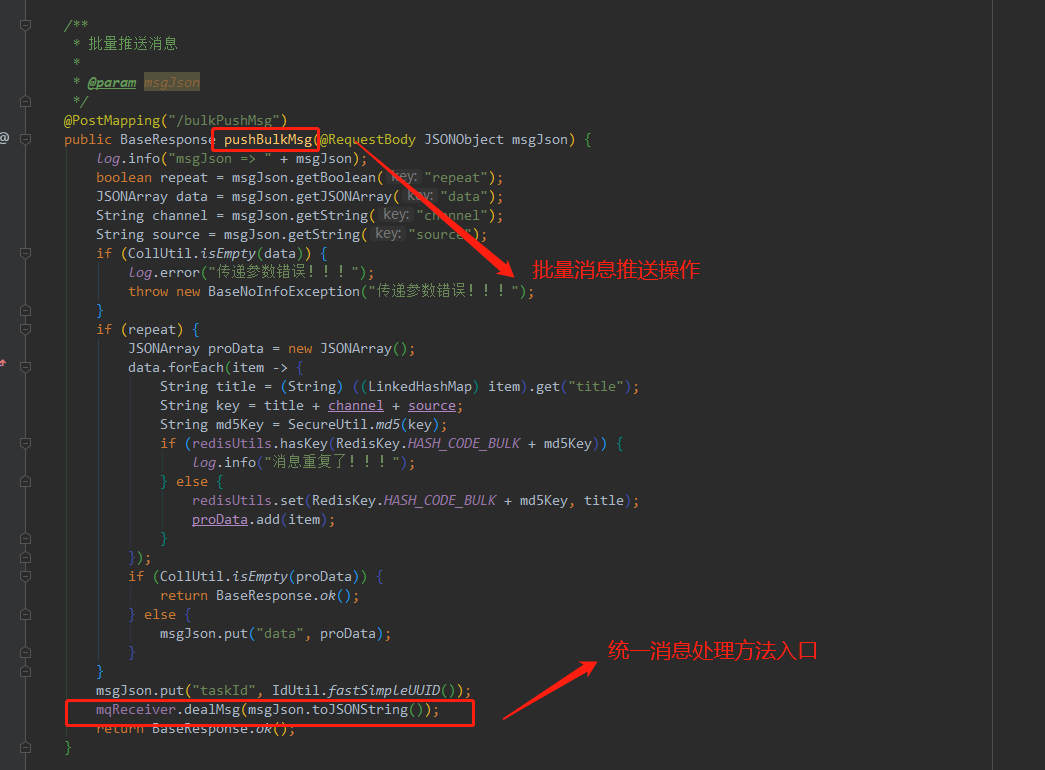
WebSecurityConfig.java



LoginController.java



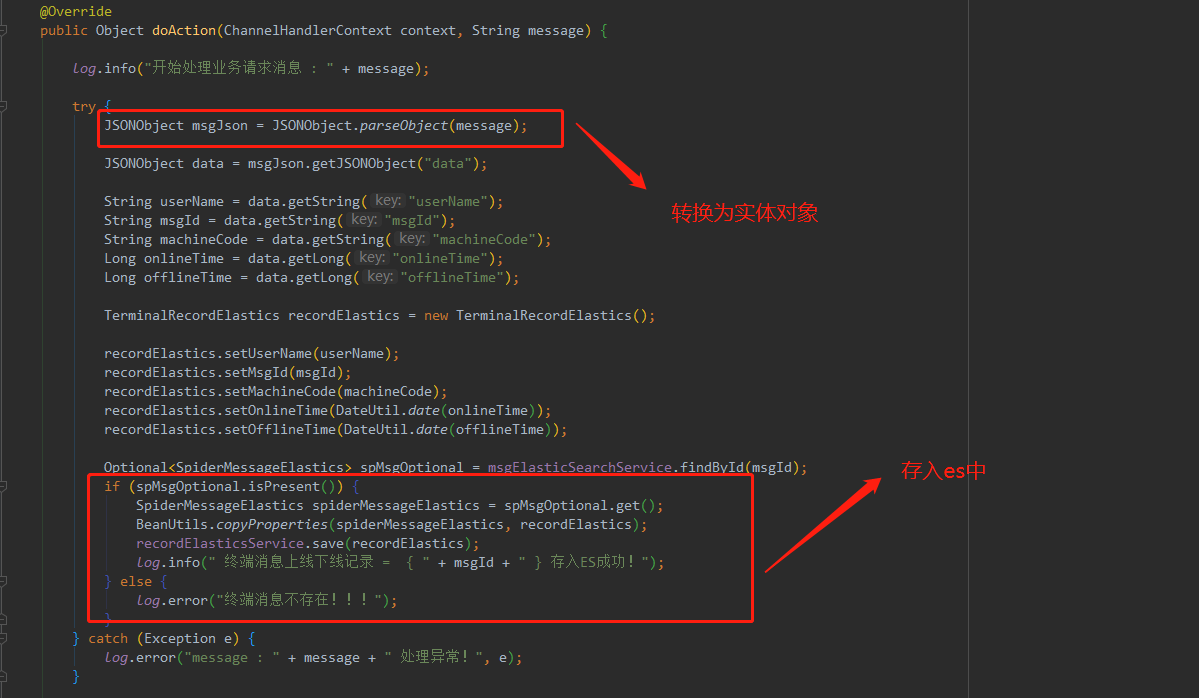




### 4.5 push-socket项目(上下线消息处理服务器)

核心功能为终端上下线消息记录操作

TerminalRecordServiceImpl.java



# 五．Websocket报文协议文档

### 5.1 Spider-server 部分

**websocket地址: ws://115.29.174.245:9700**

#### 正常转发爬虫端消息

发送json报文:  
**{  
    "action":702,  
    "ip":"127.0.0.1"  
}**加入分组，即可正常接收终端消息。

正常收发爬虫消息样例为  
{

    "msgDesc":"正常转发爬虫消息",  
    "msgType":702,  
    "result":{  
        "label":{  
            "金十":"#6A5ACD",  
            "杜然":"#8A2BE2",  
            "纯钢":"#8B008B"  
        },  
        "list":[  
            {  
                "category":"文章",  
                "categoryColor":"#FF00FF",  
                "channel":"通知公告",  
                "msgId":"4a8f6ea3d2e24b3db842162cda6cdf7a",  
                "source":"百度新闻",  
                "time":1575524825087,  
                "title":"新时代新民俗引领社会新风尚",  
                "url":"[https://mbd.baidu.com/newspage/data/landingsuper?context=%7B%22nid%22%3A%22news\_10035132804382781217%22%7D&n\_type=0&p\_from=1](https://mbd.baidu.com/newspage/data/landingsuper?context={"nid":"news_10035132804382781217"}&n_type=0&p_from=1)"  
            },  
            {  
                "category":"文章",  
                "categoryColor":"#FF00FF",  
                "channel":"通知公告",  
                "msgId":"f598f1fe0b4d4f6d8b1edab69014a0a1",  
                "source":"百度新闻",  
                "time":1575524825087,  
                "title":"戴姆勒·奔驰掌门康林松携EQS等多款新车亮相广州车展",  
                "url":"[https://mbd.baidu.com/newspage/data/landingsuper?context=%7B%22nid%22%3A%22news\_9700654362080774031%22%7D&n\_type=0&p\_from=1](https://mbd.baidu.com/newspage/data/landingsuper?context={"nid":"news_9700654362080774031"}&n_type=0&p_from=1)"  
            },  
            {  
                "category":"文章",  
                "categoryColor":"#FF00FF",  
                "channel":"通知公告",  
                "msgId":"89cdd7952688464eac119c04b53297b1",  
                "source":"百度新闻",  
                "time":1575524825087,  
                "title":"航天科技：圆满完成超音速大机动靶标验收兼鉴定飞行试验任务",  
                "url":"[https://mbd.baidu.com/newspage/data/landingsuper?context=%7B%22nid%22%3A%22news\_10142446032652951846%22%7D&n\_type=0&p\_from=1](https://mbd.baidu.com/newspage/data/landingsuper?context={"nid":"news_10142446032652951846"}&n_type=0&p_from=1)"  
            }  
        ]  
    }  
}

#### 客户端补全消息

爬虫消息补全发送请求json :  
**{  
    "action":301,  
    "ip":"127.0.0.1",  
    "data":{  
        "time":1574389771000  
    }  
}**  
其中action表示请求类型 ， ip为请求的客户端ip ，time为需要补全消息的时间戳类型节点，单位为ms。

返回示例：

{

 "result":{  
