# RocketMq特性详解

rocketmq-console:

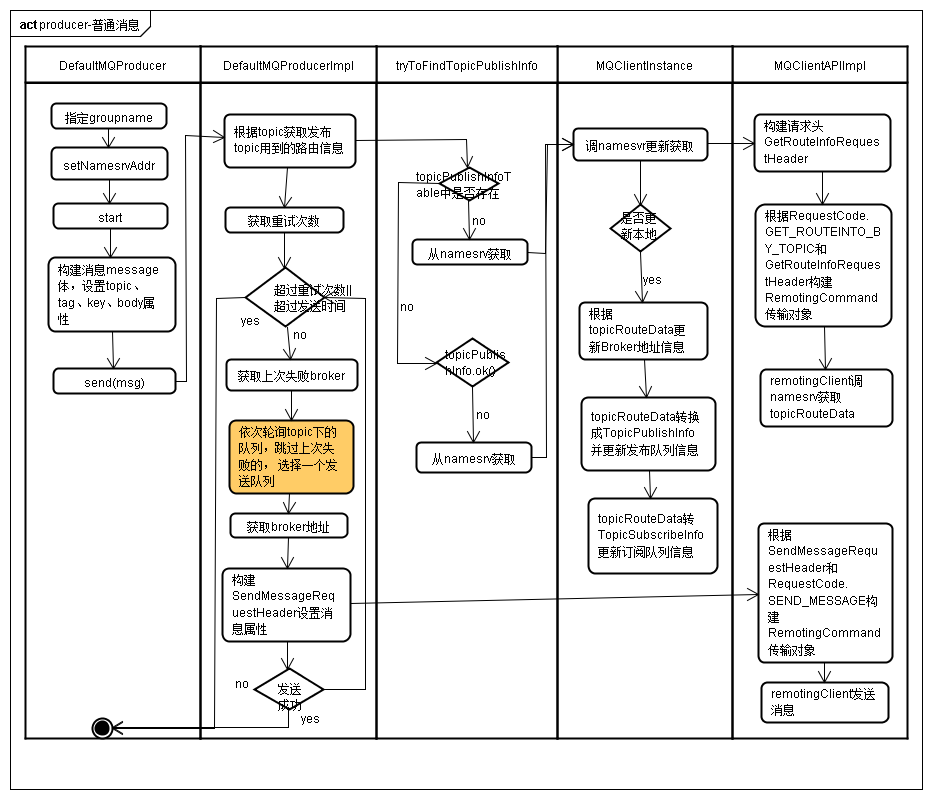
<https://github.com/apache/rocketmq-externals>

mvn package -DskipTests

java -jar target/rocketmq-console-ng-1.0.0.jar --rocketmq.config.namesrvAddr=192.168.0.12:9876;192.168.0.13:9876 --server.port=8888

