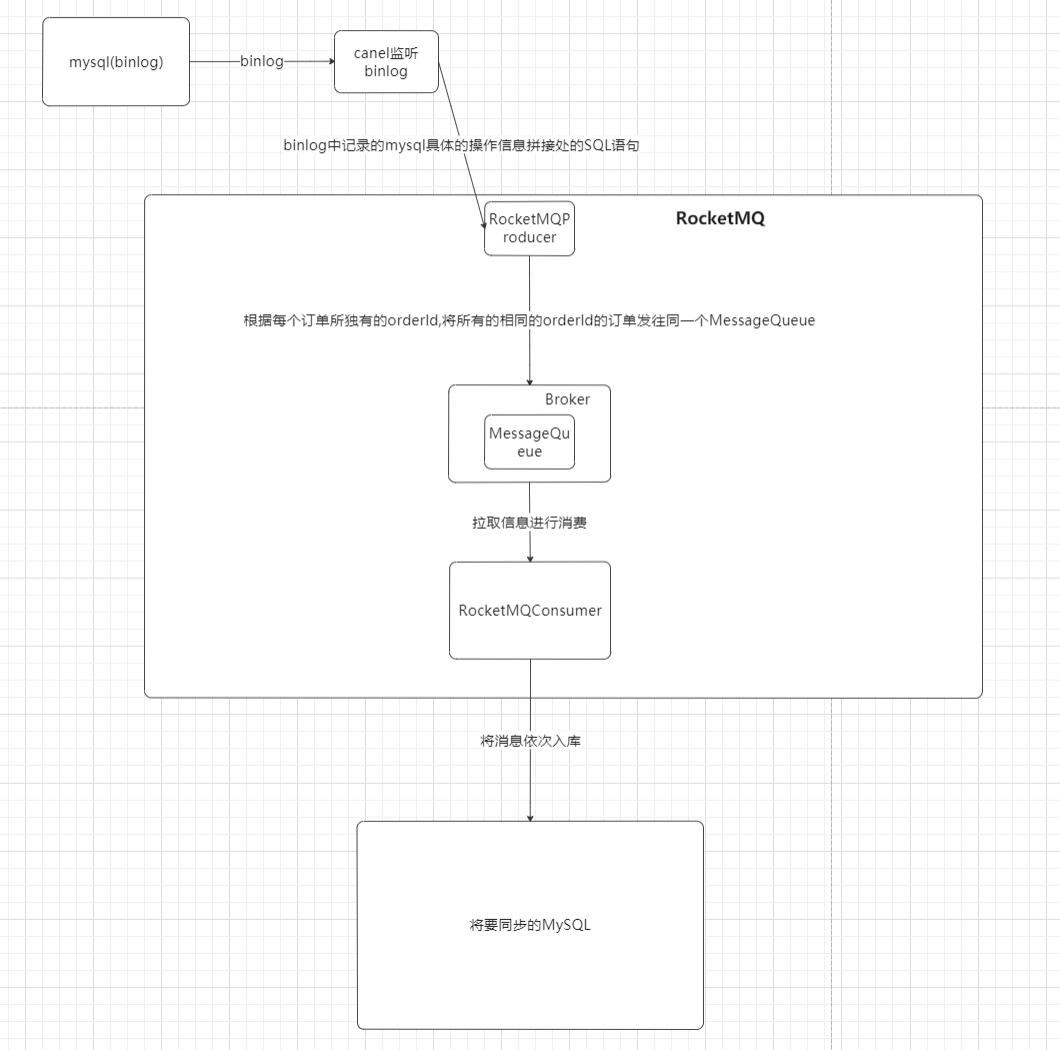
# 项目说明

本项目主要是为了解决大数据组所需要的业务数据。由于大数据项目组要访问数据库时，会进行大SQL查询，如果直接对业务数据库进行查询，会造成业务数据库的压力过大。所以我们采取的是用Canel对数据库进行同步。Canel监听MYSQL的Binlog对于直接访问数据库来说，开销相对较小。