        "label":{  
            "金十":"#6A5ACD",  
            "杜然":"#8A2BE2",  
            "纯钢":"#8B008B"  
        },  
        "list":[  
            {  
                "categoryColor":"#FF00FF",  
                "channel":"通知公告",  
                "msgId":"314ef395be094abeb94679fac146a123",  
                "source":"百度新闻",  
                "time":1575524800000,  
                "title":"航天科技：圆满完成超音速大机动靶标验收兼鉴定飞行试验任务",  
                "category":"文章",  
                "url":"[https://mbd.baidu.com/newspage/data/landingsuper?context=%7B%22nid%22%3A%22news\_10142446032652951846%22%7D&n\_type=0&p\_from=1](https://mbd.baidu.com/newspage/data/landingsuper?context={"nid":"news_10142446032652951846"}&n_type=0&p_from=1)"  
            },  
            {  
                "categoryColor":"#FF00FF",  
                "channel":"推荐",  
                "msgId":"e13e6a30ec4f47d1bc43be046510658b",  
                "source":"每经手机网",  
                "time":1575522624000,  
                "title":"证监会复文全国人大代表建议：将积极探索金融交易风险分级",  
                "category":"文章",  
                "url":"<https://m.nbd.com.cn/articles/2019-12-05/1391275.html>"  
            },  
            {  
                "categoryColor":"#FF00FF",  
                "channel":"推荐",  
                "msgId":"d89cec6f11094abaa5c35675e2f651ac",  
                "source":"每经手机网",  
                "time":1575522553000,  
                "title":"违法违规采集个人信息！100款APP被下架整改",  
                "category":"文章",  
                "url":"<https://m.nbd.com.cn/articles/2019-12-05/1391273.html>"  
            }  
        ]  
    },  
    "msgDesc":"消息补全",  
    "msgType":302  
}  
  
其中msgType为消息类型，msgDesc为消息描述，result为消息结果集~~~~ label为高亮标签，list为数据列表~~~~

#### 客户端记录终端消息上下线记录

发送json报文：

**{  
    "action":380,  
    "ip":"127.0.0.1",  
    "data":{  
        "msgId":"884babf2c927471684be1fa44f94afcf",  
        "userName":"admin",  
        "onlineTime":1576045431000,  
        "offlineTime":1576045431000,  
        "machineCode":"34efeqr4353f3425243ty567465hg5637y36"  
    }  
}**

其中380代表报文类型   
msgid为消息唯一id  
userName为终端登陆用户  
machineCode为终端机器码  
onlineTime为消息上线时间  
offlineTime为消息下线时间  
客户端发送之后不会返回消息，服务器记录到es中供后续查询对比

#### 客户端推送终端消息

发送json报文：

**{**

**"action":370,  
    "ip":"127.0.0.1",  
    "msgIdList":[  
        "39ee8577a5d349cca91b65fff1c23ec3",  
        "2059db60ea294d5aa6a8ff2ff3022932",  
        "dc6b82384db249259588cd8edf73a5b6"  
    ]  
}**

如果是开发人员，则带上env字段

**{  
    "action":370,  
    "ip":"127.0.0.1",  
    "env":"dev",  
    "msgIdList":[  
        "39ee8577a5d349cca91b65fff1c23ec3",  
        "2059db60ea294d5aa6a8ff2ff3022932",  
        "dc6b82384db249259588cd8edf73a5b6"  
    ]  
}**

