**开发避坑指南**

1. **代码篇**

**1.1 分页问题**

部分查询不需要分页，部分查询需要分页（sqlmap中一定添加分页过滤语句）

物流 framework 的 BaseQuery 的 分页 pageSize 默认为50，如果是不需要分页的场景，一定要加

setIgnorePage来忽略分页

|  |
| --- |
| Java  productSupplierRelQuery.setIgnorePage(true); //忽略分页 |

如上如果需要分页查询，请检查sqlmap中如下条件不可少：

|  |
| --- |
| XML  <sql id="queryPageable">  <if test="!ignorePage">  limit #{startIndex}, #{pageSize}  </if>  </sql> |

**1.2 考虑是否加锁**

1. 如果后台写操作没有考虑幂等性，需要考虑页面操作是否加锁（可以后端使用Redis分布式锁实现）
2. 后台关键公共调用方法也要考虑是否加锁

**1.3 避免Dubbo的提供方限流设置与泛化调用同时出现**

设置Dubbo提供方限流设置如下：

|  |
| --- |
| XML  <dubbo:provider executes="5000"/> |

Dubbo调用设置以上限流设置的服务：

|  |
| --- |
| Java  GenericService barService = (GenericService) applicationContext.getBean("barService");  Object result = barService.$invoke("sayHello", new String[] { "java.lang.String" }, new Object[] { "World" }); |

请避免Dubbo的提供方限流设置与泛化调用同时使用，接口配置了全局的executes参数后，对并发请求接口做了限制，跟踪代码发现接口请求计数不会扣减。这是Dubbo的一个bug，请参考如下：

[配置<dubbo:provider ... />的executes参数引发的bug · Issue #5052 · apache/dubbo](https://github.com/apache/dubbo/issues/5052)

**1.4 大数据量报表从TiDB查询**

参考：

logistics-data-job.BOHStockJobHandler

**1.5 防止TiDB同步失败MySQL DDL操作尽量避免以下操作**

1）删除字段

2）修改字段类型

3）将字段长度调小

4）修改主键

5）添加或修改多个字段，避免加括号

6）禁止使用case when

错误写法：

|  |
| --- |
| SQL  ALTER TABLE `reservation\_order` ADD COLUMN  (box\_qty decimal(13,3) DEFAULT NULL COMMENT '预约箱数',  forecasting\_pallet\_qty int(11) DEFAULT NULL COMMENT '预计托盘数',  sku\_item\_count int(11) DEFAULT NULL COMMENT '商品行项数') |

正确写法：

|  |
| --- |
| SQL  ALTER TABLE `reservation\_order`  ADD COLUMN box\_qty decimal(13,3) DEFAULT NULL COMMENT '预约箱数',  add column forecasting\_pallet\_qty int(11) DEFAULT NULL COMMENT '预计托盘数',  add column sku\_item\_count int(11) DEFAULT NULL COMMENT '商品行项数'; |

**1.6 查询数据库是否存在记录**

不要使用：

|  |
| --- |
| SQL  select count(1) from table where column = xxx  就算column 是索引字段也不建议用 count 来查询数据是否存在 |

建议使用：

|  |
| --- |
| SQL  select id from table where column = xxx limit 1  如果返回值>=1 则说明数据存在 |

**1.7 Java 8 Lambda表达式问题**

**1.7.1 List映射Map注意事项**

使用Collectors.toMap时需要记住几点：

|  |
| --- |
| Java  key不能有重复，否则会报错，因为Map的key不能重复；  value不能为空，否则报空指针 |

对应报错：

|  |
| --- |
| Java  当key重复时，会抛出异常：java.lang.IllegalStateException: Duplicate key \*\*  当value为null时，会抛出异常：java.lang.NullPointerException |

当key重复时替换方案（不建议使用）：

|  |
| --- |
| Java  //下面这句会出异常java.lang.IllegalStateException: Duplicate key  Map<String, Person> map1 = list.stream().collect(Collectors.toMap(Person::getName , (p) -> p));  //修改为下面这句  Map<String, Person> map2 = list.stream().collect(Collectors.toMap(Person::getName , (p) -> p,(k,v)->v)); |

当value为null时替换方案（推荐使用）：

|  |
| --- |
| Java  //如果User.getAddress()返回结果为null，会直接报错  list.stream().collect(Collectors.toMap(User::getName,User::getAddress));  //替换方案  list.stream().collect(HashMap::new,  (m, v)->m.put(v.getName(), v.getAddress()), HashMap::putAll); |