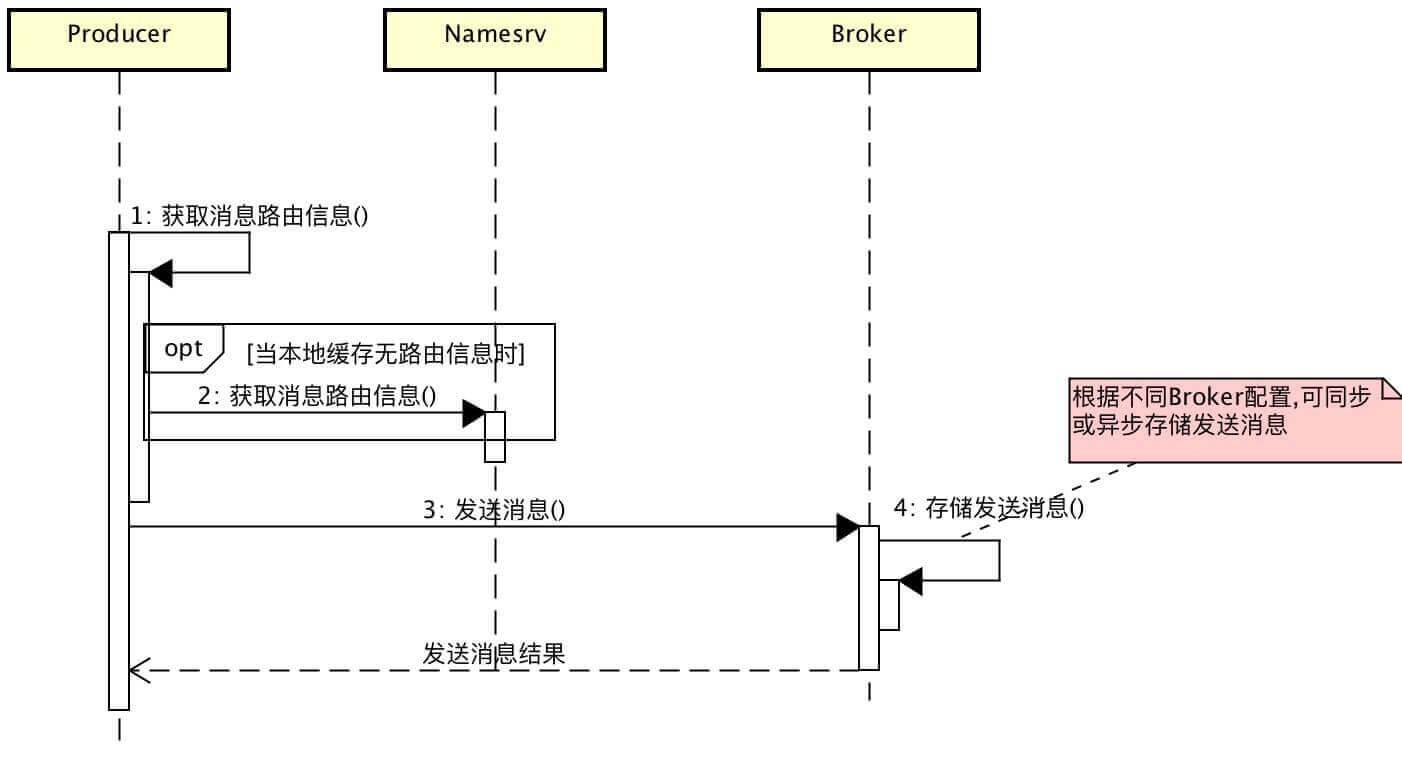
# Producer详解

## 普通消息：



org.apache.rocketmq.client.impl.CommunicationMode

org.apache.rocketmq.client.impl.producer.DefaultMQProducerImpl#sendDefaultImpl



## 定时消息

定时消息是指消息发到 Broker 后，不能立刻被 Consumer 消费，要到特定的时间点或者等待特定的时间后才能被消费。

固定精度：

**1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m 10m 20m 30m 1h 2h**

org.apache.rocketmq.store.config.MessageStoreConfig#messageDelayLevel

## 顺序消息：

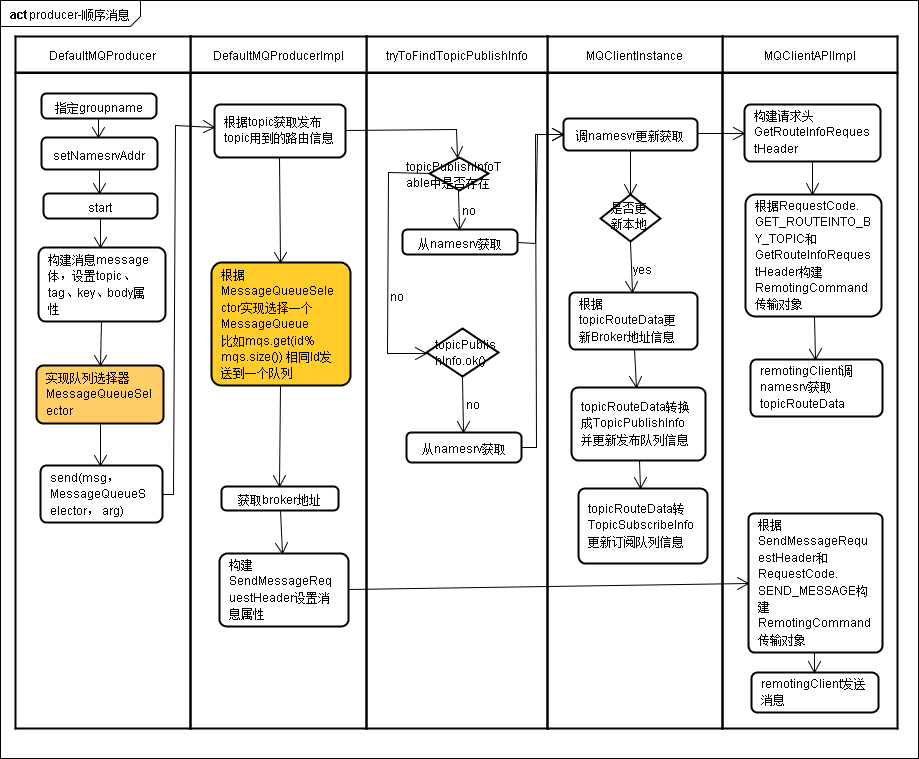
org.apache.rocketmq.client.impl.producer.DefaultMQProducerImpl#sendSelectImpl