其中action表示报文类型，370定义为客户端筛选消息发送报文，ip为填充字段固定即可，msgIdList为唯一消息Id列表  
发送json报文：

**{  
    "action":371,  
    "ip":"127.0.0.1"  
}**加入成功分组后，即可正常接收返回的分拣消息列表中筛选后需要推送的消息。  
其他客户端收到广播的推送消息样式~~~~为：  
{  
    "msgDesc":"推送消息",  
    "msgType":371,  
    "result":{  
        "label":{  
            "圆满":"#bdff60",  
            "测试333":"rgba(130, 74, 74, 1)",  
            "ce33":"rgba(96, 41, 41, 1)",  
            "航天":"#2C1212"  
        },~~~~  
        "env":"prod",  
        "list":[  
            {  
                "category":"快讯",  
                "categoryColor":"#FF0000",  
                "channel":"首页快讯",  
                "msgId":"2059db60ea294d5aa6a8ff2ff3022932",  
                "source":"金十数据-美港电讯",  
                "time":1583375299000,  
                "title":"标普：维持京东（JD.O）全年收入增至少15%预期，疫情有利在线购物。",  
                "url":""  
            },  
            {  
                "category":"快讯",  
                "categoryColor":"#FF0000",  
                "channel":"全部快讯",  
                "msgId":"39ee8577a5d349cca91b65fff1c23ec3",  
                "source":"汇通网",  
                "time":1583375442000,  
                "title":"工行纸钯金RMB日内涨幅达到7.01%，报589.1500元/克",  
                "url":""  
            },  
            {  
                "category":"快讯",  
                "categoryColor":"#FF0000",  
                "channel":"市场行情",  
                "msgId":"dc6b82384db249259588cd8edf73a5b6",  
                "source":"金十数据",  
                "time":1583375298000,  
                "title":"【行情】银行板块拉升走强，张家港行涨停，苏州银行、苏农银行、江阴银行、紫金银行等跟涨。",  
                "url":""  
            }  
        ]  
    }  
}

#### 心跳消息

客户端向服务端发送心跳消息并响应

客户端发送json报文  
**{  
    "action":998,  
    "msg":"ping"  
}**

服务端向客户端回应心跳消息

**{  
    "msgType":999,  
    "msgDesc":"心跳消息",  
    "result":{  
        "msg":"pong"  
    }  
}**

#### 接收其他渠道快讯消息

客户端向服务端发送json请求报文（此为示例报文，实际根据自己的数据进行填充）:

**{  
    "action":395,**

**"data":{  
        "category":"快讯",  
        "channel":"市场行情",  
        "source":"金十数据",  
        "time":1581931232000,  
        "title":"受美国总统日假期影响，美股今日休市一日，CME旗下贵金属、美国原油与外汇合约交易提前于18日02:00收市，美国至2月14日当周API原油库存和EIA原油库存分别推迟至周四（20日）05:30和周五（21日）00:00公布。",  
        "url":"[https://mbd.baidu.com/newspage/data/landingsuper?context=%7B%22nid%22%3A%22news\_10134514950340906833%22%7D&n\_type=0&p\_from=1](https://mbd.baidu.com/newspage/data/landingsuper?context={"nid":"news_10134514950340906833"}&n_type=0&p_from=1)"  
    }  
}**

其中action表示报文标识字段，data为传输的消息数据实体，category为消息类型，channel为分类的频道，一般用网页栏目名称，source为网站名称，time为时间，title消息结果，url为当前消息的链接。

### Websocket部分

**Websocket地址： ws://115.29.174.245:12550**

#### python端接收推送socket服务器推送消息

加入到推送消息分组：

客户端向socket服务器发送json报文:  
**{  
    "action":721,  
    "ip":"192.168.88.139"  
}**即可加入待推送消息分组  
  
国外服务器发送  
**{  
    "action":711,  
    "ip":"192.168.88.132"  
}**

推送到每台服务器上的消息格式样例为：  
{  
    "temp":{  
"channel":"t5",  
        "source":"t5",  
        "tempId":813,  
        "category":"文章",  
        "encoding":"UTF-8",  
        "filterBy":1,  
        "tempRule":{  
            "method":"GET",  
            "rules":[  
                {  
                    "rule":"div a",  
                    "key":"url"  
                }  
            ],  
            "type":1  
        }  
    },  
    "taskUuid":"ebeddf1588814e6d9723b52706a5f8a6",  
    "url":"<http://10.218.61.91/list/45781>"  
}

#### python端向socket服务器上报当前状态

客户端向socket服务器发送json报文：

**{**

**"action":723,  
    "ip":"192.168.88.139",  
    "status":1  
}**上报当前服务器状态。

国外服务器发送json报文：  
**{  
  "action":713,  
    "ip":"192.168.88.132",  
    "status":1  
}**  
无返回报文。

#### 客户端接收监控服务器上报消息

客户端向socket服务器发送json报文：

**{**

**"action":1001,  
    "ip":"127.0.0.1"  
}**

立即接收到socket服务器返回的监控服务器列表信息：  
报文标识为1002，消息类型为“爬虫服务器消息列表”  
{  
    "msgType":1002,  
    "msgDesc":"爬虫服务器信息list",  
    "result":[  
        {  
            "system":{  
                "hostname":"hostname",  
                "os\_name":"Linux",  
                "linux\_distro":"linux\_distro",  
                "platform":"platform"  
            },  
            "load":{  
                "min5":2.89,  
                "min15":2.79  
            },  
            "mem":{  
                "total":3705348096,  
                "percent":89.1  
            },  
            "createTime":"2020-02-07 16:29:17",  
            "ip":{  
                "public\_address":"115.29.215.82"  
            },  
            "warned":true,  
            "cpu":{  
                "cpucore":2,  
                "total":74.6,  
                "system":6,  
                "user":23  
            },  
            "logId":"cdf400daa1174658bd60c6a3b426bcd4",  
            "fs":{  
                "/.size":42139451392,  
                "/data.size":105553780736,  
                "/.percent":31.7,  
                "/data.percent":1.6  
            },  
            "network":{  
                "eth1.interface\_name":"eth1",  
                "eth1.tx":98727,  
                "eth1.rx":108113,  
                "eth0.rx":3572,  
                "eth0.tx":1844,  
                "eth0.interface\_name":"eth0"  
            },  
            "uptime":{  
                "seconds":13065235  
            }  
        }  
    ]  
}