**1.7.2 List排序注意事项**

当要比较字段里面可能存在null的时候，会报空指针异常（不建议使用）：

|  |
| --- |
| Java  list.sort(Comparator.comparing(System::getSystemName)) |

使用如下方法可将空值排在最后(建议使用)：

|  |
| --- |
| Java  list.sort(Comparator.comparing(System::getSystemName，Comparator.nullsLast(String::compareTo))) |

判断对象及字段都可能为空（推荐使用）：

|  |
| --- |
| Java  list.sort(Comparator.nullsLast(Comparator.comparing(System::getSystemName, Comparator.nullsLast(String::compareTo)))) |

**1.8 TiDB与MySQL差异**

<https://blog.csdn.net/m0_37683758/article/details/86738403>

**1.9 基本数据类型比对**

错误的比对（不能使用）

|  |
| --- |
| Java  Long i = 2000L;  Long j = 2000L;  boolean isEqual = (i == j) |

推荐使用（推荐使用）

|  |
| --- |
| Java  i.longValue() == j.longValue();  或  i.equals(j); |

**分析：**

Java 基本类型的包装类的大部分都实现了常量池技术，即Byte,Short,Integer,Long,Character；

这5种包装类默认创建了数值[-128，127]的相应类型的缓存数据，但是超出此范围仍然会去创建新的对象。

当Long型的值超过了【-128,127】的时候，会new一个新的对象，如果没有超过就会放到缓存中。

所以当Long型的值超过了【-128,127】的时候，“==”判断就不会相等了，可以理解为此时“==”比较的是两个对象的地址而不是值。

Long源码：

|  |
| --- |
| Java  public static Long valueOf(long l) {  final int offset = 128;  if (l >= -128 && l <= 127) { // will cache  return LongCache.cache[(int)l + offset];  }  return new Long(l);  } |

**1.10 logger info调试日志**

logger info上线完一定要删除，并发量大的时候，es和服务器压力很大

**1.11 表索引**

MySQL及TiDB大数据量数据查询一定考虑加索引

**1.12 redis缓存的对象类加field时，必须放在已有field后面追加，不能中间加**

redis的序列化目前使用的protobuf序列化，对类属性的顺序有要求，不可以改变field顺序，否则可能导致反序列化错乱

**1.13 使用分布式Lock的时候，tryLock一定要放到try代码块外面，unLock一定要放到finally里面**

unlock放finally是因为要保证解锁一定成功，tryLock如果放到try里面，tryLock没成功的时候，也会走到finally里面把相应key给unlock

**1.14 尽量避免往redis里面放永久不过期key**

会导致redis内存压力大，key的数量很庞大

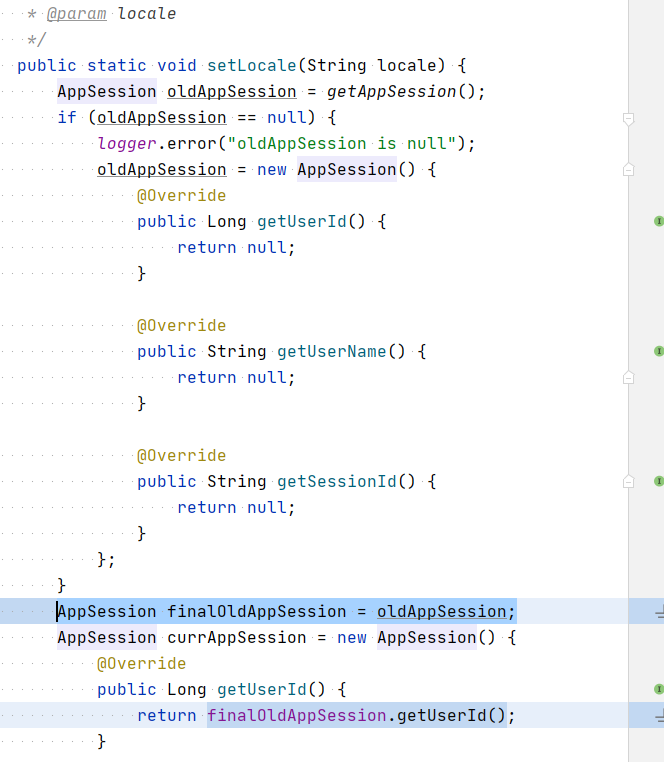
**1.15 Java 工具类**



**1.16 研发不能假定某业务的业务量不能超过某个值**

**1.17 方法中匿名内部类引用成员变量存在栈溢出风险**

如下图所示，如果单个线程重复setLocale,这样currAppSession会变成新的oldAppSession,导致引用一直被持有，因为引用句柄占用栈内存，这样就会出现栈溢出风险