项目中主要涉及到的有：Mysql、Canel、RocketMQ

# 项目的流程



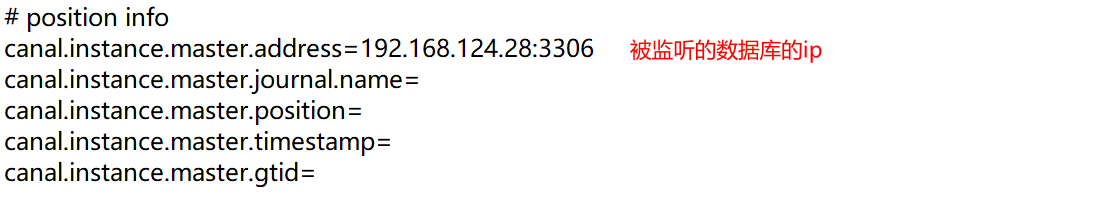
# 具体的操作

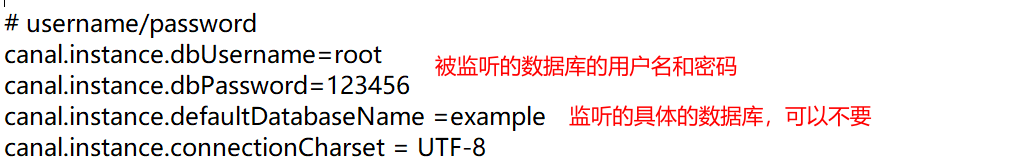
## 3.1部署Canel

具体操作 参考链接：<https://www.cnblogs.com/java-spring/p/8930740.html>

<https://www.cnblogs.com/javazhiyin/p/10836846.html>

这里主要说明一下主要应该修改的部分：/usr/local/java/canel/conf/example/ instance.properties





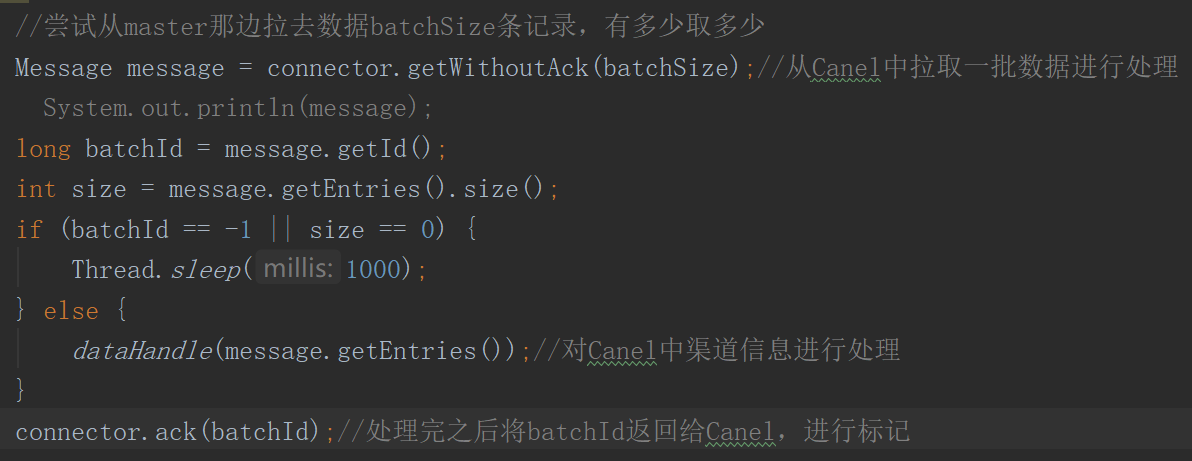
两个重要的log

/usr/local/java/canel/logs/canal/canel.log 记录canel的工作情况