后续如有监控服务器上报状态：

则会接收到如下报文消息：  
报文标识为1003，消息类型为“单个爬虫服务器消息”  
{  
    "result":{  
        "system":{  
            "hostname":"hostname",  
            "os\_name":"Linux",  
            "linux\_distro":"linux\_distro",  
            "platform":"platform"  
        },  
        "load":{  
            "min5":2.89,  
            "min15":2.79  
        },  
        "mem":{  
            "total":3705348096,  
            "percent":89.1  
        },  
        "createTime":1581065411694,  
        "ip":{  
            "public\_address":"115.29.215.82"  
        },  
        "warned":false,  
        "cpu":{  
            "cpucore":2,  
            "total":74.6,  
            "system":6,  
            "user":23  
        },  
        "logId":"73fe7a29559c4fd69b64d31408e07e90",  
        "fs":{  
            "size":42139451392,  
            "dataPercent":1.6,  
            "dataSize":105553780736,  
            "percent":31.7  
        },  
        "network":{  
            "eth1Tx":98727,  
            "eth0Rx":1844,  
            "eth1Rx":108113,  
            "eth0Tx":3572,  
            "eth0InterfaceName":"eth0",  
            "eth1InterfaceName":"eth1"  
        },  
        "uptime":{  
            "seconds":13065235  
        }  
    },  
    "msgDesc":"单个爬虫服务器信息",  
    "msgType":1003  
}

#### 客户端接收实时生产任务监控信息

业务描述：  
此为socket服务器向客户端推送实时队列的监控信息，每次生产任务时新增的消息任务。  
  
客户端发送json报文：

**{  
    "action":1011,  
    "ip":"127.0.0.1"  
}**

每次生产任务向队列中新加入的消息。  
{  
    "msgType":1013,  
    "msgDesc":"新增任务",  
    "result":{  
        "url":"<http://10.218.61.91/list/851>",  
        "taskUuid":"a0718452d56b45a1b02698a168984f0b",  
        "weForeign":false,  
        "createTime":1583476902015,  
        "temp":{  
            "tempRule":{  
                "type":1,  
                "headers":null,  
                "method":"GET",  
                "params":null,  
                "rules":[  
                    {  
                        "rule":"div a",  
                        "key":"url"  
                    }  
                ]  
            },  
            "detailTempRule":null,  
            "encoding":"UTF-8",  
            "filterBy":1,  
            "tempId":2839,  
            "channel":"测试源851",  
            "category":"文章",  
            "deptId":null,  
            "remark":null,  
            "source":"t851",  
            "spiderTempId":null,  
            "encryptType":null,  
            "proxy":null  
        }  
    }  
}

#### 客户端接收爬虫节点实时返回爬虫日志消息

业务描述：  
客户端接收python端爬取的实时日志消息。

1.通过模板id接收

客户端向服务器发送json报文：  
**{  
    "action":871,  
    "ip":"127.0.0.1",  
    "tempId":809  
}**

2.通过爬虫ip接收

客户端向服务器发送json报文：  
**{  
    "action":871,  
    "pythonIp":"47.110.225.57"  
}**

3.通过代理ip接收

客户端向服务器发送json报文：

**{**

**"action":871,  
    "pythonIp":"171.80.161.191:20621"  
}**

socket服务器会向客户端推送对应模板id的爬虫日志信息：  
返回的爬虫日志消息示例格式如下：

{  
    "newResultInfo":[],  
    "ip":"47.110.225.57",  
    "snapshotInfo":[  
        {  
            "headers":"{"Expect":"","User-Agent":"Mozilla/5.0 (Windows; U; Windows NT 5.1; en) AppleWebKit/522.4.1+ (KHTML, like Gecko) Version/3.0.1 Safari/522.12.2","Pragma":""}",  
            "snapshotHash":"96866d599d8edf20e3f2813f00122cca",  
            "detailPage":false,  
            "url":"<http://10.218.61.91/list/8>",  
            "responseCode":200  
        }  
    ],  
    "source":"t9",  
    "foreign":false,  
    "puted":true,  
    "success":true,  
    "startTime":1581913916266,  
    "endTime":1581913918188,  
    "tempId":809,  
    "spiderId":"0",  
    "taskId":"8b1da040b71b4951b639f1885ffadf8c",  
    "status":2  
}

# 六．服务器部署

## 5.1 生产服务器配置

1.生产服务器：

**115.29.174.245（公网）| 172.16.30.177（内网） 4c8g 200g硬盘**

**公用密码：zSsjcDn9aoj0tontZmllqpVsKSAjqE**

2.数据库服务器：

**rm-bp1iml36y247329g2lo.mysql.rds.aliyuncs.com  
用户名：sprider  
密码：j6ixcvDn!M@s2ofOg9&VcX^^c7luxP**

## 5.2 docker环境搭建

1.安装需要的软件包  
yum install -y yum-utils device-mapper-persistent-data lvm2  
  
2.设置yum源  
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo  
或者 用阿里云的  
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo  
  
3.安装docker  
yum install -y docker-ce  
  
4.查询版本列表  
yum list docker-ce --showduplicates | sort -r  
  
5.指定版本安装(这里的例子是安装上面列表中的第二个)  
yum install -y docker-ce-17.09.0.ce  
  
6.启动docker  
systemctl start docker.service  
  
7.验证安装是否成功  
docker version  
  
8.设置容器日志最大值  
  
新建/etc/docker/daemon.json，若有就不用新建了。添加log-dirver和log-opts参数，样例如下：  
  
  
# vim /etc/docker/daemon.json  
  
{  
 "registry-mirrors": ["http://f613ce8f.m.daocloud.io"],  
 "log-driver":"json-file",  
 "log-opts": {"max-size":"250m", "max-file":"2"} # max-size=500m，意味着一个容器日志大小上限是500M，max-file=3，意味着一个容器有三个日志，分别是id+.json、id+1.json、id+2.json。  
}  
  
