一、想看问题总结

1.Git相关的问题及对应的解决办法

|  |  |
| --- | --- |
| 问题描述 | 解决方法 |
| arc diff失败  本地分支与最新分支相差版本太多 | 1. idea左下角的version control -> Log   -> 选中自己的分支 ->cherry pick  Picture1   1. git pull   Git rebase |
| Arc lint没问题，arc diff出现checkstyle错误，且错误的地方不是自己修改的地方 | 1. 查看本地分支是否是从master下拉取下来的 2. 如果不是，重新从master分支拉取。然后cherry-pick自己的提交 |
| 将本地分支与远端分支进行绑定 | git branch --set-upstream-to=origin/v-4.9.60 v-4.9.60 |
| Git ignore | 使用git status查看状态时，如果修改的地方在gitignore里面，那么可以暂时忽略改文件的修改，直接在本地提交。具体的问题在push之前执行mvn clean package后检查 |
| Arc diff 提交后显示的不同有误 | 可能本地分支不是从master上拉取下来的。  保留本地修改后将本地分支删除，从master拉取。再合并保留的修改 |
|  |  |
|  |  |

Arc land提交代码时，显示有问题

1. arc land --onto master
2. 根据提示 arc land --onto master --revision id

这里的id就是arc list提交后生成的url中的一部分id:

如url = <https://phabricator.d.xiaomi.net/D218007>

那么id 就是218007

1. 包升级问题及对应的解决办法

注意：

1. 从master下载源程序后，在本地新建自己的分支并在自己的分支上进行代码的修改
2. Git push 到远端自己的分支上
3. 如果需要先根某个约定的分支（如stage）进行合并时,先切换到约定的分支上再合并自己的分支

Git checkout stage

Git merge 自己的分支

1. 工具类的升级需要通过引用该工具类的web工程来进行测试
2. 工具类升级如果需要修改版本号，应先使用 mvn deploy提交到本地仓库中（注意版本号命名加上test，以防别人错误引用），然后在web工程中引用自己的版本进行测试。经测试无误后再进行 code review(arc diff | arc land)

|  |  |
| --- | --- |
| 问题描述 | 解决方法 |
|  |  |
| 源程序本地执行失败 | （1）先在xbox环境下将别人部署过的程序重新编译部署一遍，排查是否是源程序的问题。 |
| 源程序本地执行失败，但历史版本运行正确 | 将程序修改后到直接到xbox环境测试修改。 |
| 项目部署到线上 | 需要写脚本部署，一般在build.sh文件里面 |
| Eureka(<http://c3.eureka.browser.srv/>)  登陆问题 | 账号和密码都是：  browser |

1. **checkStyle问题：**

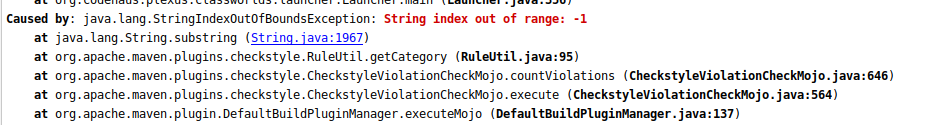
Push 和 arc diff 之前要使用

**Mvn clean package(首选)**

arc lint（增量检查）或者是

（mvn clean compile -U全量检查）

(1)maven-compiler-plugin 不显示错误的位置，但是报错：



是因为：checkstyle 错误太多，导致编译器溢出。

完整的checkstyle错误在 /target/checkstyle-result.xml文件中

将该文件拖到浏览器中，搜索error，即可找到所有的checkstyle错误

注意：checkstyle 不会检测test文件下的格式错误，为了保险，应该在本地使用

Mvn clean package来测试项目的编译部署是否有问题。

1. 别人的问题及解决办法

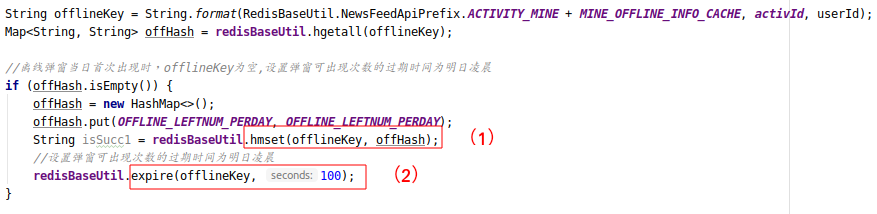
|  |  |
| --- | --- |
| 问题 | 解决办法 |
| 二次分享token失效 | 建立一个缓存，定义一个timeout |
| 想看激励任务的时间优化 | 新手任务与日常任务异步优化 |
| Swagger | 只在测试环境使用，线上不使用 |

1. **idea 问题**
2. idea工程目录显示不正常时，可以使用

Mvn -u idea:idea

就能展示idea风格的工程了

1. 开发过程中遇到的问题。
2. 给某一个key设置过期时间的前提是先将这个key保存到redis中。然后再设置过期时间。如果先设置过期时间，再保存，过期时间不生效。



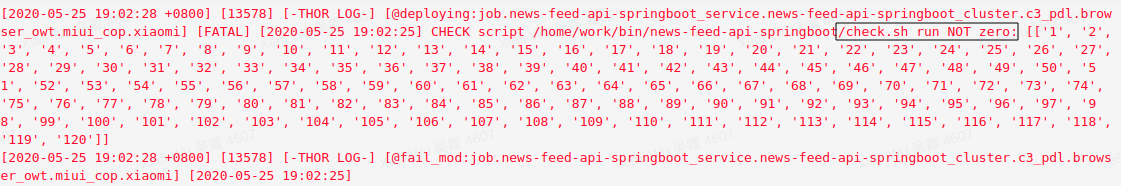
|  |
| --- |
| String offlineKey = String.*format*(RedisBaseUtil.NewsFeedApiPrefix.***ACTIVITY\_MINE*** + ***MINE\_OFFLINE\_INFO\_CACHE***, ***activId***, userId); Map<String, String> offHash = **redisBaseUtil**.hgetall(offlineKey);  *//离线弹窗当日首次出现时，offlineKey为空,设置弹窗可出现次数的过期时间为明日凌晨* **if** (offHash.isEmpty()) {  offHash = **new** HashMap<>();  offHash.put(***OFFLINE\_LEFTNUM\_PERDAY***, ***OFFLINE\_LEFTNUM\_PERDAY***);  String isSucc1 = **redisBaseUtil**.hmset(offlineKey, offHash);  *//设置弹窗可出现次数的过期时间为明日凌晨* **redisBaseUtil**.expire(offlineKey, 100); } |

1. 创建某个key-value时，需要给value赋值。



如果不给offHash put一个值而直接hmset，会报异常

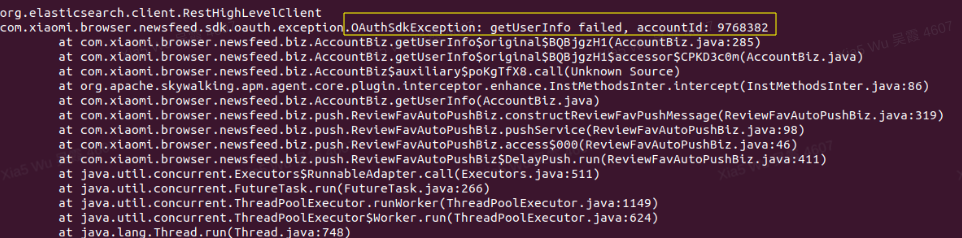
（3）问题：在正式环境部署服务的时候，部署报错



原因：服务部署上去之后，2分钟内服务没有启动，报错 check.sh run not zero

登陆部署服务器，查看该服务的端口

Netstat -anp | grep 8134





1. 在配置健康检查中台评论参数的过程中，按照程序的指示配置需要的参数，却仍然返回

{‘status’:500,’msg’:’review error’}。

原因：有些参数在程序中无法看到，但是发送请求却是必须的。

解决：找相关的负责人（中台评论组的同学）讨论，或者wiki查找相关接口说明，解决问题。

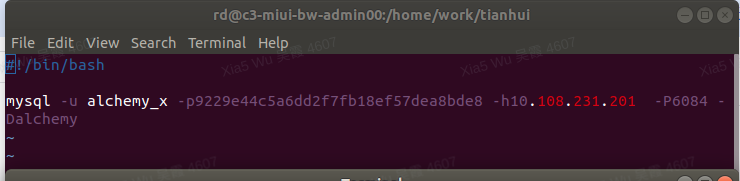
1. 想看线上文案修改问题：

线上机器数据库的修改：

登陆跳板机：ssh [wuxia5@relay.xiaomi.com](mailto:wuxia5@relay.xiaomi.com)

进入c3-miui-bw-admin00机器:

/home/work/tianhui 目录，有个alchemyDb 脚本里，是数据库账号信息

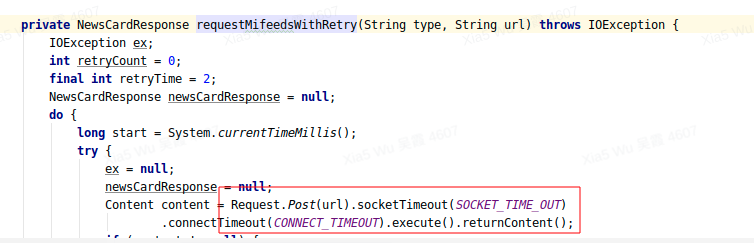


在**线上机器c3-miui-bw-admin00上执行mysql数据库连接命令**即可。

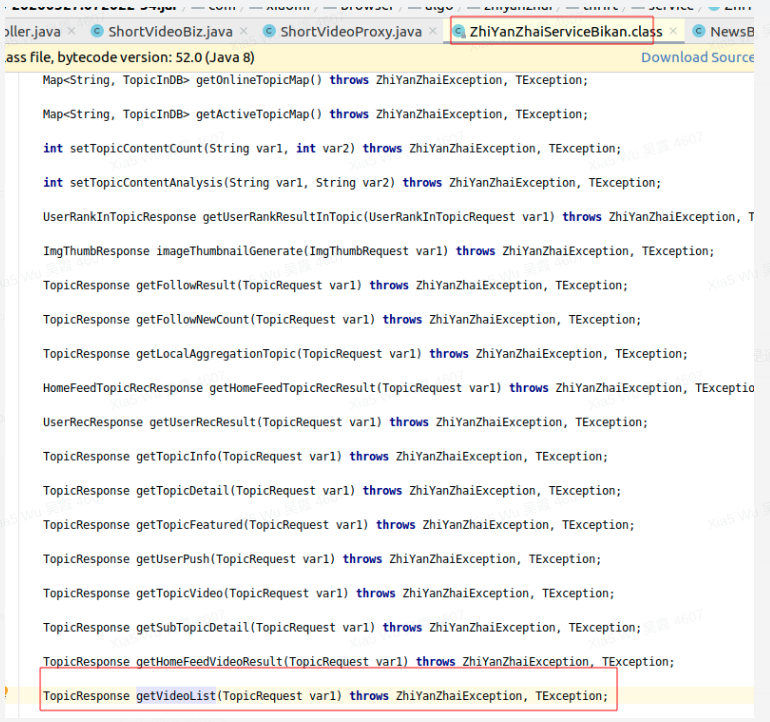
（测试环境可以直接在本机的terminal上执行）

mysql -u alchemy\_x -p9229e44c5a6dd2f7fb18ef57dea8bde8 -h10.108.231.201 -P6084 -Dalchemy