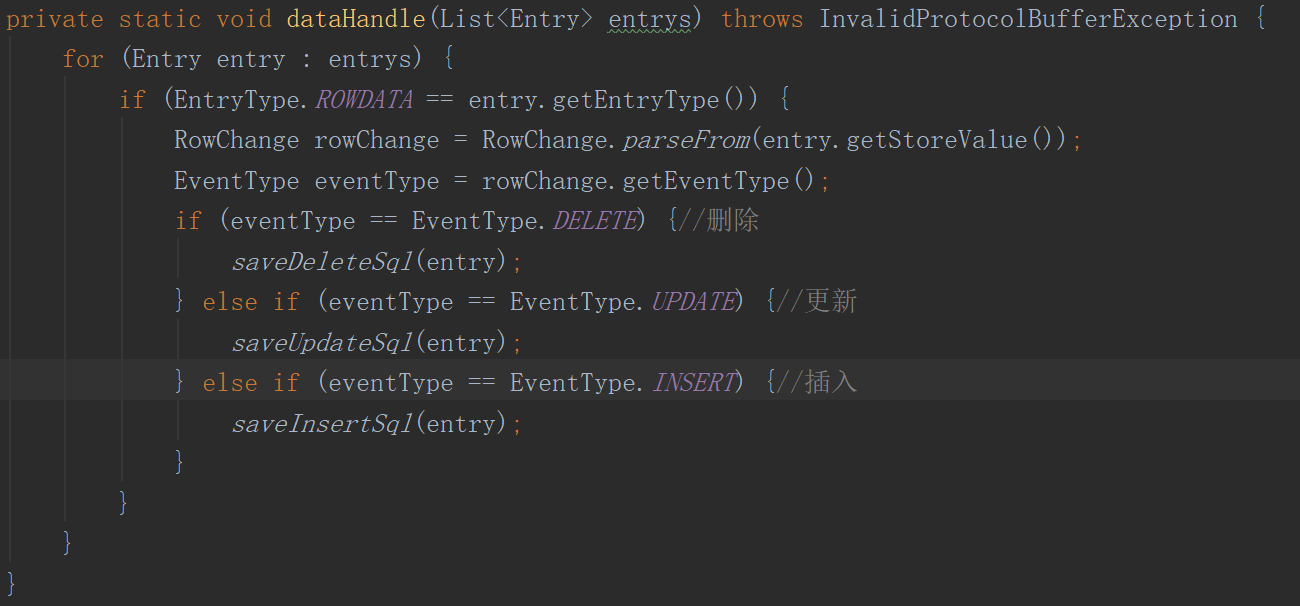
/usr/local/java/canel/logs/example/example.log 记录监听mysql日志

## 3.2CanelClient

CanelClient用于读取Canel中的信息，并且进行进一步的加工



具体的操作



在具体法人操作中将拼接好的SQL语句发送到RocketMQ中



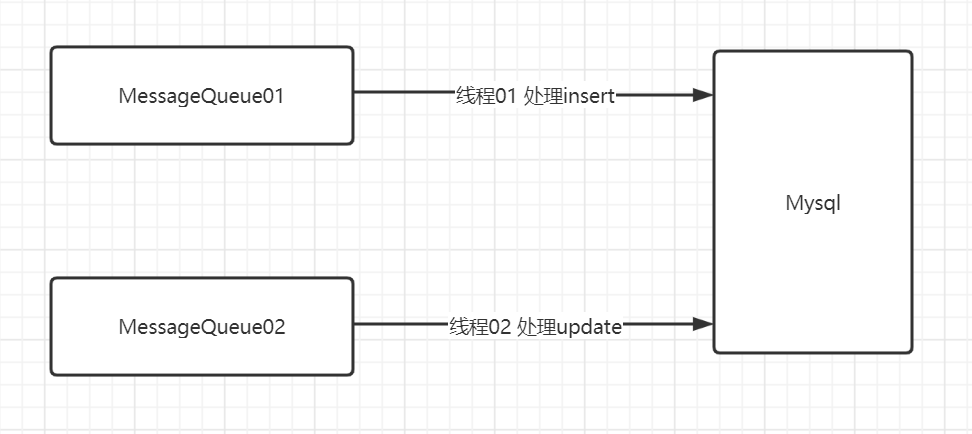
### 3.2.1 数据库同步的消息乱序问题？

这里有一个问题，就是如果直接不进行任何特殊处理的就直接将小心发送到RocketMQ中，消息就会随机均匀的发送到Broker中的MessageQueue中，Broker中可能有多个MessageQueue。如果消费者集群开启多线程处理消息的话，就会出现乱序问题。比如