可以使用下次的实现方式



**1.18 代码必须考虑多商家，多授权方及多仓库关联查询**

代码关联多表关联（或在内存中关联）查询时一定要判断是否使用merchant\_id,ep\_id及wh\_id进行关联，防止在生产多商家部署使用一套库，查窜的情况

**1.19 非高质量查询避免全表扫描的代码实现方式**

在大表查询时，如果录入的查询条件不够优秀，即使where有索引字段，mysql可能存在放弃索引而走全表扫描的情况。

针对此情，比较简单的思路是通过索引字段查询id的区间范围。能够避免全表扫描（因为有id区间一定会做主键range扫描）。

这里有几个版本的演进

版本一：不建议使用

|  |
| --- |
| SQL  select min(id) as minId, max(id) as maxId  from gi\_task  where `created\_time` >= #{createdTimeFrom} and `created\_time` &lt;= #{createdTimeTo} |

原因：虽然where条件是索引字段，但是min，max的函数，mysql会有计算逻辑，依然会将索引命中的区间做计算，性能较低。以3000W的数据量来看大概在1-3s左右

版本二：不建议使用

|  |
| --- |
| SQL  select id  from gi\_task  where  <include refid="queryWhereSql"/>  order by t.created\_time asc limit 1 |

原因：查询条件不要加非索引字段的其他字段，否则mysql会回表检索，性能较低

版本三：建议使用

|  |
| --- |
| SQL  select minId, maxId  from (  select id as minId, 'a' as 'col'  from gi\_task  where `created\_time` >= #{createdTimeFrom} and `created\_time` &lt;= #{createdTimeTo}  order by `created\_time` , id  limit 1  ) a,  (  select id as maxId, 'a' as 'col'  from gi\_task  where `created\_time` >= #{createdTimeFrom} and `created\_time` &lt;= #{createdTimeTo}  order by `created\_time` desc , id desc  limit 1  ) b  where a.col = b.col |