1. 推荐内容频道的获取



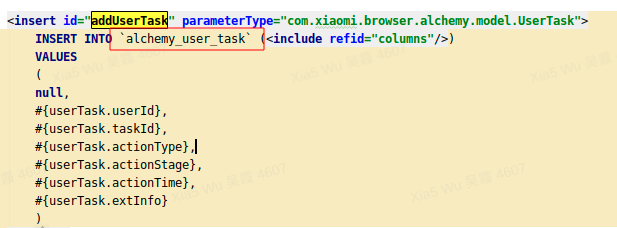
Request接口请求的是大数据的推荐数据



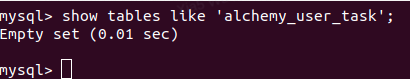
ZhiYanZhaiServiceBikan接口请求的是算法推荐的数据

1. 数据库问题：

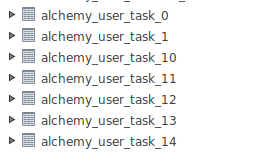
代码逻辑里面查询的是：



查询数据库：



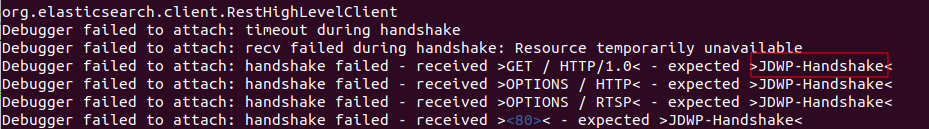
但是数据库中有很多



这是因为：

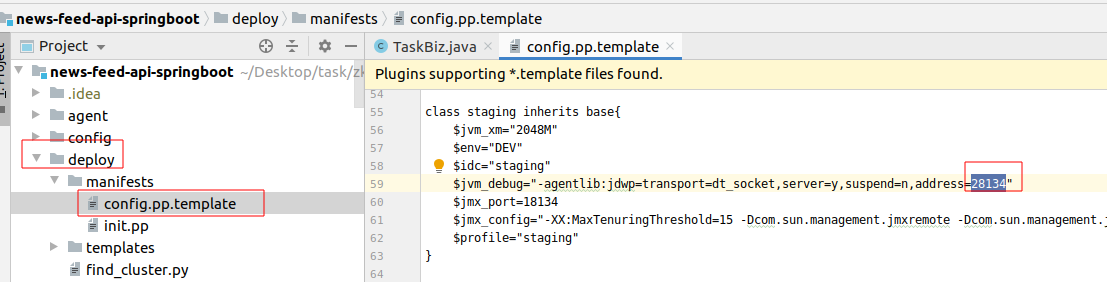
Mybatis自动分表了，将一个alchemy\_user\_task根据某种规则（想看是根据userid）分表。这个转换是Mybatis自动做的。业务逻辑直接查询alchemy\_user\_task即可。

1. 本地使用mvn clean package命令build 成功，但是在部署时失败了。登陆部署的机器，查看项目的run.log日志。



这是因为调试程序端口被占用。

进入目录查看调试端口为28134：



登陆机器查看具体的端口使用情况：



杀掉进程15647，重新部署即可。

1. 本地测试打点程序，显示打点成功，但是在数据工厂用sql命令查询不到结果。

可能的原因：

（1）修改数据库hive表结构模式，但是该模式没有时时更新到数据工厂。这样导致的结果是：

打点数据已经写入了hdfs，但是在数据工厂查不出来

1. 数据的写入有时延
2. 请求有问题

打点的逻辑是：

客户端发起打点请求（进程1） ------->LCS 代理（进程2） ------>打点落库

LCS代理是SRE 安装到机器上的。

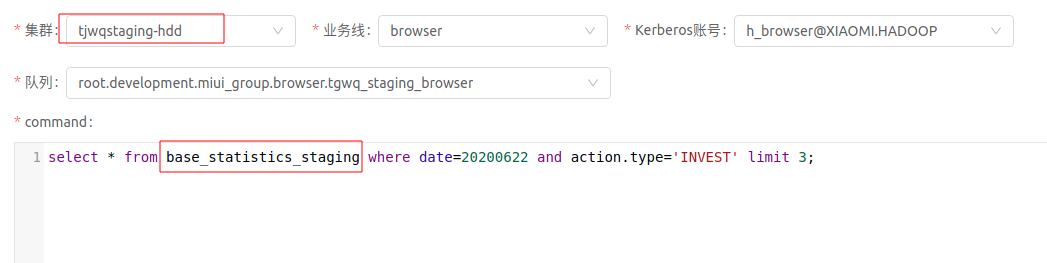
如果服务请求的是本机，由于本机没有安装LCS代理，所以打点实际上没请求。

正确的方式：

1. 用postman直接访问测试服务机实现打点
2. 测试环境的打点记录查询的集群不一样。如base\_statistics。

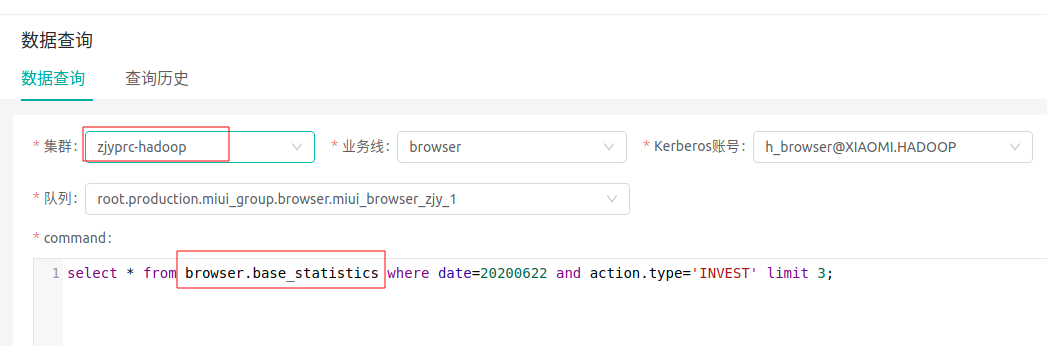
在测试环境应该查询：

select \* from base\_statistics\_staging where date=20200622 and action.type='INVEST' limit 3;



在正式环境应该查询：

select \* from browser.base\_statistics where date=20200622 and action.type='INVEST' limit 3;



1. 在新增激励金币时，需要做查重判断。防止接口恶意被刷（黑产攻击）。

查重：查询数据库，根据userId和imei 查询金币增加记录。如果因为同一个理由增加过金币，那么就不再重新增加，并返回提示信息。

1. 中台评论的优化总结

【优化文档】<https://wiki.n.miui.com/pages/viewpage.action?pageId=175125231>

1. since 不是索引，是一种查询方式，借助索引实现
2. 大部分索引都不要自定义
3. 简单的说 刚开始**since** 为空，第一次按主键查10条， 最后一条主键假设是10 ，第二次查询 since 传 10 将相当于 where id> 10 limit 10 的效果。
4. 这个since我可以理解为offset，查询的起点偏移量。就是你上次请求的最后一条的id--endMark。

15.misc-service在xbox上部署失败。查看部署日志发现check.sh生效，即服务在部署完2分钟后还没有起来。

解决过程：

(1)ssh登陆部署机器，查看进程是否起来：

ps aux|grep misc-service

发现有进程号，即进程已经启动

(2)查看tomcat启动日志

cd /home/work/bin/misc-service

vim run.logs

发现启动日志没有报错

(3)查看业务日志

tail -n 1000 misc-service.log 或

tail -f misc-service.log

发现业务日志一直在滚动（并不代表没有问题！！！！！）

(4)查看服务端口是否起来

lsof -i:8790

无响应，即服务没有起来。

服务是否起来要看这个服务对应的端口是否开始监听！！！！！！

(5)查看启动日志是否存在异常

**grep -i ‘exception’ misc-service.log**

Screenshot from 2020-07-18 11-02-16

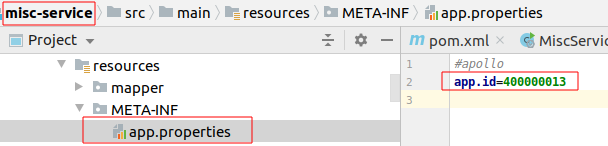
发现springboot bean 初始化异常

(6)在**单测**中debug测试程序（直接启动服务可能debug不到sping管理的bean的初始化问题），并在初始化异常的bean中打断点，找到bean初始化异常的位置。

发现这个bean在Apollo配置项获取为空，从而导致bean初始化异常。

原因：**每一个项目都对应一个Apollo配置，通过 app.id进行关联**。

当news-feed-api-springboot 部分功能迁移到misc-service时，如果涉及到Apollo配置获取，则应当将对应的配置迁移到对应项目的对应Apollo项目中。



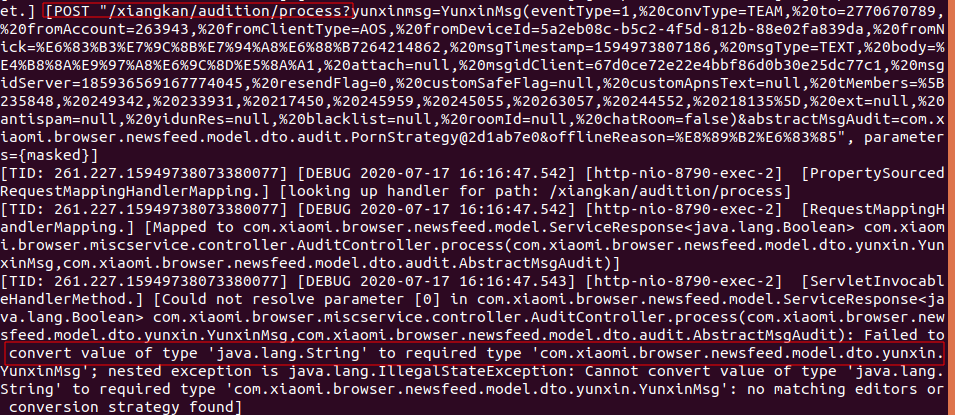


(7)如果单测失败，启动本地服务，确认本地是否启动成功。

**如果单测或者本地服务有一个能起来，说明代码无问题，可以push**

否则需要仔细查找问题!!!!!!

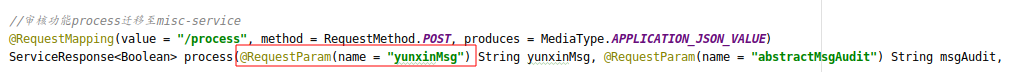
1. feign client远程调用时的传参问题



<https://www.cnblogs.com/wwct/p/12377994.html>

使用fein client进行远程调用时，需要注意一下几点：

(1)如果有多个参数，需要使用修饰府@RequestParam(name = **"xxxxx"**)

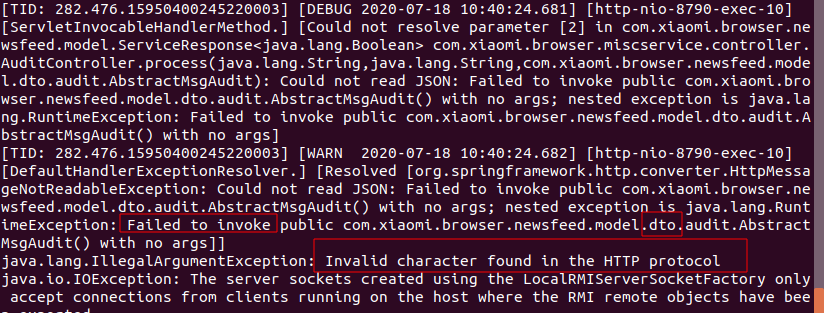


（2）如果要传递的参数不是基本类型，需要使用**@RequestBody**进行修饰



1. Feign client最多只传递一个**@RequestBody 对象参数**

**注意：**当这个对象里面包含接口类型的属性时，会报错400

****

如果将该对象先转成json进行传输，在feign server端，该接口类型的数据为空，在用json反向解析成obj时会报500的Inner Exception。

解决：前提(这个接口对象的属性与方法一旦初始化后就不再改变)