在分布式队列实现排序，这个需求怎么做？

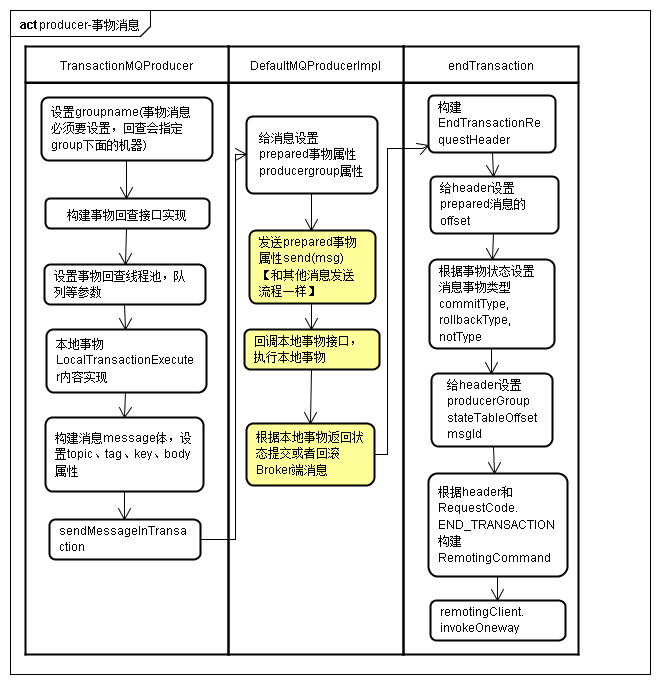
排序 分布式？

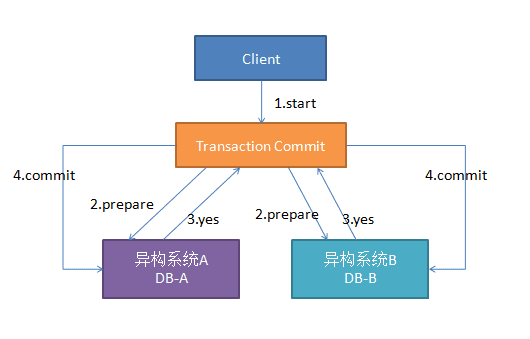
优先级：订单号20180206（queu1 ）订单号20180207(queue2)