原理：利用mysql的索引的特性，底层是以BTree方式存储，所以直接使用索引字段的顺序能够快速定位最小与最大的主键id。

使用方式：建议两段查询，现在业务代码里查询id区间，然后作为查询条件二次查询。

1. **部属篇**

**2.1 topic未授权**

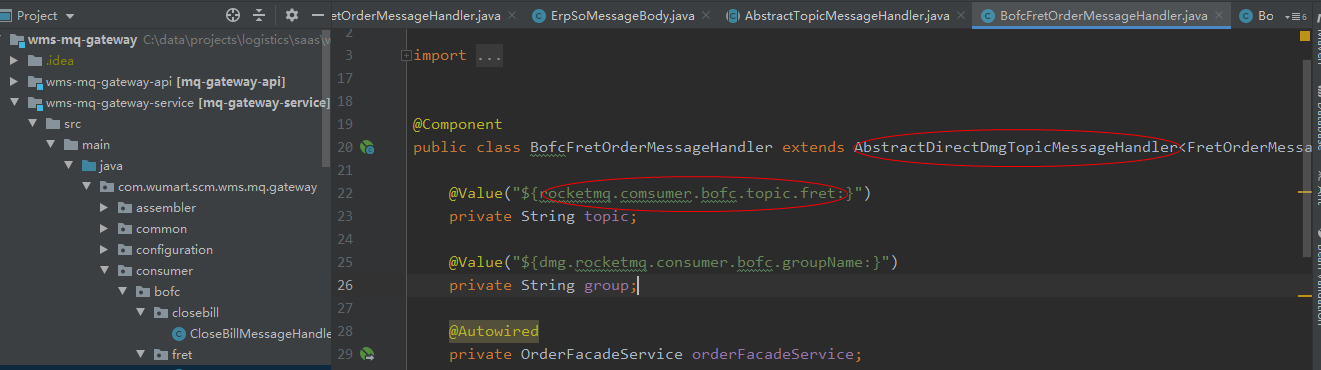
提示topic没有读写权限，请申请相关topic的读写权限

消息发送提示没有写权限，请在配置的message.queue.dmg.producer.topics中关联上相关topic

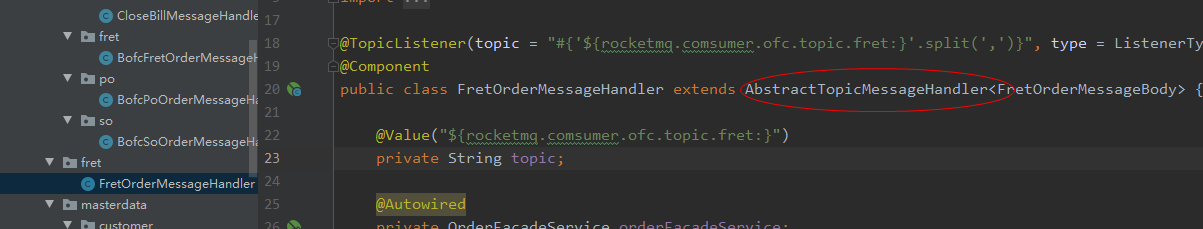
**2.2 topic重复消费**

|  |
| --- |
| CSS  [ERROR] 2021-07-10 18:44:18.177 [main] [org.springframework.boot.SpringApplication] [] - Application startup failed  com.dmall.dmg.sdk4.comm.exception.DMGException: [rkt\_rdp\_dc\_closeBill\_hk\_dev\_test], start RocketMQ consumer Exception!  at com.dmall.dmg.sdk4.rocketmq.consumer.CoreRocketConsumerImpl.start(CoreRocketConsumerImpl.java:287)  at java.util.ArrayList.forEach(ArrayList.java:1257)  at com.dmall.scm.framework.msgqueue.consumer.dmg.CoreRocketConsumerImplDelegate.init(CoreRocketConsumerImplDelegate.java:48)  at java.lang.Iterable.forEach(Iterable.java:75)  at com.dmall.scm.framework.msgqueue.consumer.dmg.DmgMessageQueueConsumer.init(DmgMessageQueueConsumer.java:35)  at com.dmall.scm.framework.msgqueue.consumer.dmg.DmgMessageQueueConsumer.onApplicationEvent(DmgMessageQueueConsumer.java:45)  at com.dmall.scm.framework.msgqueue.consumer.dmg.DmgMessageQueueConsumer.onApplicationEvent(DmgMessageQueueConsumer.java:14)  at org.springframework.context.event.SimpleApplicationEventMulticaster.doInvokeListener(SimpleApplicationEventMulticaster.java:172)  at org.springframework.context.event.SimpleApplicationEventMulticaster.invokeListener(SimpleApplicationEventMulticaster.java:165)  at org.springframework.context.event.SimpleApplicationEventMulticaster.multicastEvent(SimpleApplicationEventMulticaster.java:139)  at org.springframework.context.support.AbstractApplicationContext.publishEvent(AbstractApplicationContext.java:393)  at org.springframework.context.support.AbstractApplicationContext.publishEvent(AbstractApplicationContext.java:347)  at org.springframework.boot.context.event.EventPublishingRunListener.finished(EventPublishingRunListener.java:101)  at org.springframework.boot.SpringApplicationRunListeners.callFinishedListener(SpringApplicationRunListeners.java:79)  at org.springframework.boot.SpringApplicationRunListeners.finished(SpringApplicationRunListeners.java:72)  at org.springframework.boot.SpringApplication.run(SpringApplication.java:305)  at org.springframework.boot.SpringApplication.run(SpringApplication.java:1118)  at org.springframework.boot.SpringApplication.run(SpringApplication.java:1107)  at com.wumart.scm.wms.mq.gateway.WmsMqGatewayServiceApplication.main(WmsMqGatewayServiceApplication.java:13)  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)  at java.lang.reflect.Method.invoke(Method.java:498)  at org.springframework.boot.loader.MainMethodRunner.run(MainMethodRunner.java:48)  at org.springframework.boot.loader.Launcher.launch(Launcher.java:87)  at org.springframework.boot.loader.Launcher.launch(Launcher.java:50)  at org.springframework.boot.loader.JarLauncher.main(JarLauncher.java:51)  Caused by: org.apache.rocketmq.client.exception.MQClientException: The consumer group[group\_wms\_mq\_gateway-rkt\_rdp\_dc\_closeBill\_hk\_dev\_test] has been created before, specify another name please.  See <http://rocketmq.apache.org/docs/faq/> for further details.  at org.apache.rocketmq.client.impl.consumer.DefaultMQPushConsumerImpl.start(DefaultMQPushConsumerImpl.java:599)  at org.apache.rocketmq.client.consumer.DefaultMQPushConsumer.start(DefaultMQPushConsumer.java:456)  at com.dmall.dmg.sdk4.rocketmq.consumer.CoreRocketConsumerImpl.start(CoreRocketConsumerImpl.java:284)  ... 26 common frames omitted |