在用feign进行远程调用时，不传递包含该接口的对象，而是传递一个能够标识该对象的String标识

以想看审核功能的迁移为例：

**Feign终极解决办法：**

**发起者：用GsonUtil.toJson(obj)将所有的对象转成json串**

**接受者：用GsonUtil.parse(obj,class)将json转成对象**

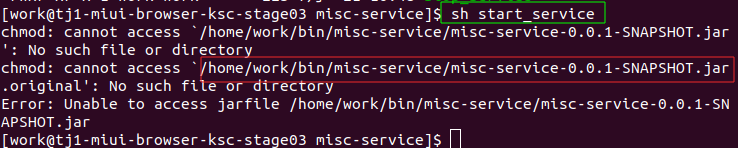
18.xbox编译失败，报错如下：



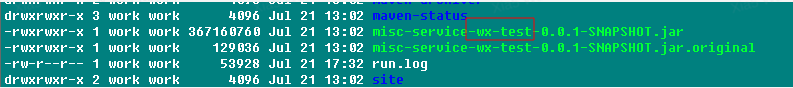
查看部署启动日志

cd /home/work/bin/misc-service

sh start\_service



ls -l



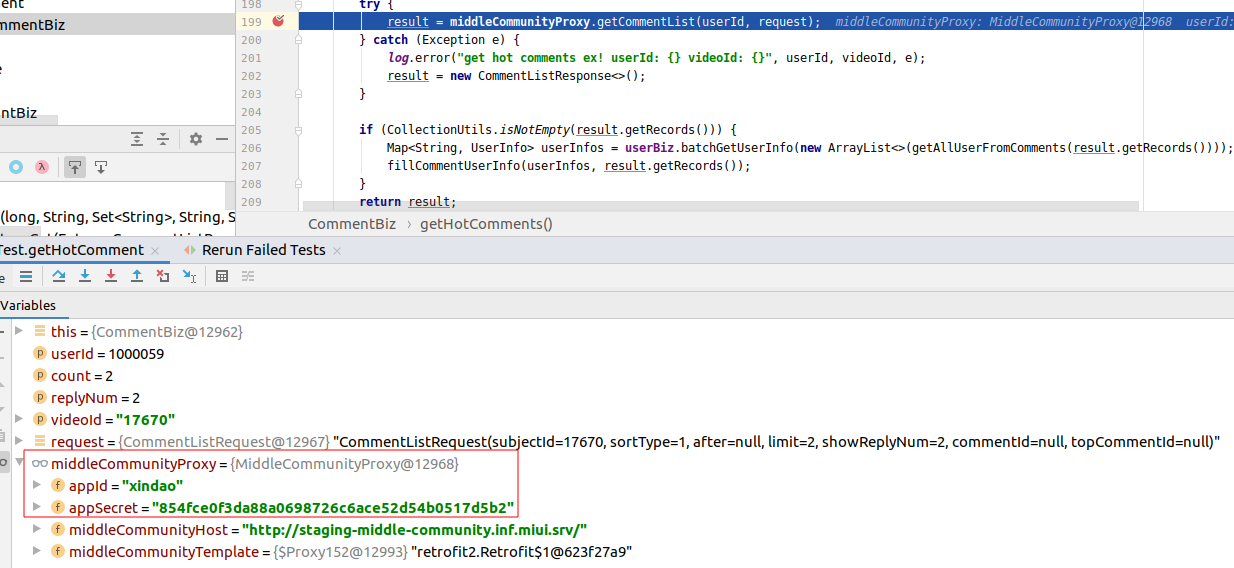
发现生成的项目名不对，导致xbox无法识别

原来是项目的pom工程名称写错了

19.代码逻辑没问题，但是远程调用经常失败(一半一半)

查看是否有熔断配置，修改熔断配置参数即可

1. 新岛问题总结
2. CommentBiz 199



middleCommunityProxy的属性值appId和appSecret的来源



2. 在实现添加弹幕总数的功能时，所有被@LegalRequestChecked

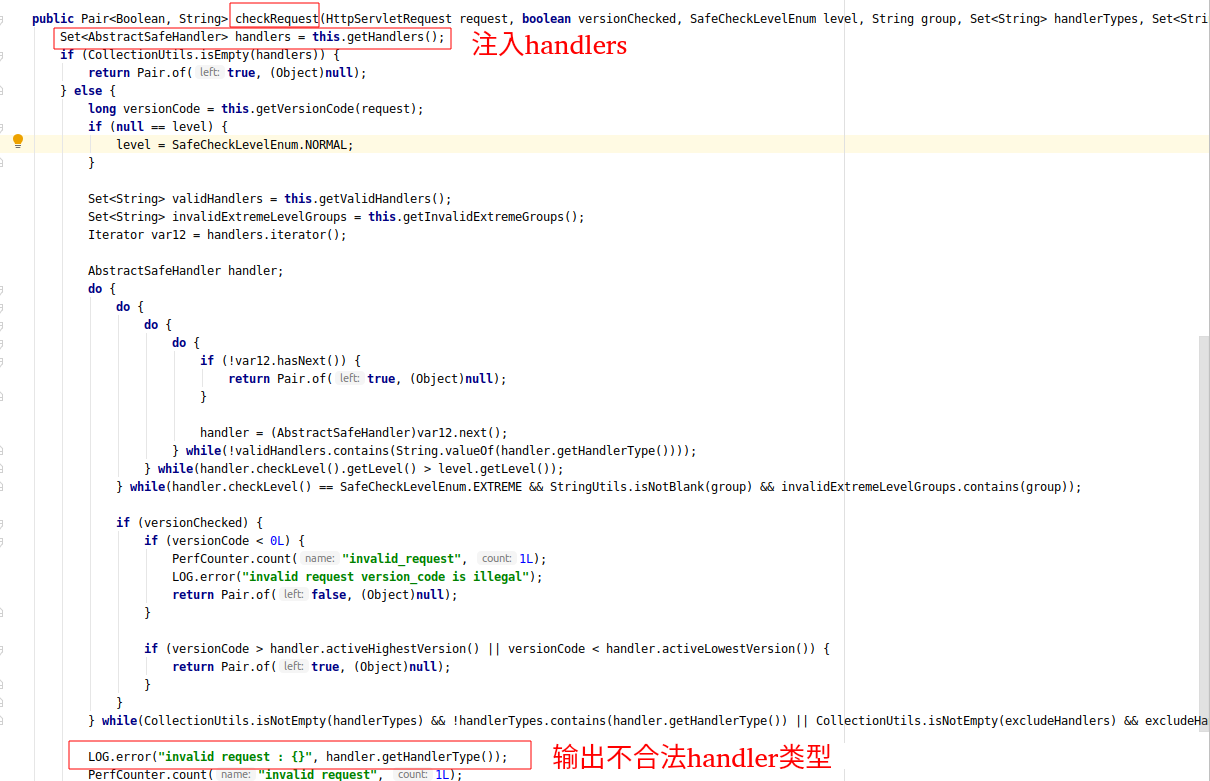
修饰的controller都返回”1002 非法请求”.

解决：

（1）找到 @LegalRequestChecked的拦截器LegalRequestCheckedInterceptor

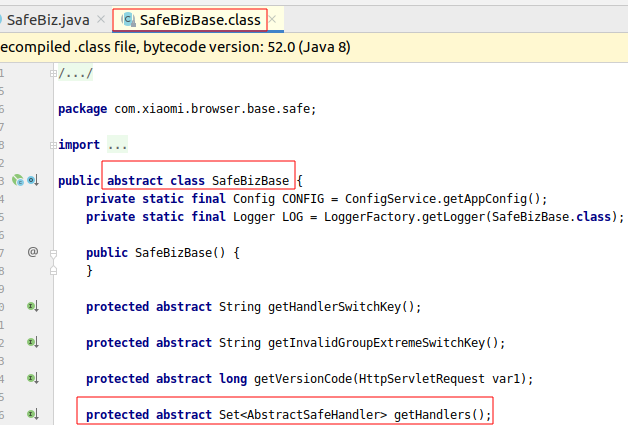


断点调试发现在safeBiz.checkRequest 返回 “(false,ip)”。查看safeBiz.checkRequest方法详情



在Log.error发现不合法的hander类型是IpCheckerhandler。

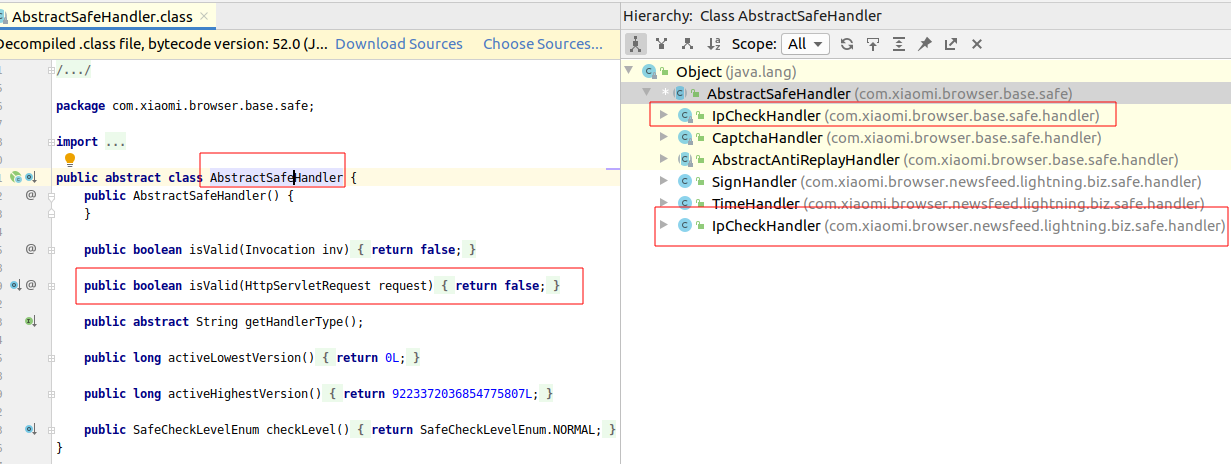
所有的hander注入在this.getHandlers()



this.getHandlers()是一个抽象方法，需要找实现类SafeBiz的实现方法。



SafeBiz的handlers是通过@Autowired类型注入的，即Spring会注入所有AbstractSafeHandler类型的handler。查看AbstractSafeHandler的实现子类：



发现有两个IpCheckHandler。一个是自定义实现，另一个是引入browser-base包时顺便引入的。

查看该类，发现该类没有重写isValid(HttpServletRequest request)

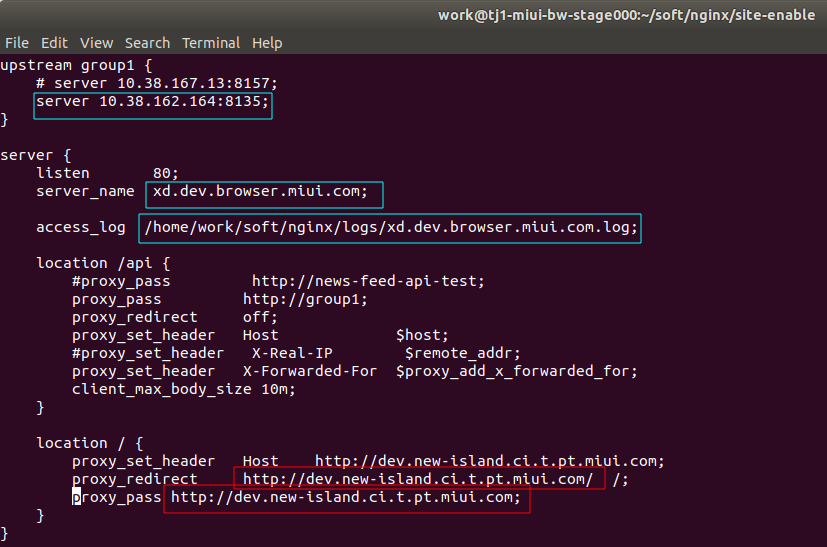
方法，因此直接继承父类的方法返回false,导致IP检查失败.