// 重启docker守护进程  
# systemctl daemon-reload  
  
# systemctl restart docker  
  
9.设置docker开机自启动  
  
systemctl enable docker  
  
10. 查看docker容器状态  
  
docker stats

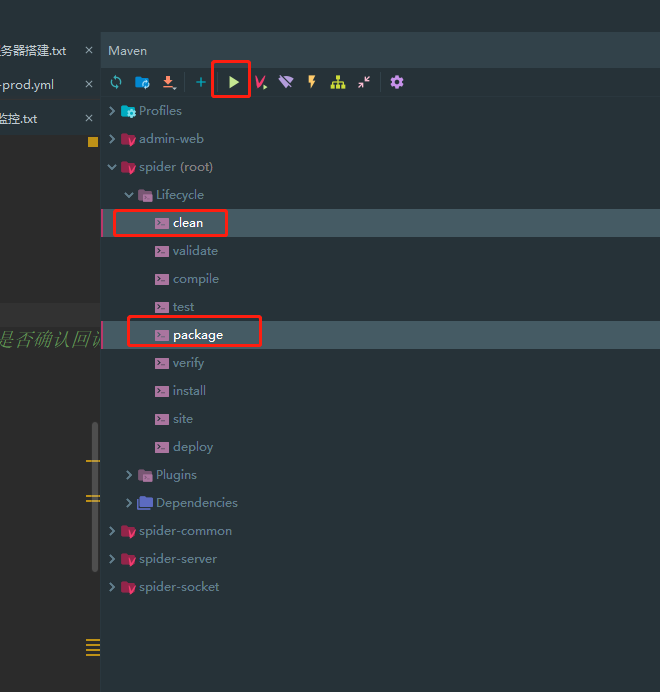
## 5.3 核心容器启动命令

1.spider-server  
docker run --restart always -d -p 7200:7200 -p 9700:9700 --network jin10net --network-alias net-spider-terminal -v /mnt/spider/terminal/log:/log --name spider-terminal spider-server:7.3  
  
  
2.admin  
docker run --restart always -d -p 7100:7100 --network jin10net --network-alias net-spider-scheduler -v /mnt/spider/scheduler/log:/log --name spider-scheduler admin:7.2  
  
  
3.websocket  
docker run --restart always -d -p 12550:12550 --network jin10net --network-alias net-spider-socket -v /mnt/spider/websocket/log:/log --name socket-server spider-socket:4.8  
  
  
4.pushsocket  
docker run --restart always -d -p 13340:13340 -p 14570:14570 --network jin10net --network-alias net-push -v /mnt/spider/push/log:/log --name push push:1.5