第一种消息消费实现方式：



第二种消息实现方式：



第一种只需要rocketmq.comsumer.bofc.topic.fret=rkt\_bofc\_wms\_fret\_hk\_dev\_test一个topic配置就行，不需要再在message.queue.dmg.consumers[7].subscriptions类似数组里面加这个topic，不然就会报下面错误

**2.3 自动循环重启**

考虑是否环境服务申请资源过低（内存，磁盘或CPU），考虑扩容

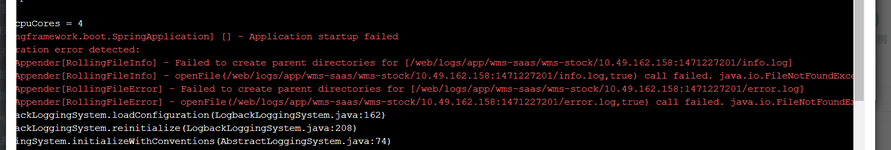


以下重启错误码及原因：

|  |
| --- |
| SQL  多点云平台 常见的容器退出状态码解释 [3]  **Exit Code 0**  退出代码0表示特定容器没有附加前台进程  该退出代码是所有其他后续退出代码的例外  这不一定意味着发生了不好的事情。  如果开发人员想要在容器完成其工作后自动停止其容器，则使用此退出代码。  比如：kubernetes job 在执行完任务后正常退出码为 0    **Exit Code 1**  程序错误，或者Dockerfile中引用不存在的文件，如 entrypoint中引用了错误的包  程序错误可以很简单，例如 “除以0”，也可以很复杂，比如空引用或者其他程序 crash    **Exit Code 137**  表明容器收到了 SIGKILL 信号，进程被杀掉，对应kill -9  引发SIGKILL的是docker kill。这可以由用户或由docker守护程序来发起，手动执行：docker kill  137 比较常见，如果 pod 中的limit 资源设置较小，  会运行内存不足导致 OOMKilled，此时state 中的 ”OOMKilled” 值为true，  你可以在系统的 dmesg -T 中看到 oom 日志    **Exit Code 139**  表明容器收到了 SIGSEGV 信号，无效的内存引用，对应kill -11  一般是代码有问题，或者 docker 的基础镜像有问题    **Exit Code 143**  表明容器收到了 SIGTERM 信号，终端关闭，对应kill -15  一般对应 docker stop 命令  有时docker stop也会导致Exit Code 137。  发生在与代码无法处理 SIGTERM 的情况下，docker进程等待十秒钟然后发出 SIGKILL 强制退出。 |