3.把测试环境的前端资源代理代理到http://dev.new-island.ci.t.pt.miui.com

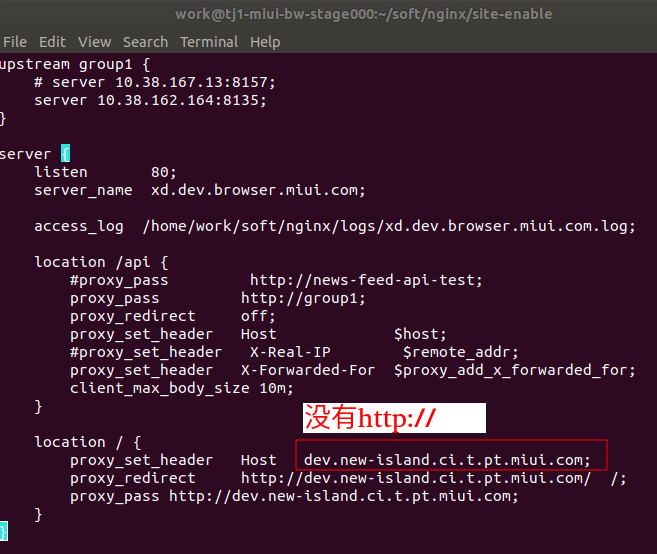
ssh 登陆机器：tj1-miui-bw-stage000.kscn

cd /home/work/soft/nginx/site-enable

vim lightning-service.conf



发现报404，正确的lightning-service.conf是



修改完成之后还需要让SRE将tj1-miui-bw-stage000.kscn上的nginx reload一下

4.在排查端外分享页的接口时，打断点发现接口能正常返回结果，但是在返回时还是会返回IO异常。

原因：服务响应慢，打断点导致响应超时了。

5.新岛部署线上机器，出现问题：

部署成功后，查看端口是否启动：

Screenshot from 2020-08-14 18-18-29

发现端口没有启动成功

解决：

使用netstat -ano|grep 8157

Screenshot from 2020-08-14 18-30-46

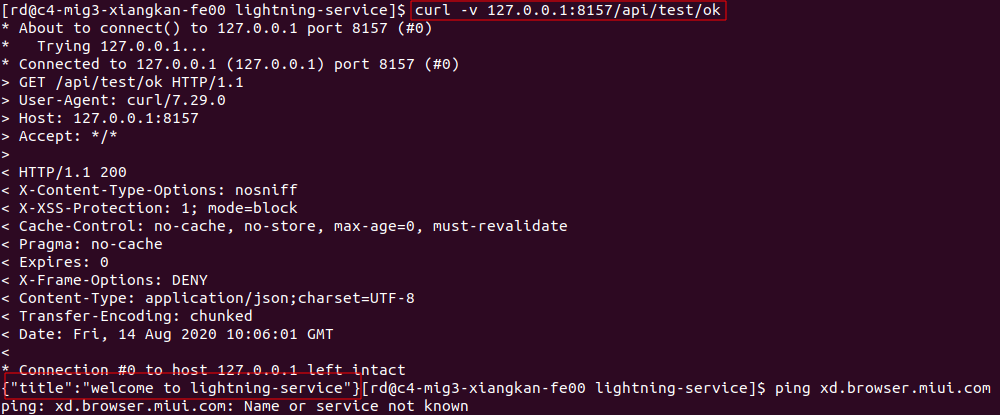
原因：

实际上服务已经起来了，只是使用lsof -i命令时的身份是rd，没有权限查看端口。

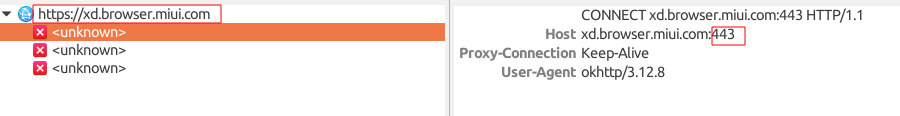
查看线上机器的服务是否起来(!!!!!!!)：

在已经启动的服务器上面发送请求，请求地址为localhost，即可实现对线上的服务进行判断

curl -v **127.0.0.1**:8157/api/test/ok



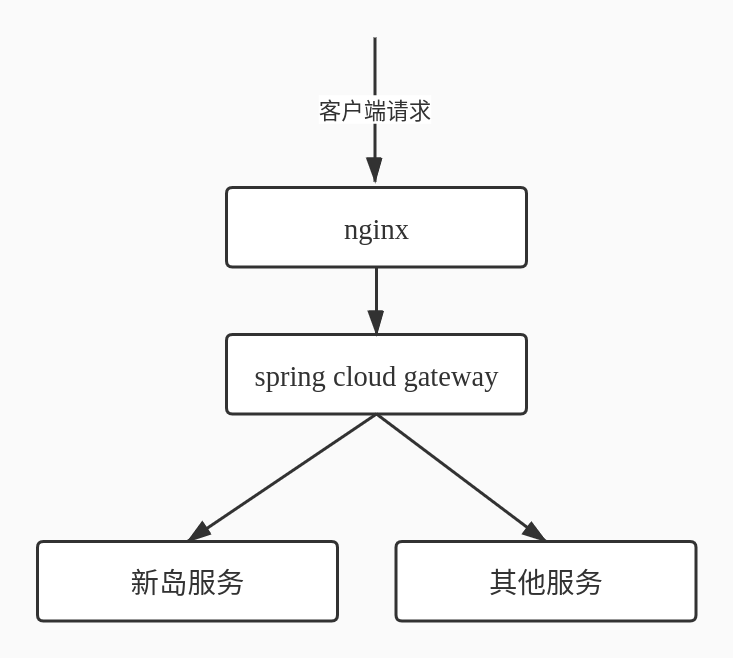
由于SRE没有配置域名解析，所以请求一直443



6.新岛服务接入spring cloud gateway网关new-business-gateway

|  |  |  |
| --- | --- | --- |
| 域名 | 测试环境 | 线上环境 |
| 新岛 | xd.dev.browser.miui.com  （已申请） | xd.browser.miui.com  （已申请） |
| 网关 | gateway.mig3.srv  （未申请） | dev.gateway.mig3.srv  （未申请） |

理论上：



即：请求经过nginx后，应该将服务转发到网关的服务。网关的路由配置如下：



即：网关是通过请求Header中的Host字段的值进行请求的分发。

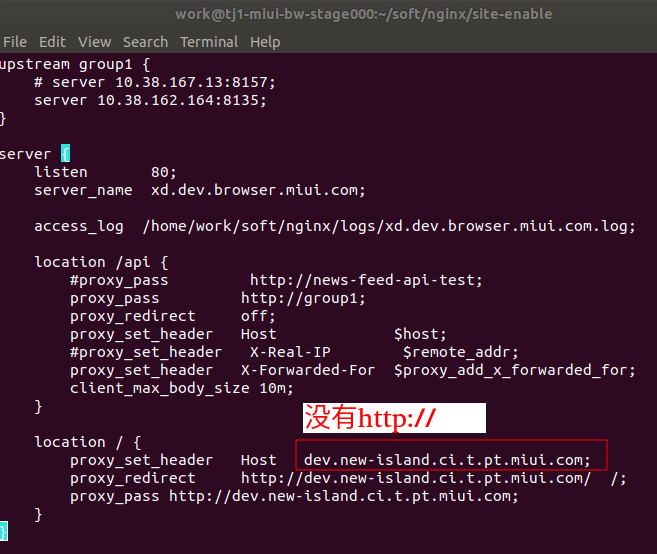
因此，我们**希望**的是请求经过nginx时，nginx首先将请求统一路由到网关，并在请求的Header的Host字段填入新岛服务。这样即实现了请求的统一接受，当网关接收到请求后又可以通过Host字段判断请求对应的微服务，从而进行下游服务的转发。

实际上：

1. 请求新岛服务，如果nginx转发到网关域名，会将host里面填入新岛域名。
2. 当nginx向gateway转发时,首先会查找请求Header中的Host字段所在的nginx集群。
3. nginx发现Host是新岛域名。新岛域名没有对应的nginx集群，直接返回403。

**解决：**

Nginx转发给网关时，不通过域名转发，直接ip导流。如新岛测试环境的nginx配置就是：



当请求过来时，代理地址是http://group1。代理头是请求的域名

而group1对应的地址是10.38.162.164:8135 (8135是网关端口)

即请求过来时，Nginx直接向网关转发请求。

7.接小米通道的自动化push

1. 自动化push接入的是通用后台.

<https://wiki.n.miui.com/pages/viewpage.action?pageId=403316510>

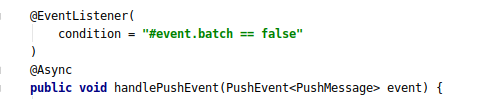
**服务端Push 流程：**

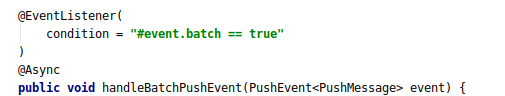
Spring constructPushMessage

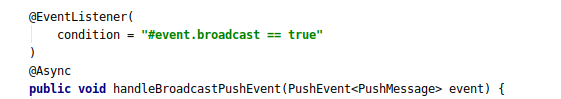
Spring eventPublisher.publishEvent(event)

Spring 监听消息PushEventProcessor

(参考Spring 事件监听机制)







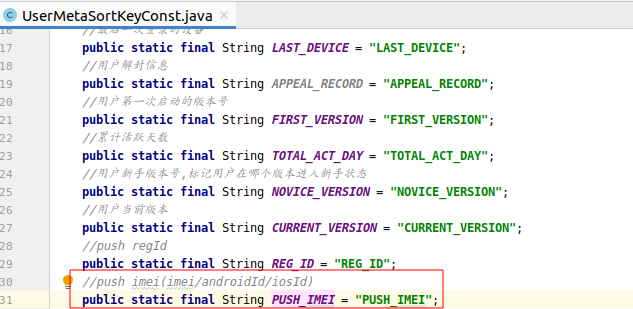
根据@EventListener中的condition判断消息是单发，批量发还是广播发送。但消息的真正发送是在news-feed-common.PushBaseOper.doSendPushMsg.根据regId/Imei/Oaid接入小米的通道进行消息的真正的发送。

注意：

新岛采用的是deviceId(androidId/iosId)，但是在接小米通道发送的时侯只能接入imei通道进行发送，接入别的通道会产生问题。

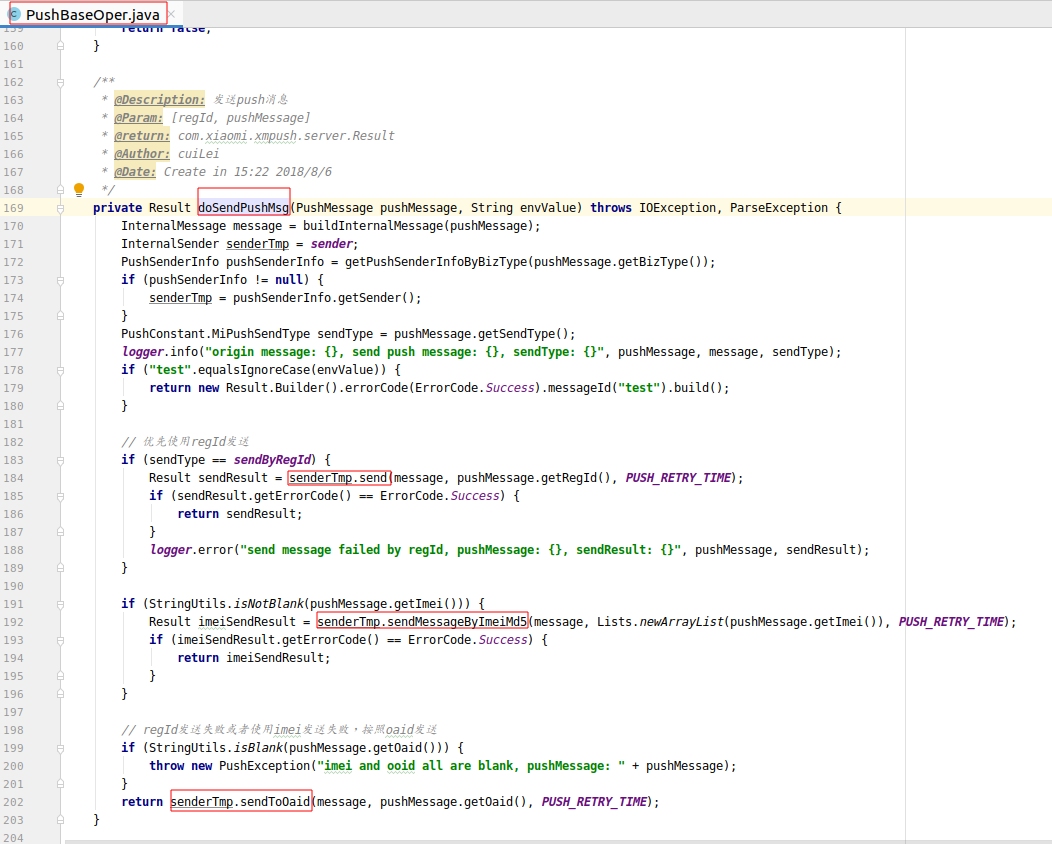
在发送push之前，还需要进行id注册。新岛具体的实现在：/push/register

关键在于：将要push的id写入Pegasus。

后续判断从哪个通道发送的时侯：



消息的真正的发送还是使用小米通道方式进行消息发送



问题：使用小米push通道时服务端和客户端需要先申请APPID和APPKEY：



这个sender指明了发送的app地址。因此使用对应的sender会发送对应app的Push消息.

|  |  |
| --- | --- |
| APPID | 2882303761518633246 |
| APPKEY | 5601863378246 |
| APPSECRET | I3JFzLY2A4bbZtnjnU49bg== |



问题:

（1）新岛的审核分机审和人审。先机审再人审。机审和人审的审核回调都是在misc-service服务(

域名：

测试：staging.mig3-new.browser.miui.srv

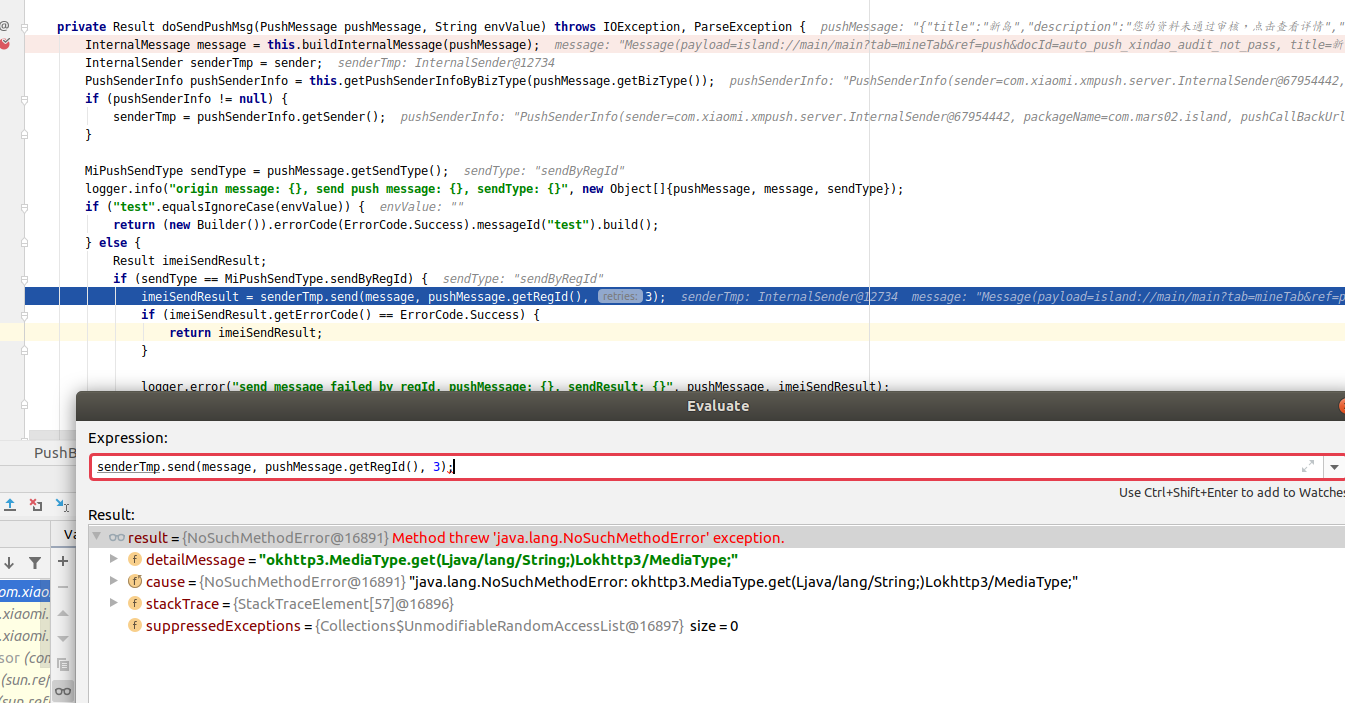
线上：misc.mig3-new.miui.srv

)

的AuditCallbackController.请求地址为：/audit/callback

最终的审核结果以人审为主。

1. 同样的代码，同样的入参,在使用小米sender进行发送时，发生异常：



这个原因是因为okhttp的版本过低导致的。应该使用3.14.2

8.在新岛个性化签名编辑中，出现了两个问题：

1. 当用户更新的签名中包含”&”时，就会自动截断签名。

比如用户准备更新的签名是”&&&”。即传入的是sign=”&&&”,userId=1000059

这个参数在进行远程传输时，就会被解析成：**sign=&&&userId=1000059**

**即：sign=null,userId=1000059.**

原因：

在对字符’&’进行远程传输解析时，将其解析为参数的分隔符号，而不是字符串。

解决：

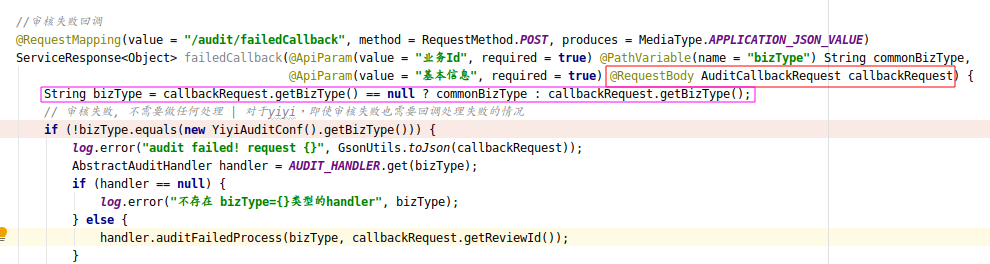
检测输入是否有特殊字符，禁止并提示输入或者对字符’&’进行转义(“\”)--未测试

1. 远程参数传递的问题

审核失败时，回调misc 的callback方法时需要将 callbackRequest参数传递给Uni-user-service



Uni-user-service接受参数时需要使用 **@RequestBody修饰，**否则不能正确接受参数,且此时的commonBizType = misc。真正要审核的bizType = callbackRequest.getBizType()



9.智能增长平台的需求：在第三方（如浏览器）上投放关于新岛的广告，从而为新岛引入新用户。

代码：miuibn-common-api项目FeedExchangeController的getXindaoAds

<https://wiki.n.miui.com/pages/viewpage.action?pageId=307794833>

流程：

服务端接受请求参数：

http://ad.dev.browser.miui.com/api/ad/xindao?**request\_entity**={"deviceInfo":{"screenWidth":0,"screenHeight":0,"power":0,"totalSpaceKB":0,"freeSpaceKB":0},"userInfo":{"imei":"86e54c77836773ba"},"appInfo":{"platform":"xiaomi","packageName":"com.android.browser","version":"120825000"},"impRequests":[{"tagId":"3","scheduleId":"215","width":0,"height":0,"adsCount":0,"bizType":92}]}

服务端返回类型：FeedsExchangeResponse



浏览器拿到数据后，填充并作为广告显示到浏览器界面上。

此时从Apollo拉取数据：

|  |
| --- |
| 正确的用法：  *xkScheduleIdAndAdDataMap* = **GsonUtil.*getInstance*().fromJson**(xkApolloSrcData, **new** TypeToken<Map<String, FeedsExchangeResponse.FeedsExchangeAdInfo>>() {}.getType()); |
| 错误的用法：  *xdScheduleIdAndAdDataMap* = **GsonUtils.*parse***(xdApolloSrcData, **new** TypeToken<Map<String, FeedsExchangeResponse.FeedsExchangeAdInfo>>() {}); |

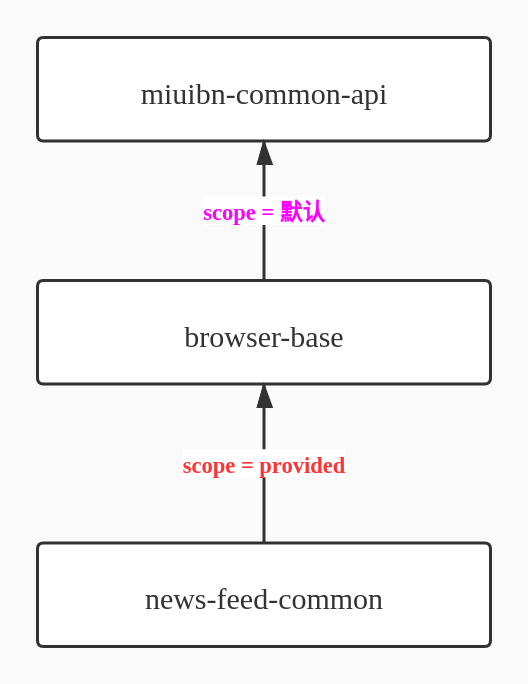
这是因为：

GsonUtils中引入了GsonIgnore.class.

**import** com.xiaomi.browser.newsfeed.model.GsonIgnore;

这个GsonIgnore来源于news-feed-common。而当项目仅仅依赖了browser-base，没有依赖news-feed-common时。编译器会通过，但是运行时，会因为找不到类而导致服务启动失败。

**引用关系**如下图所示：



<https://blog.csdn.net/qq_36874292/article/details/81072717>

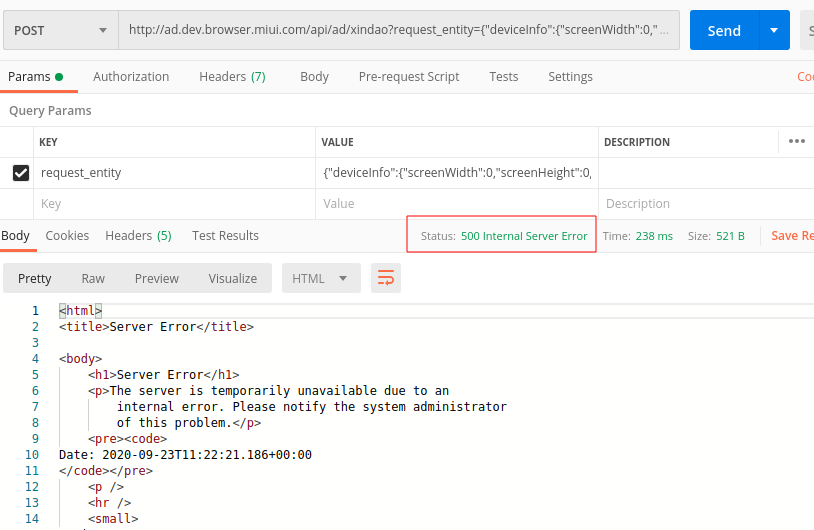


因为在browser-base中引用news-feed-common的scope=provided，所以在miuibn-common-api中无法引用到news-feed-common包。

**两个关键点：**

(1)news-feed-common是以scope=provided的方式引入browser-base的。因此browser-base 打包后的jar文件中不存在news-feed-common.

(2)在miuibn-common-api工程**编译（将Java源程序编译成class字节码文件）**的时侯，由于browser-base是以jar包的方式引入的，因此不会参与miuibn-common-api的编译过程---》编译成功。而打包时不会再检查jar包内的依赖关系---》打包成功。运行时，由于browser-base jar包缺少news-feed-common jar包。因此当需要使用news-feed-common中的GsonIgnore 时，会报类找不到异常。