## 5.4 部署方法

### 5.4.1 Docker 单个项目的打包和部署方法

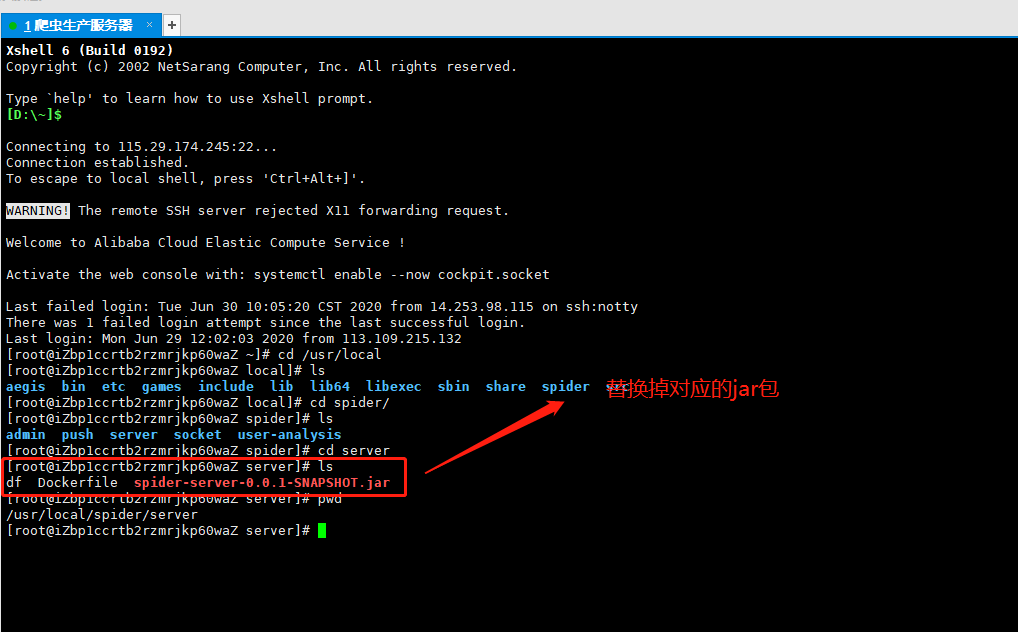
1.执行打包命令：

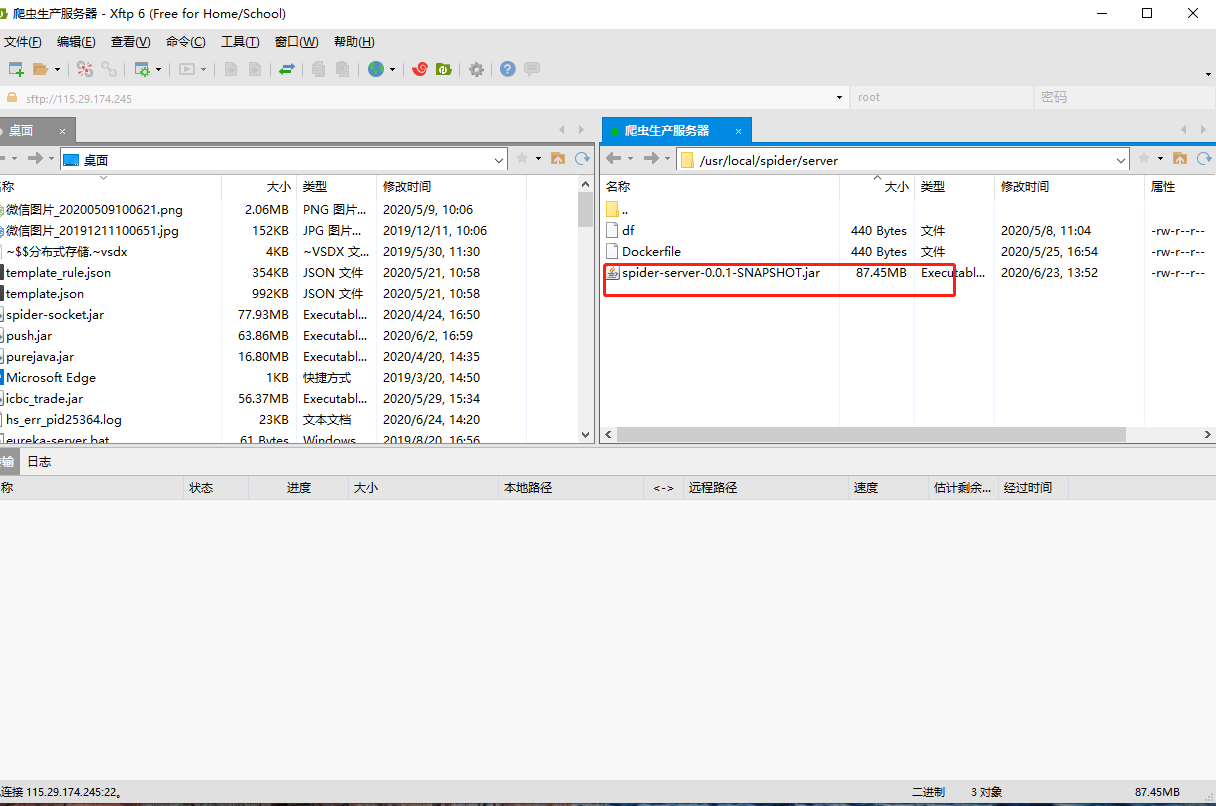


1. 将对应的jar包上传到服务器

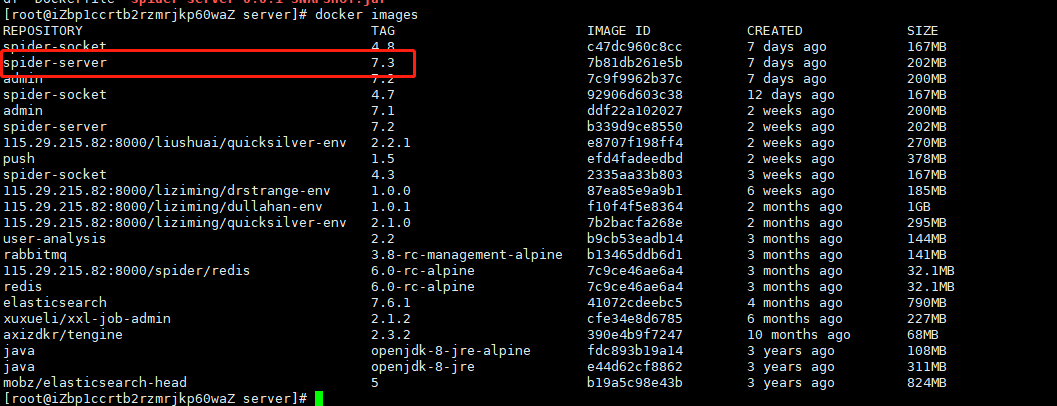


3.上传jar包





4.查看docker镜像对应的版本



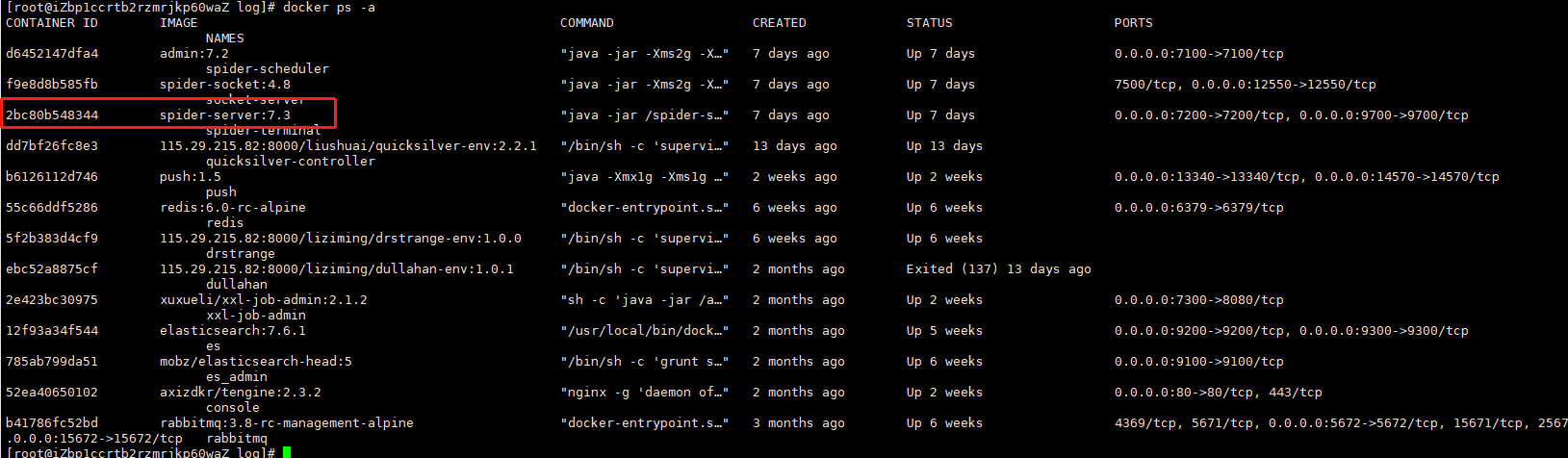