有两个SQL语句：

insert into example.user (id,orderId,username,password,age) VALUES ('8','8','8','8','8');

update example.user set id = '8', orderId = '8', username = '8', password = '80', age = '80' where id=8;



如果两个线程分别对信息进行处理，线程02先进行update，此时MYSQL中还没有该条数据，所以该操作不会对数据库中产生影响。之后，线程01开始处理，插入一条数据。最终的结果是

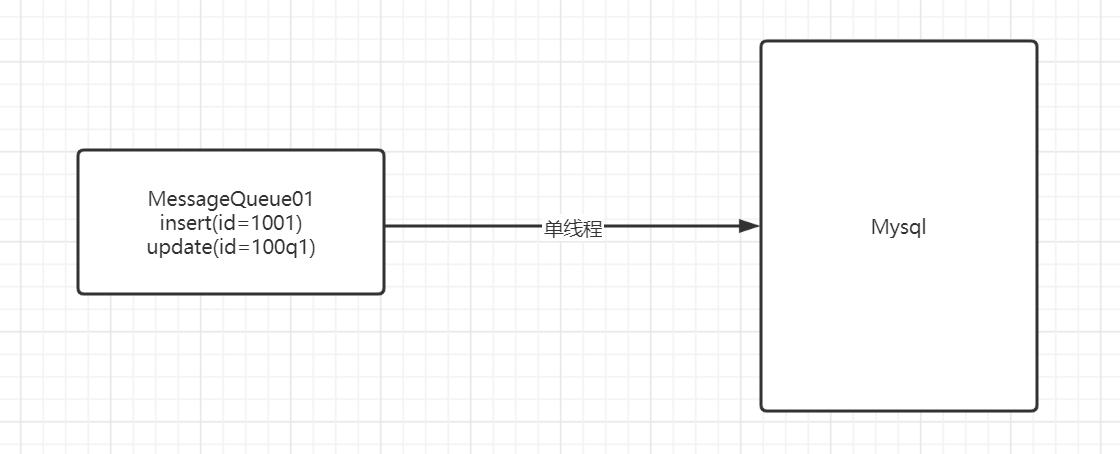


而正确的执行顺序的结果应该是



### 3.2.2解决问题的策略

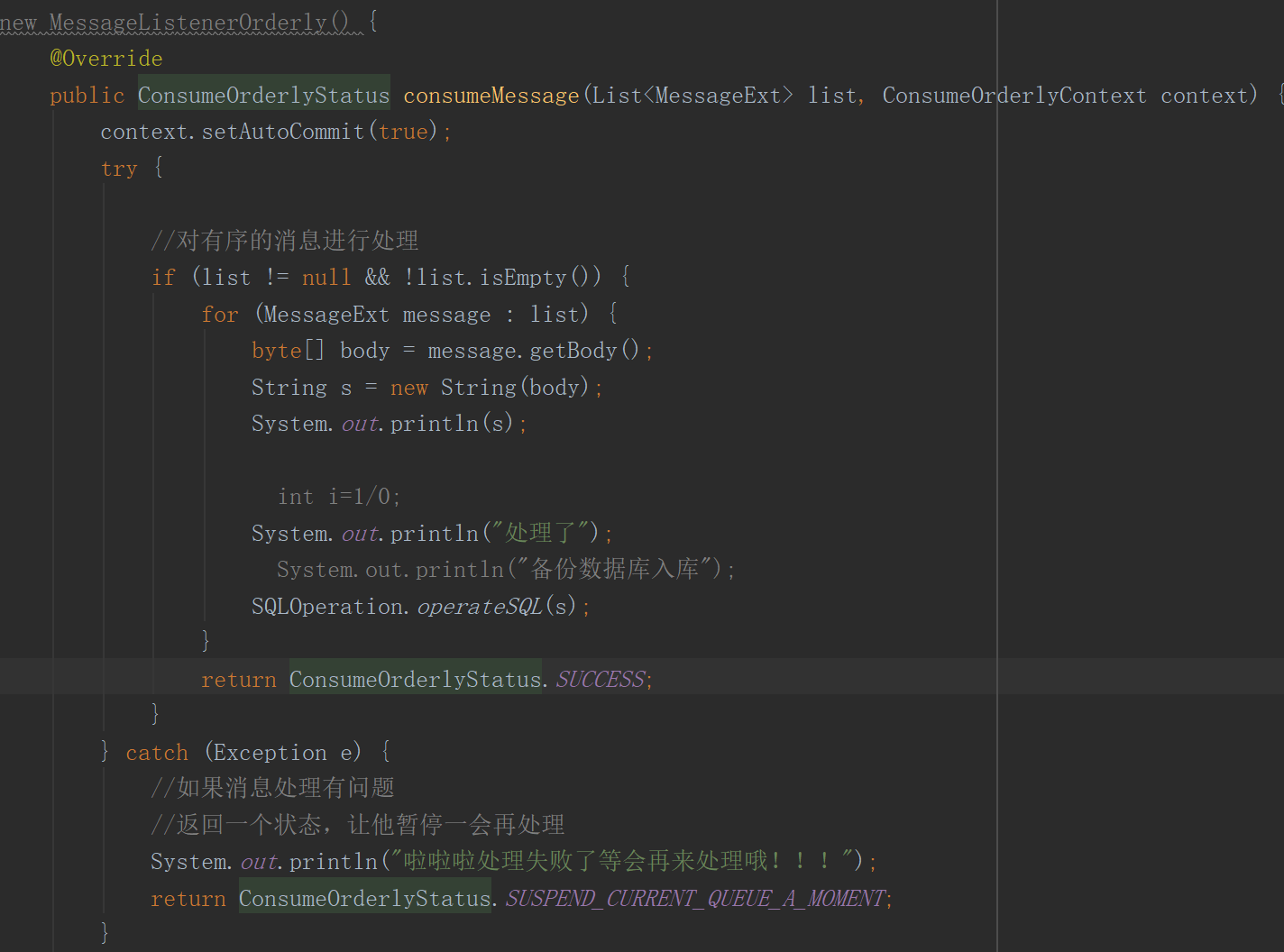
将同一个订单的binlog进入同一MessageQueue中，并且使用单线程来处理消息



Producer中的实现策略，主要实在send中实现一个对于MessageQueue的选择器



Consumer中的策略实现：在监听器中实现MessageListenerOrderly



### 3.2.3这就完了吗？没有，万一消息处理失败了可以走重试队列吗？

那么大家觉得这样就完了吗？

绝对不是，我之前给大家讲过，在Consumer处理消息的时候，可能会因为底层存储挂了导致消息处理失败，之前我们说过，此时可以返回RECONSUME\_LATER状态，然后broker会过一会儿自动给我们重试。

但是这个方案用在我们的有序消息中可以吗？

那绝对是不行的，因为如果你的consumer获取到订单的一个insert binlog，结果处理失败了，此时返回了RECONSUMELATER，那么这条消息会进入重试队列，过一会儿才会交给你重试。

但是此时broker会直接把下一条消息，也就是这个订单的update binlog交给你来处理，此时万一你执行成功了，就根本没有数据可以更新！又会出现消息乱序的问题。

在消息处理出现异常的时候不是返回RECONSUME\_LATERY(会将消息丢入重试队列)，而是返回

ConsumeOrderlyStatus.SUSPEND\_CURRENT\_QUEUE\_A\_MOMENT（会等一会在来处理这批消息）