**注意：**

**在服务启动失败时，应该查看服务的run.log**

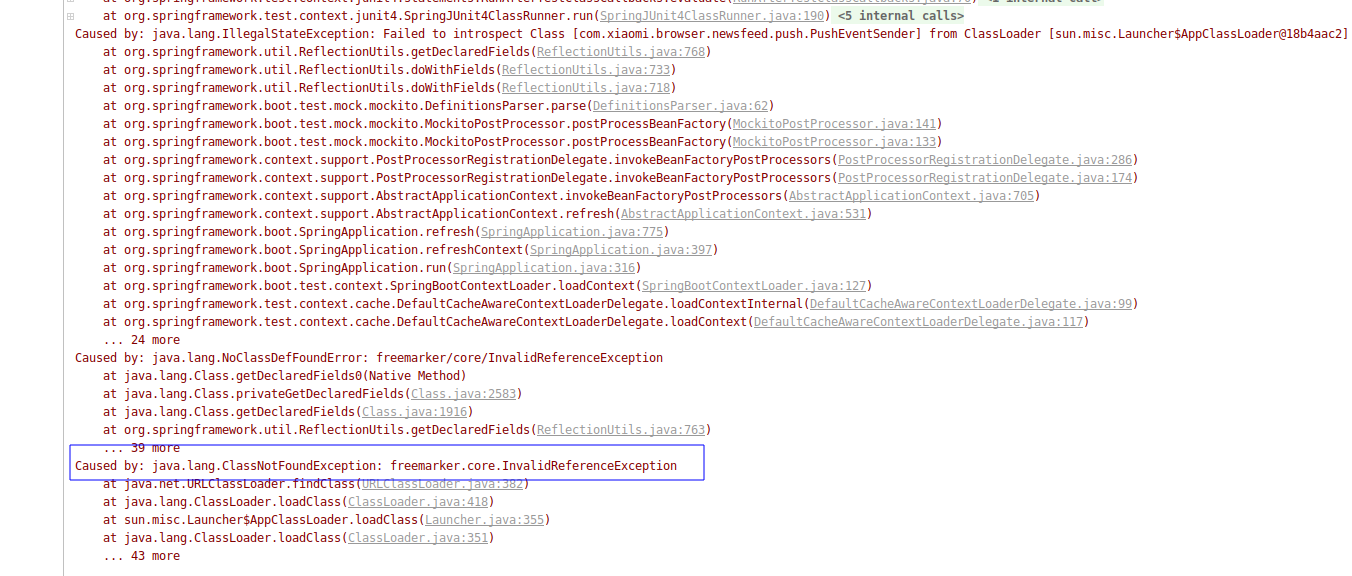
**在服务运行失败时，应该查看业务日志xxx.log**

**但是！！！：**

**在最新版的browser-base（1.5.74）和news-feed-common（1.1.30）中，只在browser-base中保留了GsonUtils工具类和GsonIgnore注解。**

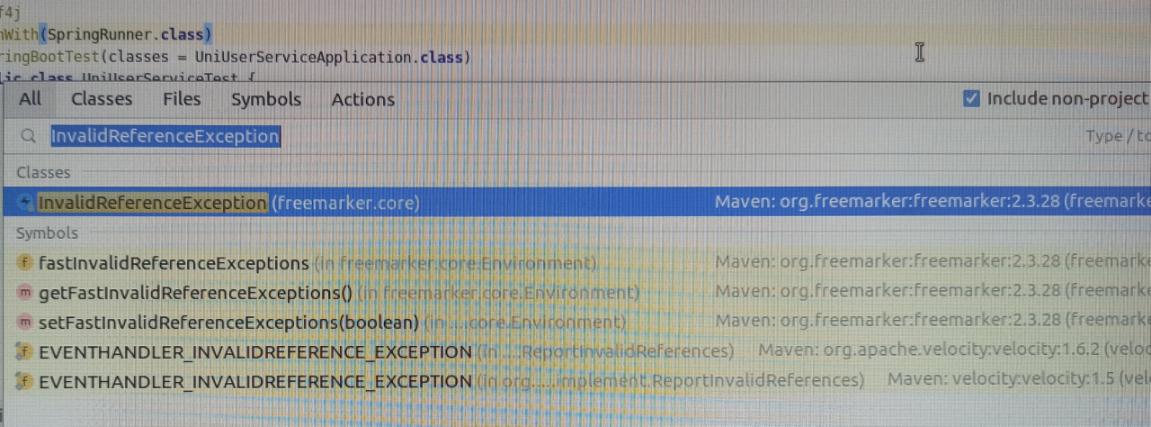
**删除了news-feed-common中的GsonUtils和GsonIgnore，并且在browser-base中添加了GsonIgnore的注解。解决了这个问题。**

10.在重构新岛审核回调时。问题：测试包news-feed-common wx-1.1.34的内容对应着master包news-feed-common1.1.32。但是在代码中，使用wx-1.1.34版本时，Spring 容器启动正常；而使用news-feed-common1.1.32时，Spring 容器启动失败。报错信息如下：



解决问题的步骤:

1. 找到InvalidReferenceException的引用出处.以防出现多个jar包冲突。



1. 排除冲突的包

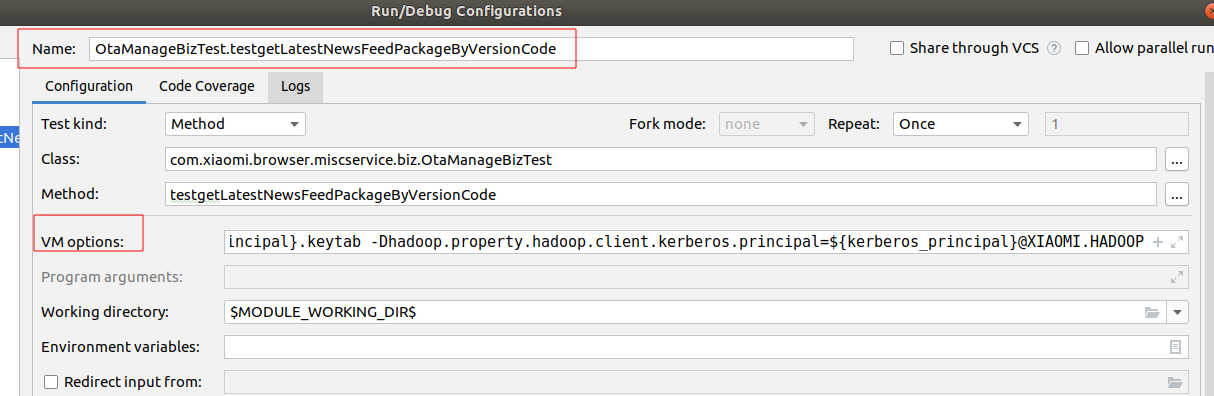
**原因：**

本地原先存在news-feed-common1.1.32包，该测试包并不是master上的最新的news-feed-common1.1.32。当maven进行包导入时，导入的是本地非最新的包，因此导致Spring 容器启动异常。

**解决：**

将本地的news-feed-common1.1.32删除后，再从远端仓库拉取最新的news-feed-common1.1.32包，该包才是真正的master。

11.在本地启动misc-service失败的解决办法。



选择测试的方式Name，并在VM options中填入以下信息即可：

-Dspring.profiles.active=staging

-Denv=DEV

-Didc=staging

-DdataCenterId=0

-Dhadoop.property.hadoop.security.authentication=kerberos

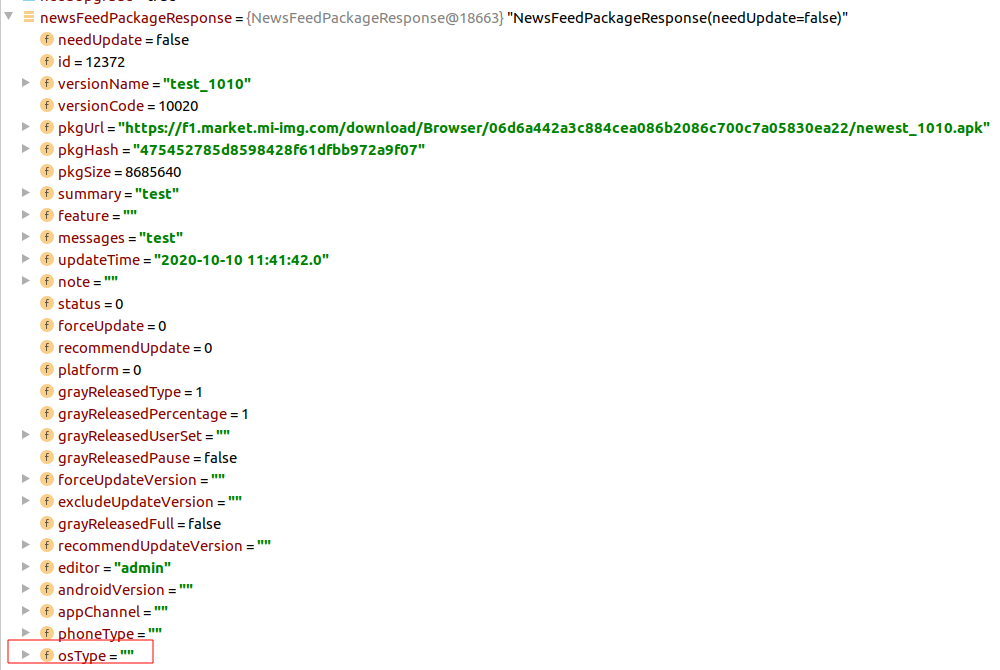
-Djava.security.krb5.conf=/home/mi/krb5-hadoop.conf

-Dhadoop.property.hadoop.client.keytab.file=${conf\_path}/${kerberos\_principal}.keytab

[-Dhadoop.property.hadoop.client.kerberos.principal=${kerberos\_principal}@XIAOMI.HADOOP](mailto:-Dhadoop.property.hadoop.client.kerberos.principal=${kerberos_principal}@XIAOMI.HADOOP)

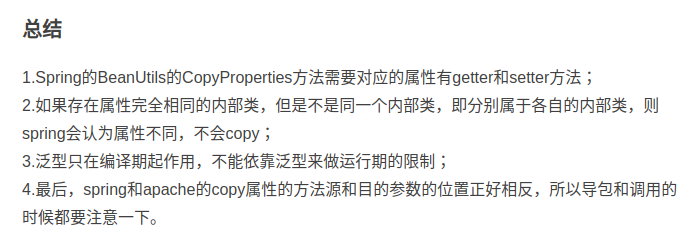
12. 在使用BeanUtils.copyProperties 拷贝对象的属性时，出现部分字段没有拷贝的情况，如下图所示：



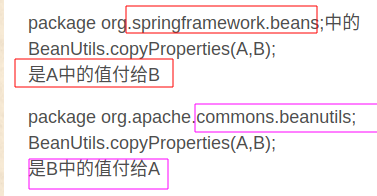


有两个可能的原因：

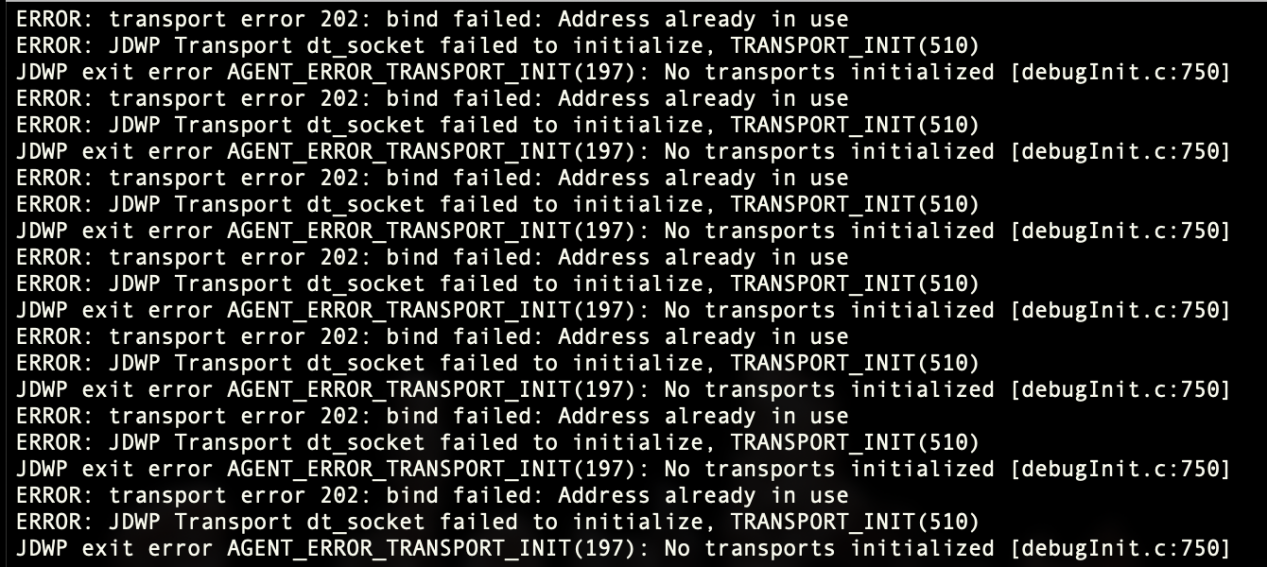
<https://www.jianshu.com/p/357b55852efc>