1. 执行docker打包命令



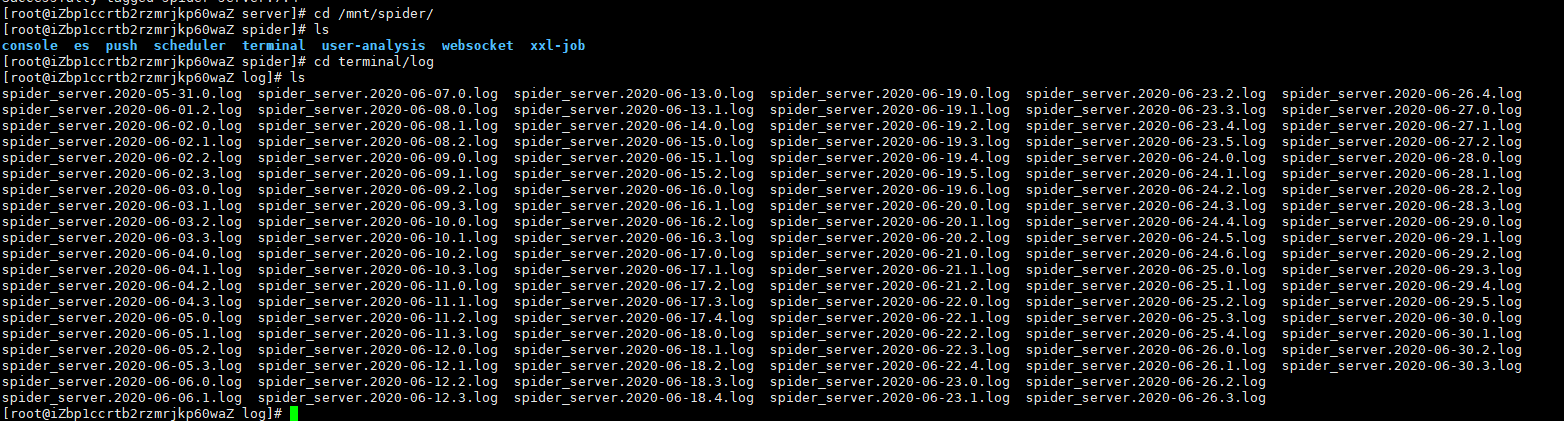
1. 用docker启动命令执行镜像

docker run --restart always -d -p 7200:7200 -p 9700:9700 --network jin10net --network-alias net-spider-terminal -v /mnt/spider/terminal/log:/log --name spider-terminal spider-server:7.4