**2.4 DF AMP两个节点无法正常启动问题**

发布时出现以下问题：



解决方案：



**2.5 内网域名配置为http而不是https**

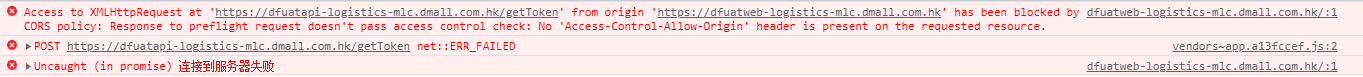
**2.6 Redis配置错误**

|  |
| --- |
| Plain Text  [ERROR] 2021-08-07 11:23:59.059 [main] [org.springframework.boot.SpringApplication] [] - Application startup failed  org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'crossDockingDistributionService': Unsatisfied dependency expressed through field 'crossDockingRelationService'; nested exception is org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'crossDockingRelationService': Unsatisfied dependency expressed through field 'delayQueue'; nested exception is org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'redissonDelayQueue' defined in class path resource [com/wumart/scm/framework/spring/boot/cachestarter/configuration/queue/DistributionQueueAutoConfiguration.class]: Unsatisfied dependency expressed through method 'redissonDelayQueue' parameter 0; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'redissonClient' defined in class path resource [com/wumart/scm/framework/spring/boot/cachestarter/configuration/redisson/RedissonAutoConfiguration.class]: Bean instantiation via factory method failed; nested exception is org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.redisson.api.RedissonClient]: Factory method 'redissonClient' threw exception; nested exception is org.redisson.client.RedisConnectionException: Unable to connect to Redis server: redis6387.test.inner-dmall.com.hk/10.40.2.22:6387  at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$AutowiredFieldElement.inject(AutowiredAnnotationBeanPostProcessor.java:586)  at org.springframework.beans.factory.annotation.InjectionMetadata.inject(InjectionMetadata.java:87)    Caused by: org.springframework.beans.factory.UnsatisfiedDependencyException: Error creating bean with name 'redissonDelayQueue' defined in class path resource [com/wumart/scm/framework/spring/boot/cachestarter/configuration/queue/DistributionQueueAutoConfiguration.class]: Unsatisfied dependency expressed through method 'redissonDelayQueue' parameter 0; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'redissonClient' defined in class path resource [com/wumart/scm/framework/spring/boot/cachestarter/configuration/redisson/RedissonAutoConfiguration.class]: Bean instantiation via factory method failed; nested exception is org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.redisson.api.RedissonClient]: Factory method 'redissonClient' threw exception; nested exception is org.redisson.client.RedisConnectionException: Unable to connect to Redis server: redis6387.test.inner-dmall.com.hk/10.40.2.22:6387  at org.springframework.beans.factory.support.ConstructorResolver.createArgumentArray(ConstructorResolver.java:749)    Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'redissonClient' defined in class path resource [com/wumart/scm/framework/spring/boot/cachestarter/configuration/redisson/RedissonAutoConfiguration.class]: Bean instantiation via factory method failed; nested exception is org.springframework.beans.BeanInstantiationException: Failed to instantiate [org.redisson.api.RedissonClient]: Factory method 'redissonClient' threw exception; nested exception is org.redisson.client.RedisConnectionException: Unable to connect to Redis server: redis6387.test.inner-dmall.com.hk/10.40.2.22:6387  at org.springframework.beans.factory.support.ConstructorResolver.instantiateUsingFactoryMethod(ConstructorResolver.java:599) |