<https://blog.csdn.net/weixin_44550490/article/details/104724903>



13.misc-service服务启动失败，报错如下：

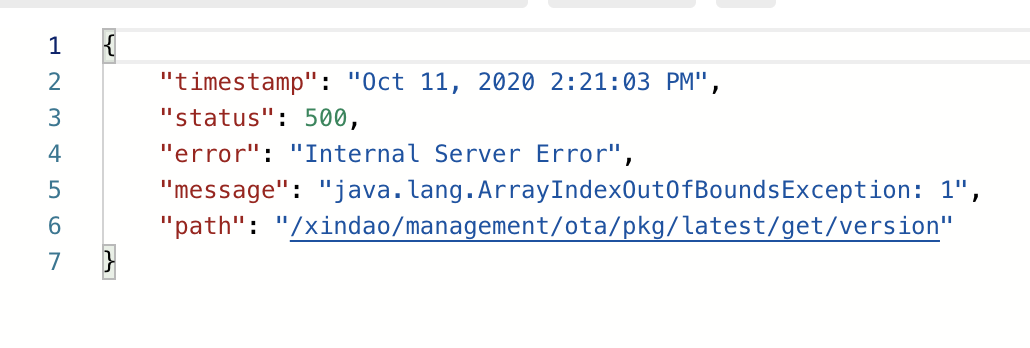


原因：

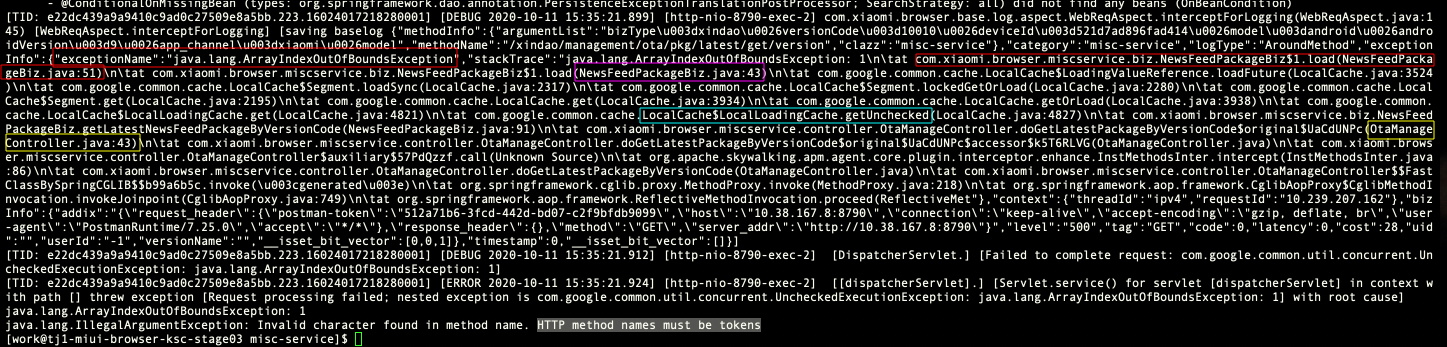
Jvm debug端口冲突

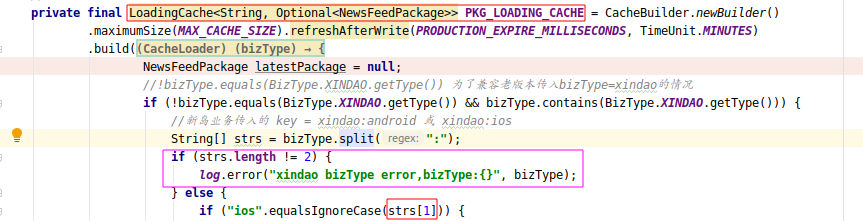
14.在misc-service项目**/{bizType}/management/ota/pkg/latest/get/version/new**

出现异常：



登陆机器，查看日志信息：





原因：Guava cache使用unchecked时，如果不加strs.length!=2判断，而直接使用strs[1]时就会报下标越界异常。如果加了strs.length!=2判断，那么进入else语句内使用strs[1]就是合法的。

15.开发的过程中遇到了一个Redis的问题: 当你在redis中插入一个key值,并且设置了对应过期时间. 当过期时间还没到的时候重新 更新 key值会导致 过期时间被刷新

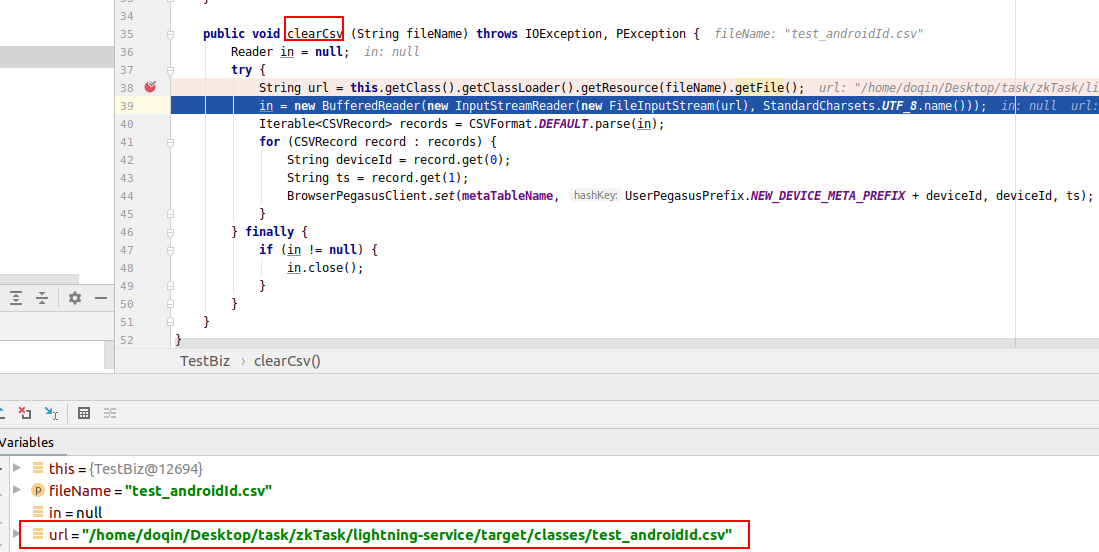
<https://blog.csdn.net/itguaicai/article/details/100569855>

如果用DEL, SET, GETSET会将key对应存储的值替换成新的，命令也会清除掉超时时间；如果list结构中添加一个数据或者改变hset数据的一个字段是不会清除超时时间的；如果想要通过set去覆盖值那就必须重新设置expire。

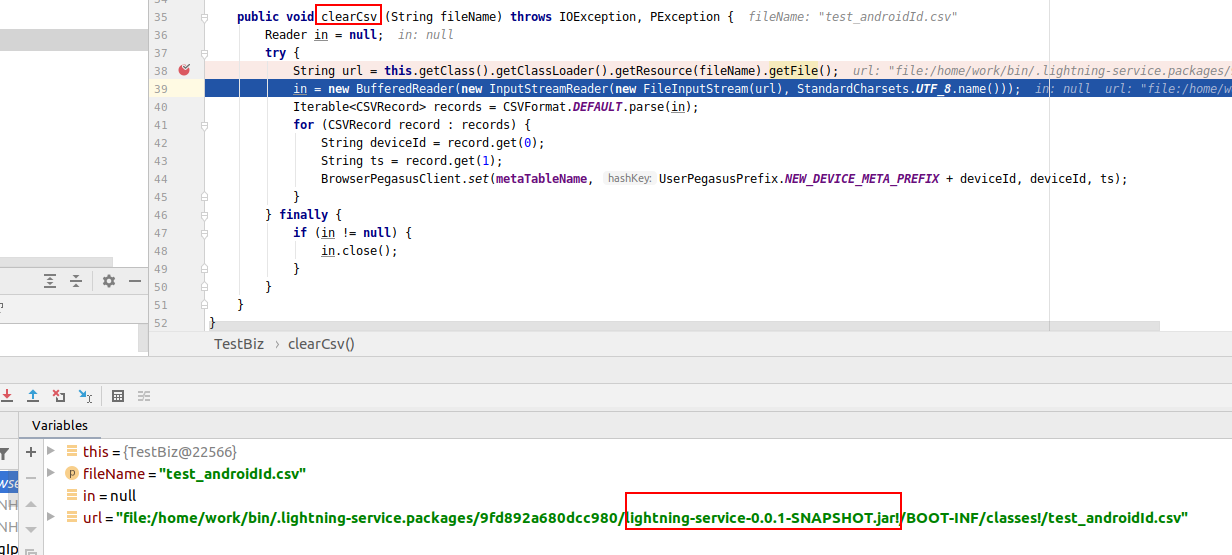


16.在清洗用户的设备ID数据时。需要读取maven项目文件src/main/resources/test\_androidId.csv。

在本地执行文件读取时，能够正确读取到文件并执行：



将代码部署到测试环境时侯，依然报错：FileNotFoundException，且此时文件的读取路径为：



此时的文件路径在JAR包里面。

**原因：**

1. 在xbox上部署后，工程先打成JAR包，再执行JAR文件

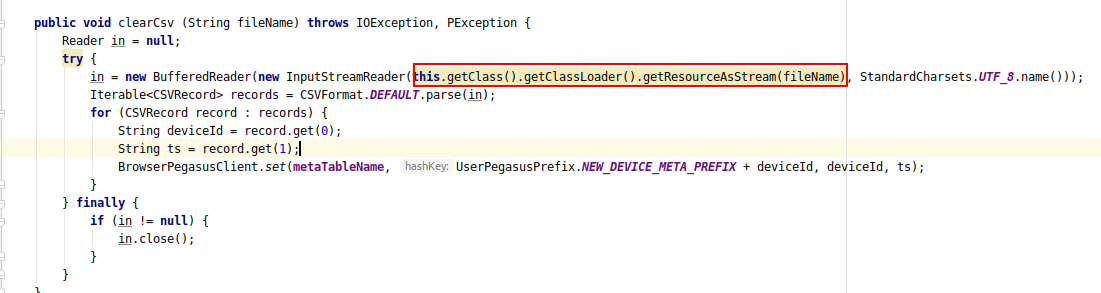
****jar包是一个单独的文件而非文件夹，绝对不可能通过"file:/home/.../lightning-service-0.0.1-SNAPSHOT.jar/target/classes/**test\_androidId.csv**"这种形式的文件URL来定位。****

<https://www.iteye.com/blog/hxraid-483115>

**解决：**

用类装载器(ClassLoader)来读取

Reader in = **new** BufferedReader(**new** InputStreamReader(**this**.getClass().getClassLoader().getResourceAsStream(fileName), StandardCharsets.***UTF\_8***.name()));



17.在实现作者头像异常报警时，出现过以下问题：

（1）执行工程lightning-service-worker，Springboot启动时报错信息如下：

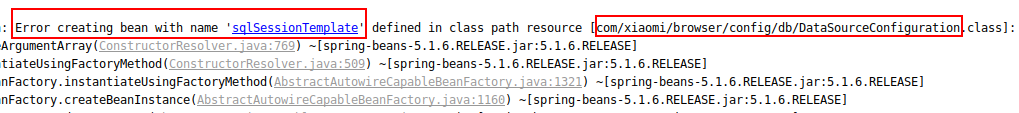
Error starting ApplicationContext. To display the conditions report re-run your application with 'debug' enabled.

**解决：**

配置执行的VM：

-ea -Denv=DEV -Didc=staging -Dspring.profiles.active=staging

（2）无法创建无法创建mapper接口的原因是无法创建sqlSessionTemplate，问题执行时截图如下：

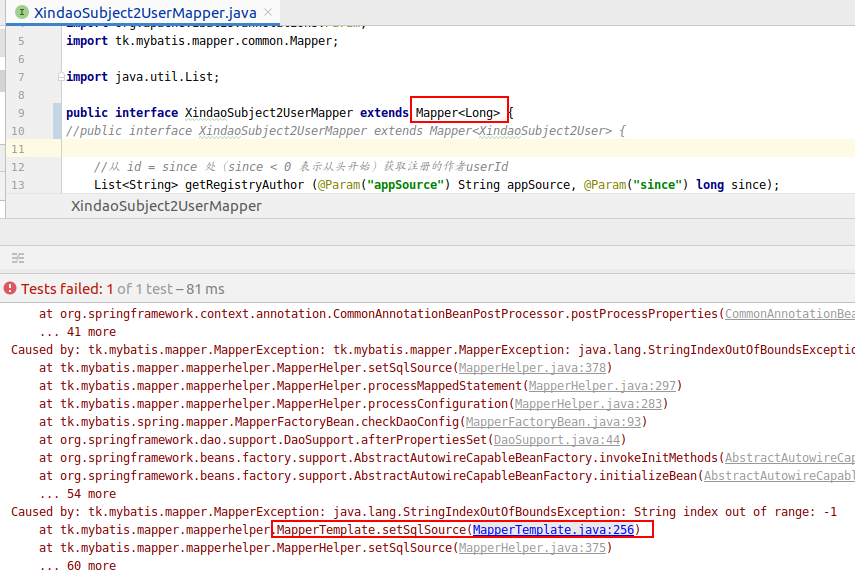


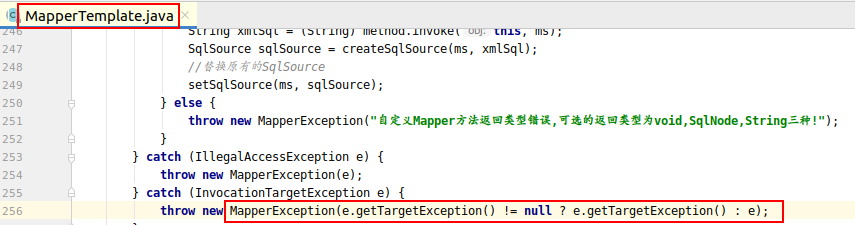
**解决：配置**执行的VM:

-ea -Denv=DEV -Didc=staging -Dspring.profiles.active=staging

|  |
| --- |
| **public interface XindaoSubject2UserMapper** **extends Mapper**<**XindaoSubject2User**> {}  Screenshot from 2020-11-26 16-12-59 |
| **@Table**(name = **"XindaoSubject2User"**) **public class XindaoSubject2User** {}  Screenshot from 2020-11-26 16-13-19 |

原因：接口XindaoSubject2UserMapper继承**Mapper<T>**,T指定了返回的数据类型。比如定义：

****

****

第一个方法返回一个list，指定返回类型是String(一个)，报下标异常错误

(3)创建了XindaoSubject2User，在mysql执行sql转成XindaoSubject2User

对象赋值时发现无法正确赋值。

原因：

在XindaoSubject2User中，@Column是JPA接口的数据库model转换注解。由于我们使用的是Mybatis，因此需要在**XindaoSubject2UserMapper加上数据库到java的转换**

|  |
| --- |
| <**resultMap id="BaseResultMap" type="com.xiaomi.mig3.feeds.lightning.worker.dao.model.XindaoSubject2User"**>  *<!--  WARNING - @mbggenerated  -->* <**id column="id" property="id" jdbcType="BIGINT"**/>  <**result column="user\_id" property="userId" jdbcType="BIGINT"**/>  <**result column="org\_user\_id" property="orgUserId" jdbcType="VARCHAR"**/>  <**result column="middle\_user\_id" property="middleUserId" jdbcType="DECIMAL"**/>  <**result column="user\_name" property="userName" jdbcType="BIT"**/>  <**result column="org\_user\_name" property="orgUserName" jdbcType="TIMESTAMP"**/>  <**result column="icon\_url" property="iconUrl" jdbcType="VARCHAR"**/>  <**result column="org\_icon\_url" property="orgIconUrl" jdbcType="VARCHAR"**/>  <**result column="user\_source" property="userSource" jdbcType="VARCHAR"**/>  <**result column="app\_source" property="appSource" jdbcType="VARCHAR"**/>  <**result column="create\_time" property="createTime" jdbcType="BIGINT"**/>  <**result column="update\_time" property="updateTime" jdbcType="BIGINT"**/>  <**result column="extra" property="extra" jdbcType="VARCHAR"**/>  </**resultMap**>   <**select id="getRegistryAuthor" resultMap="BaseResultMap"**>   </**select**> |

这样XindaoSubject2User就能正确接收数据库的值了

18.在实现作者头像监控及更新（定时任务）时，在本地执行能够正确获取不合法的作者列表。在xxl-job调度平台点击执行时，得到错误的列表。

**原因：**

在任务中定义了一个全局变量*LAST\_UPDATE\_INDEX*用来记录上次扫描的最后一条记录的id。

在本地执行时，每次执行，LAST\_UPDATE\_INDEX都是从0开始计数（多次调度对应多次服务）

在xxl-job调度时，LAST\_UPDATE\_INDEX保留了上次调度的执行值（服务没有停止，多次调度在一个服务中）。

**解决：**

*每次调度时，对LAST\_UPDATE\_INDEX清零即可。*

18.在对接中台社区运营日志时需要使用mybatis的updateByPrimaryKeySelective方法来更新对象。

现象：在运营平台执行板块更新操作时，数据库的对应数据同步有延迟。

原因：使用mybatis的updateByPrimaryKeySelective方法更新时，对象必须带有主键信息。

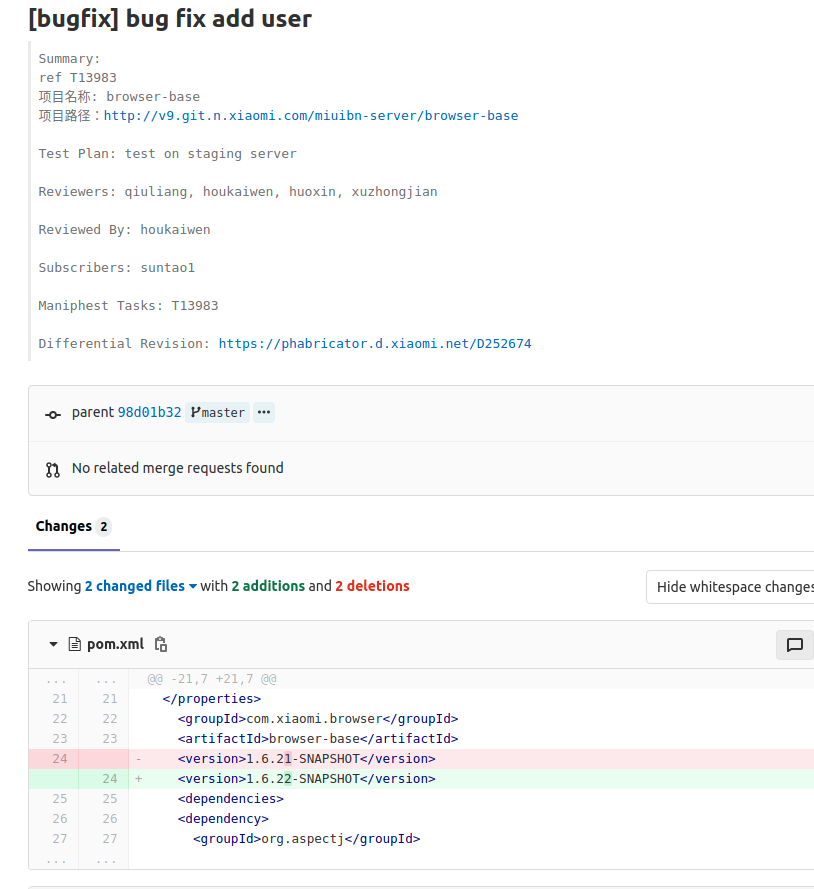
从中台账号获取的板块详情中并没有数据库中对应的id主键信息。因此更新不成功。

在测试环境出现延迟性成功的原因：

先前已有的定时任务再执行同步。

19.在中台账号与中台社区账号数据同步中，发现新增社区用户方法一直报错：无效的签名。但是肆样已经在线上使用了该方法。

**原因**：版本问题。Browser-base1.6.22之前这个新增方法有问题.1.6.22 修复了这个bug.



**过程：**

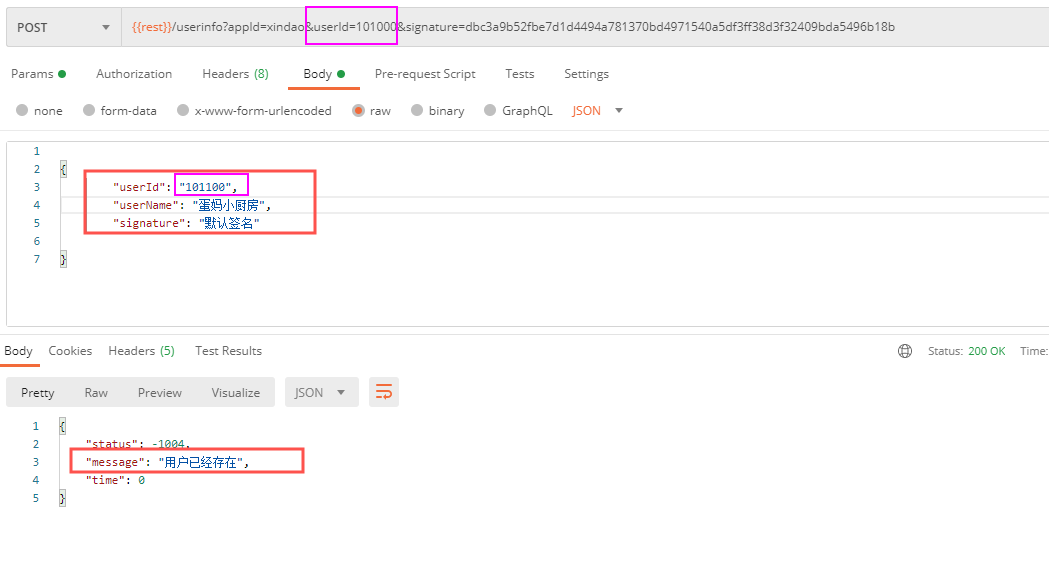
新增用户接口的签名 = 加密(userId+appSecretKey)。如果是这种方式的签名，需要在请求的参数里面带上userId。

或者

用户接口的签名 = 加密(appSecretKey)。如果是这种方式的签名，不需要在请求的参数里面带上userId。

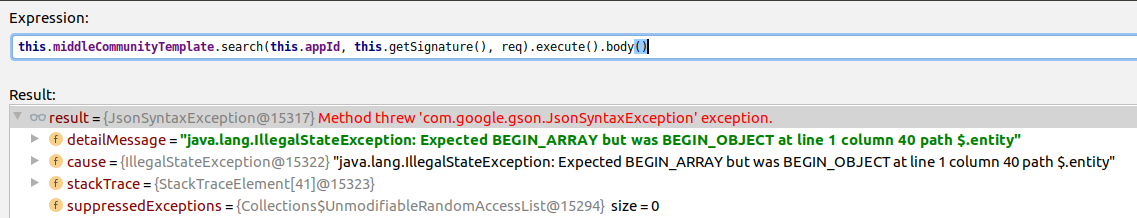
这是中台社区的研发使用postman发送的请求。

旧版本签名失败的原因：签名 = 加密(userId+appSecretKey)。但是请求的参数没有传入userId，导致验签失败。（查看execute的请求url）

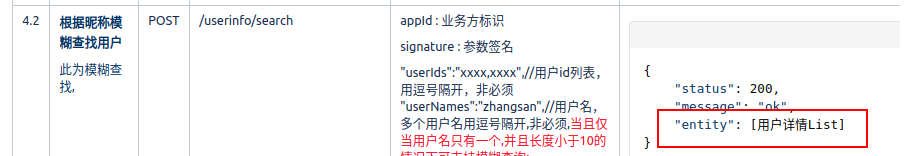


注意：请求的参数与签名是否匹配。

19.在使用中台社区的搜索方法时，一直报错。报错信息如下：



社区的接口文档如下：



原因：中台社区的接口文档中的返回值有误

**查看stackTrace**，发现是返回值解析报错。

排查过程：

使用Postman模拟发送搜索请求，结果如下图所示：



发现实际返回的类型确实跟文档不相同。

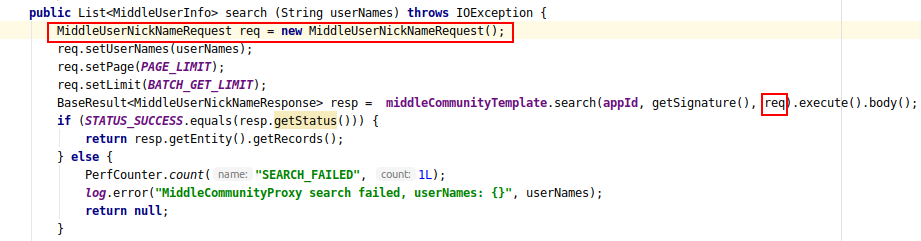
更新返回值类型后，搜索某个昵称，发现返回空。**查看返回值中的请求参数**：



发现搜索条件多出了gender = 0。使用postman加上gender = 0，发现加上该限制条件后，返回确实为空。同时发现，这是因为将gender的类型定义为int,默认值为0所导致的。而在返回值中，gender为string类型，默认为“”。使用postman加上gender=“”后可以查到正确结果。

注意：

在retrofit2中，Post请求中body的参数以json形式传递时，应该将请求的body参数封装成一个java类，如MiddleUserNickNameRequest，然后在请求时加上@Body。



Screenshot from 2020-12-22 20-59-49