排序+优先级（分片）



## 事物消息





二阶段提交

# Consumer详解

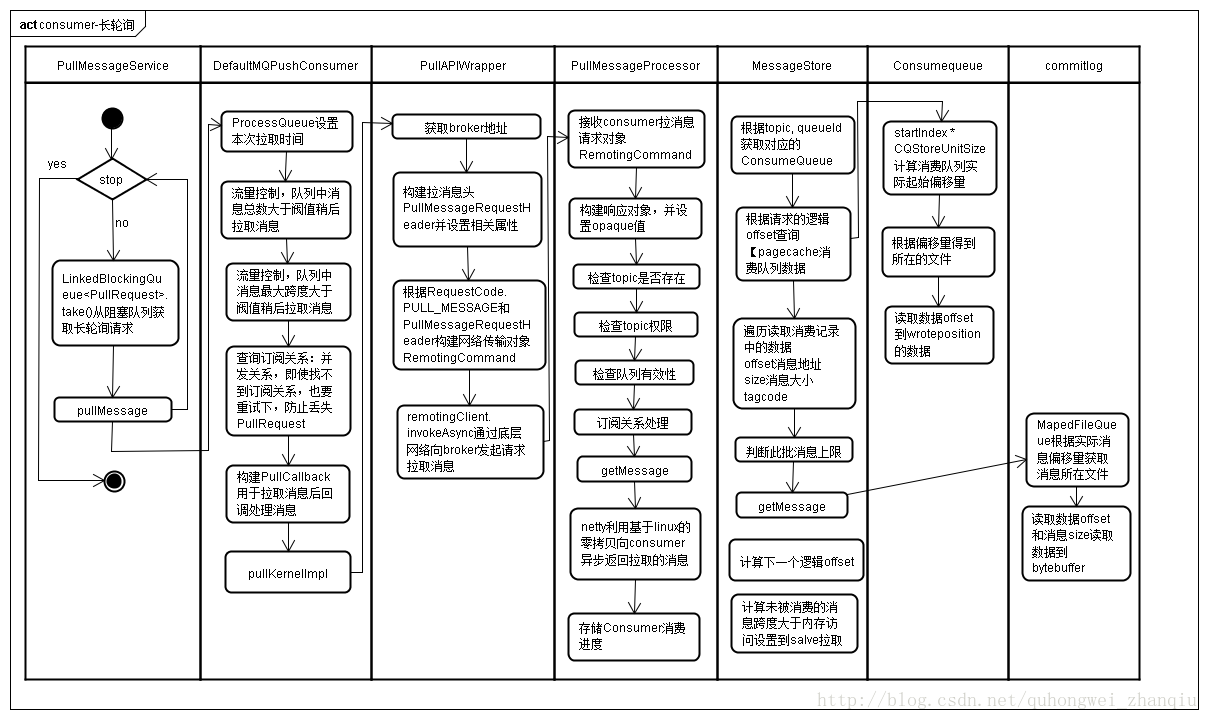
## Pull&push模式：

DefaultMQPullConsumer

DefaultMQPushConsumer

都是采用pull的方式进行消费，会有延迟，减轻服务器压力，短时间轮询可忽略延迟

长轮询：



## 消费模型：

org.apache.rocketmq.common.protocol.heartbeat.MessageModel#BROADCASTING

org.apache.rocketmq.common.protocol.heartbeat.MessageModel#CLUSTERING

## 消费选择：

org.apache.rocketmq.common.consumer.ConsumeFromWhere#CONSUME\_FROM\_LAST\_OFFSET

org.apache.rocketmq.common.consumer.ConsumeFromWhere#CONSUME\_FROM\_FIRST\_OFFSET

org.apache.rocketmq.common.consumer.ConsumeFromWhere#CONSUME\_FROM\_TIMESTAMP

## 重试策略

Consumer 消费消息失败后，要提供一种重试机制，令消息再消费一次。Consumer 消费消息失败

## 消息重复幂等：

RocketMQ无法避免消息重复，所以如果业务对消费重复非常敏感，务必要在业务层面去重

## 消息回溯：

回溯消费是指Consumer已经消费成功的消息，由于业务上需求需要重新消费，要支持此功能。

RocketMQ支持按照时间回溯消费，时间维度精确到毫秒，可以向前回溯，也可以向后回溯。

使用命令：sh mqadmin resetOffsetByTime -g xx -s yy -t tt

## Producer|consumer Group:

1、可以通过运维工具查询这个组下有多少Producer实例

2、可通过运维工具查询这个组下的消费进度,多少个Consumer实例

3、事务消息，如果Producer意外宕机，Broker会主动回调Producer Group中的任意一台机器确认事务状态

4、集群模式，一个Consumer Group下的多个Consumer均摊消费消息；广播模式，group无意义