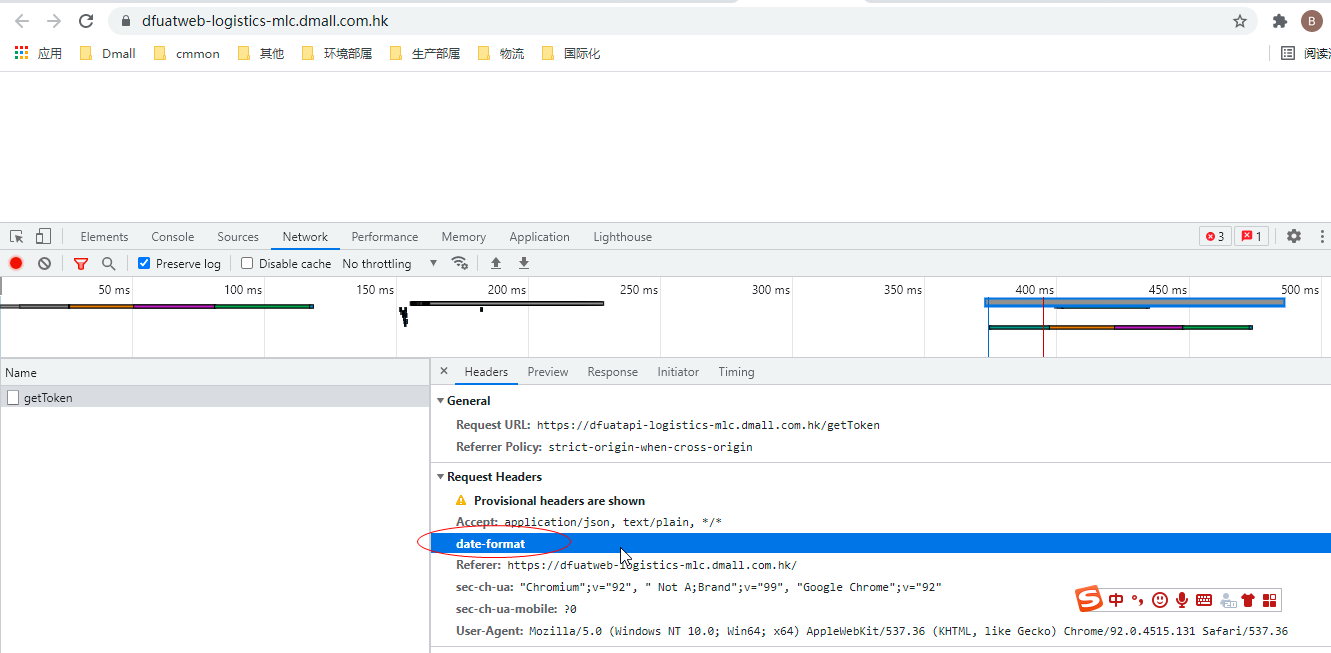
以上Redis配置错误，请检查host，port及密码中是否正确或带有空格

**2.7 页面请求跨域问题**

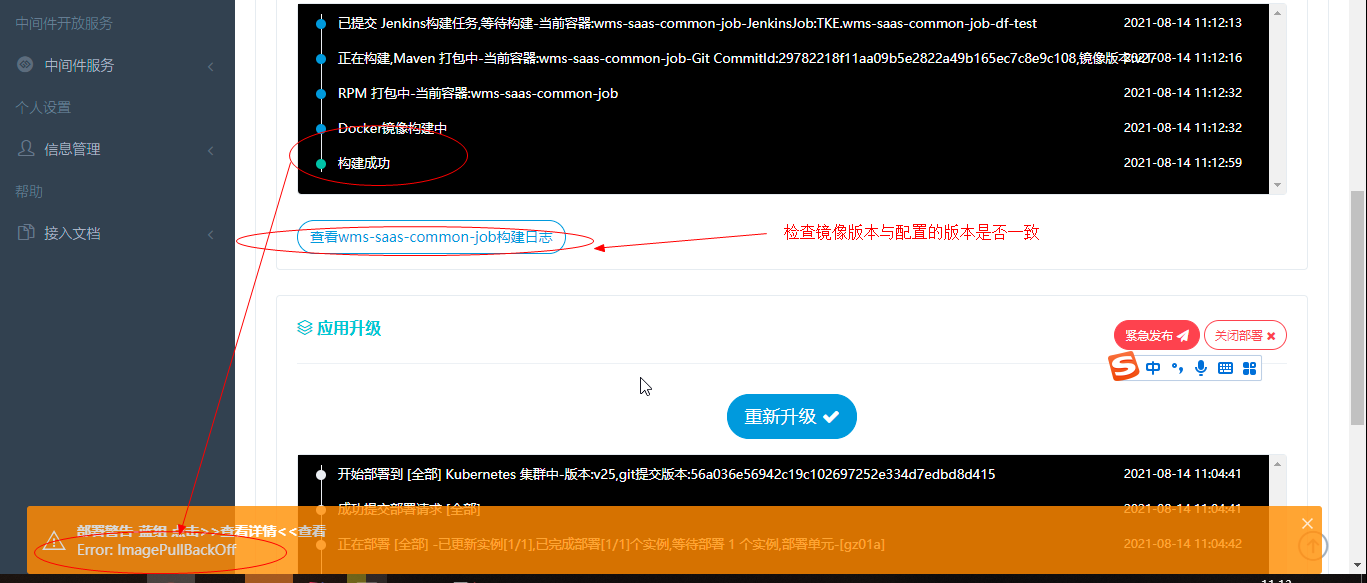
PC端header不能添加额外字段，不然会引起跨域。



PC请求时header多加了一个“date\_format”