1. 使用docker ps -a命令查看启动状态



1. 进入到/mnt/spider/目录，查看对应项目的生成环境日志



### 5.4.2 整体项目部署方法

**如果只需单独部署spider-server，则直接将打好的jar包上传至服务器，然后用docker部署即可**

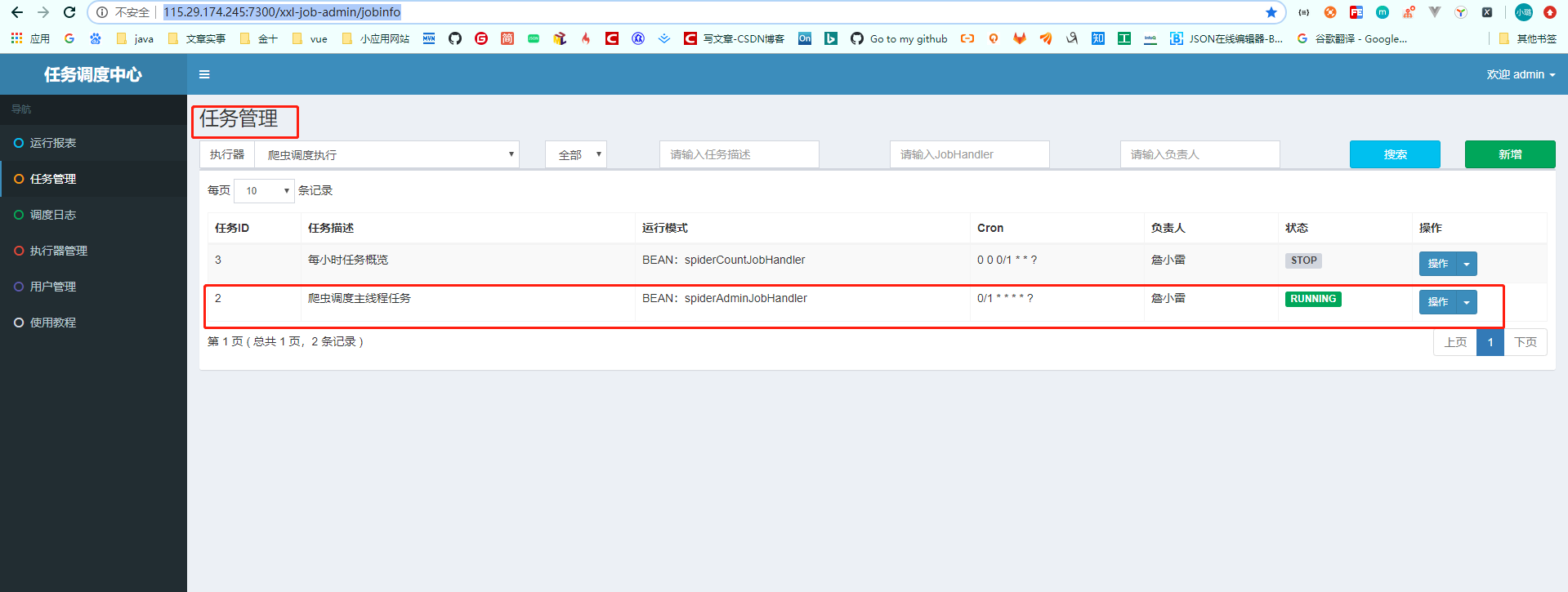
如果需要部署admin或者websocket项目，则需按以下步骤执行。

1.先打开任务调度中心

<http://115.29.174.245:7300/xxl-job-admin/jobinfo>

用户名 ： admin

密码 ：123456



点击操作->停止，

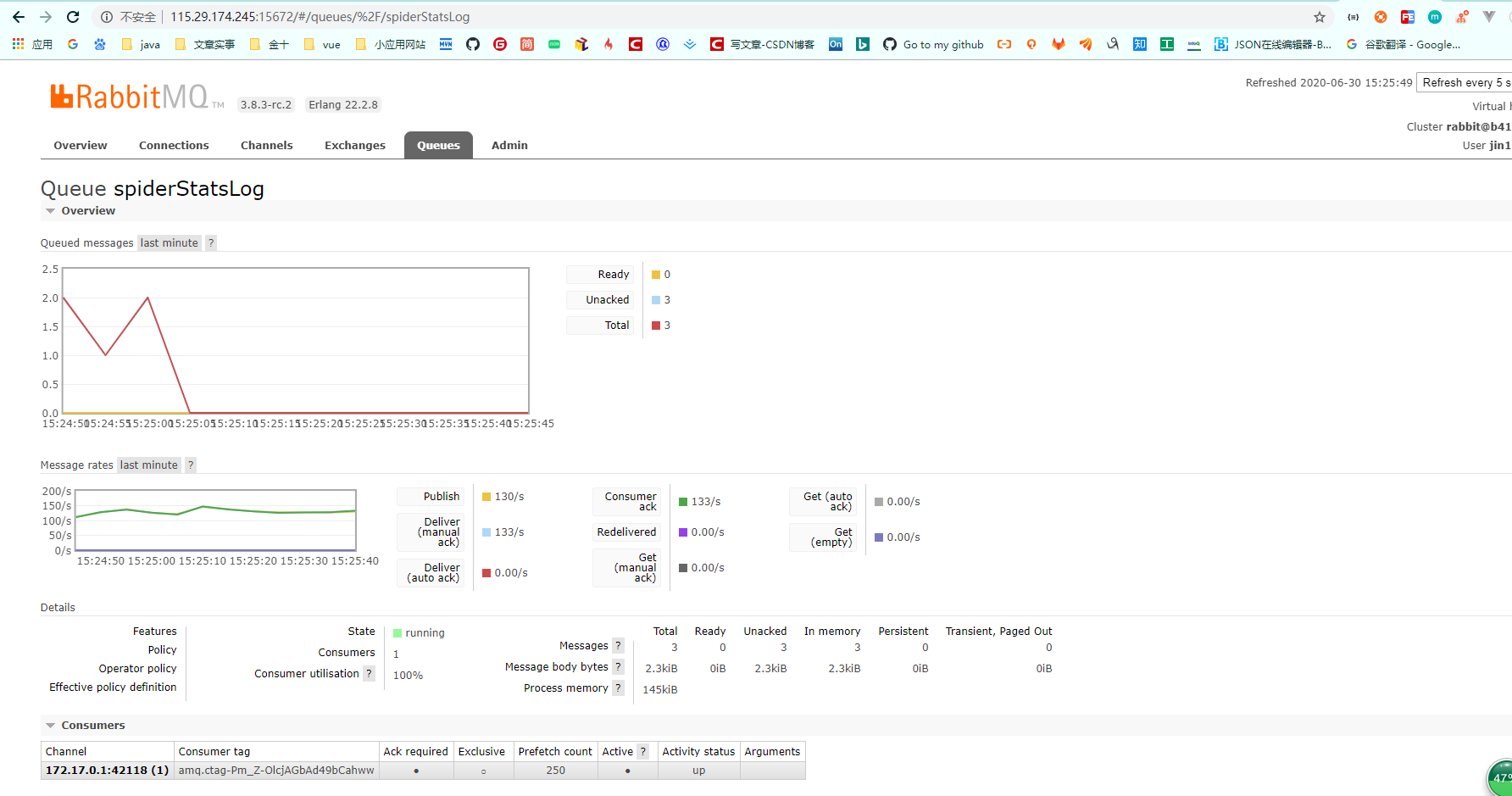
即可停止调度任务。

1. 打开rabbitmq监控页面

地址：[http://115.29.174.245:15672/#/](http://115.29.174.245:15672/" \l "/)

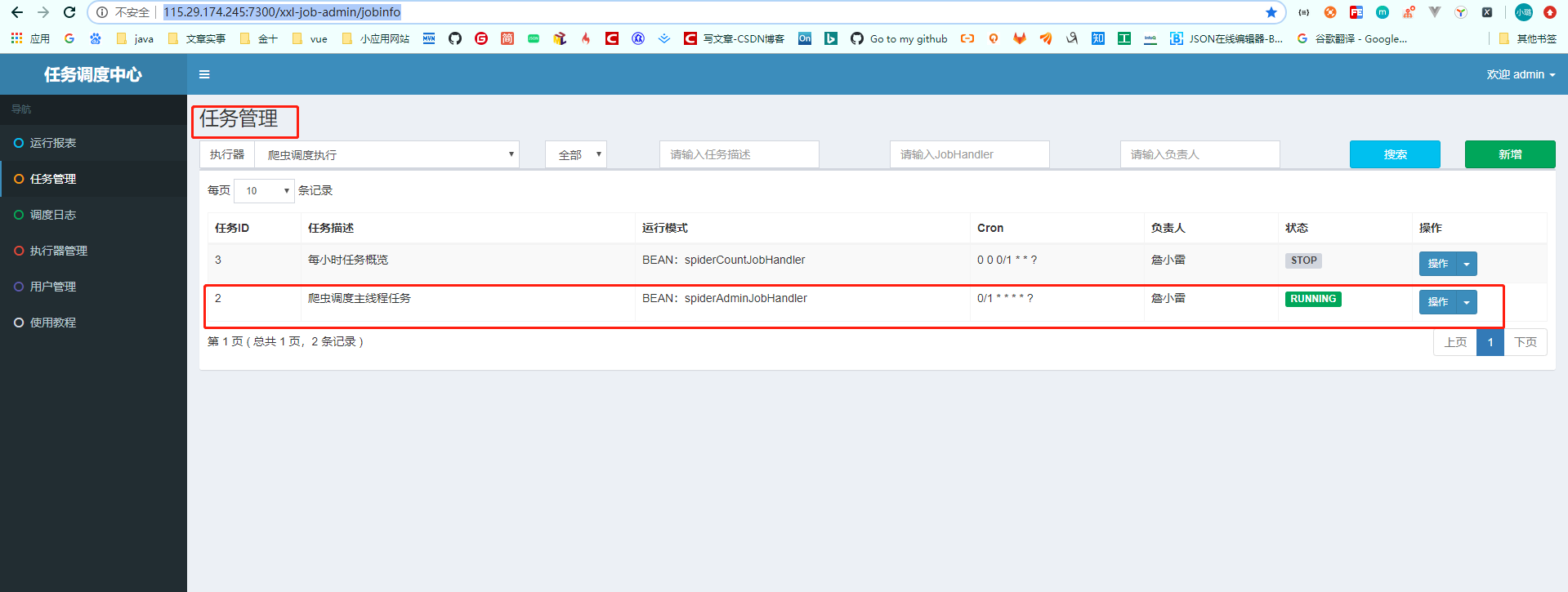
用户名： jin10

密码： j6ixcv2airw



观测到消息速率为0/s时才可开始部署项目。

1. 先部署websocket项目，再部署admin项目。
2. 部署完成后，重新回到任务调度中心页面。



点击操作->执行

5.观察后台日志，及rabbitmq的消息速率，正常后即表示部署成功。