**2.7 无法拉取镜像**



**2.8 线上各服务全部需要JVM调优**

Dmall物流香港万宁和柬埔寨LuckyWMS项目生产环境部署资源汇总

**2.9 多数据源启动报错**

|  |
| --- |
| CSS  [ERROR] 2021-11-23 18:45:55.058 [main] [LoggingFailureAnalysisReporter.report:42] [] [10.39.68.215] -    \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  APPLICATION FAILED TO START  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*    Description:    Parameter 0 of method setSqlSessionFactory in com.wumart.scm.framework.core.arch.dao.mysql.AbstractMySqlSessionDao required a single bean, but 4 were found:  - Null-SqlSessionFactory: defined in null  - dmall\_logistics-SqlSessionFactory: defined in null  - wms\_gi-SqlSessionFactory: defined in null  - logistics\_data-SqlSessionFactory: defined in null      Action:    Consider marking one of the beans as @Primary, updating the consumer to accept multiple beans, or using @Qualifier to identify the bean that should be consumed |

检查配置指定事务处理主数据库

|  |
| --- |
| CSS  complex.ds.multi.logistics\_data.primary=true |



com.wumart.scm.spring.boot.mysql.starter.complex.ComplexDataSourceRegistry

**2.10 接入网关错误码列表**

Lightning网关中定义了很多错误码，错误码以6XX开始，以下是对错误码的定义。方便查阅。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Code | Error Message | Cause | Happens On | Component | Status |
| 601 | 与下游的连接已断开 | 与下游的连接被断开 | 向下游发送数据的请求过程中 | Netty | 已启用 |
| 602 | * 当前URL被列入黑名单,禁止访问 * 当前URL未被列入白名单,禁止访问 | 客户端IP已经被列入黑名单或者不在白名单中 | 接收到上游的请求并且通过Plugin校验上游信息时 | Tomcat | 已启用 |
| 603 | 派发请求到下游超时 | 下游长时间没有返回响应 | 等待下游返回Response的过程中 | Netty | 已启用 |
| 604 | 超过流量阈值 | 用户设置了流量限制，并且并发的请求数超过阈值 | 接收到上游的请求并且通过Plugin校验流量信息时 | Tomcat | 已启用 |
| 605 | 未找到任何业务实例 | 根据域名在缓存中没有找到对应的下游节点信息 | 根据域名获取下游节点信息时 | Tomcat | 已启用 |
| 606 | 所有业务实例都被熔断 | 所有下游节点都被熔断 | 根据域名获取下游节点信息时 | Tomcat | 已启用 |
| 607 | 业务连接池已满 | 创建连接时连接数已经达到单节点的上限 | 与下游创建连接的过程中 | Netty | 已启用 |
| 608 | * 当前重试N次后,仍未找到可用的业务实例 * 最大重试N次后,仍未找到可用的业务实例 | 选取下游节点的次数达到重试次数的阈值 | 根据域名获取下游节点信息时 | Tomcat/Netty | 已启用 |
| 609 | 网关积压请求超过阈值,触发熔断 | 网关中积压了大量的未完成的请求， 请求的并发数达到阈值 | 接收到上游的请求并且通过Plugin校验流量信息时 | Tomcat | 已启用 |
| 610 |  |  |  |  | 未启用 |
| 611 |  |  |  |  | 未启用 |
| 612 | 派发请求过程中下游连接被断开 | 与下游的连接存活时间过长 | 轮询所有连接，校验存活时间过长的连接 | Task | 已启用 |
| 613 | 下游连接已断开 | 与下游的连接被断开 | 等待下游返回响应的过程中 | Netty | 已启用 |
| 614 |  |  |  |  | 未启用 |
| 615 | 请求未正确完成 | 与下游的连接存活时间过长 | Netty ChannelAliveTimeoutHandler 被触发时 | Netty | 已启用 |
| 616 | 无法指定路由单元 | 未正确配置跨单元转发规则（注意：需要在每个单元都配置）  **现阶段为了兼容性，不配置规则的情况下当做GZ处理** | 接收到上游请求时 | Tomcat | 已启用 |
| 617 | 禁止多次跨单元代理 | 跨单元转发规则不一致，导致流量会在不同单元间多次转发 | 接收到上游请求时 | Tomcat | 已启用 |
| 699 | * 处理下游回复异常 * 派发请求到下游异常 * 获取并校验EndPoints阶段失败 * 获取Channel阶段失败 * 重新获取Channel失败 * Handler流程组装阶段失败 * 读取上游body数据失败 * 读取请求内容失败 * 派发请求到下游异常 * Plugin流程处理阶段失败 | 未捕获的异常 | 以上任何阶段有未捕获的异常时 | Tomcat/Netty | 已启用 |

**2.11 MQTT不弹窗问题**

设置大一点数量：mqtt.max-inflight = 5000