## Jmq机制

[京东消息中间件JMQ](https://blog.csdn.net/javahongxi/article/details/54411464)

Jmq用户手册<http://jpcloud.jd.com/pages/viewpage.action?pageId=18909155>

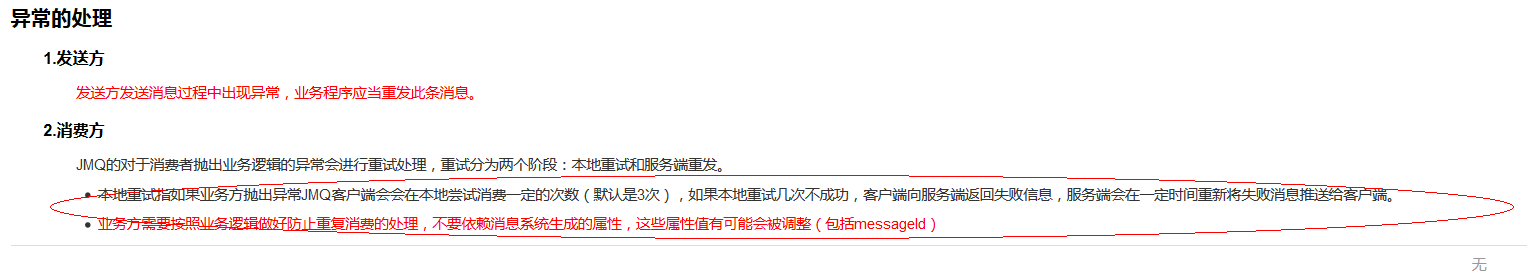
注意：在上线前进行线上的jmq订阅申请（冬哥审批），若提前订阅了，会导致接收到消息不能消费积压，会发送报警信息，只能取消订阅。测试环境不需要审批。

只要订阅，消费端就会去主动拉去该topic的消息，拉取后不能消费就会积压，将ip限制消费后也一样

[JMQ配置消费策略](http://jpcloud.jd.com/pages/viewpage.action?pageId=15355261)

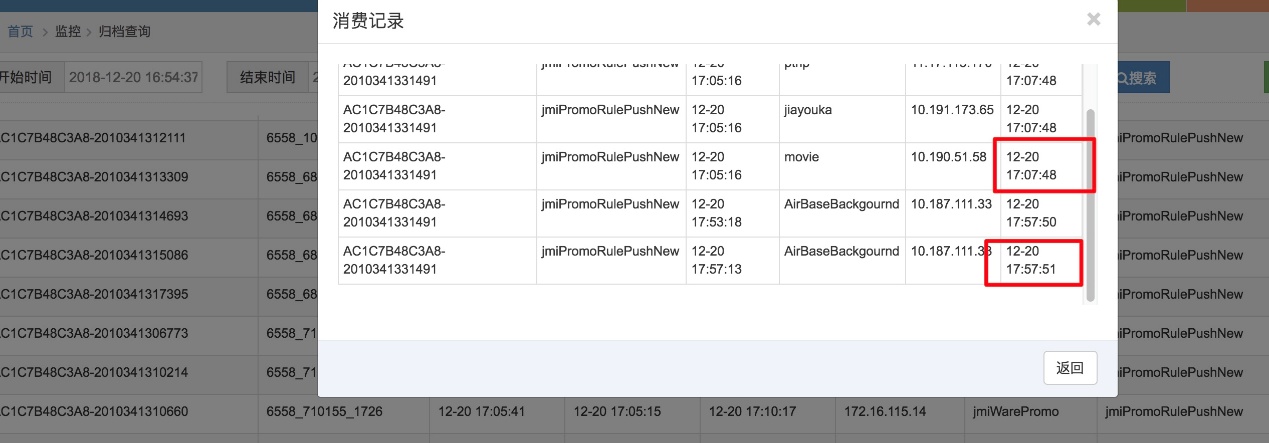
[JMQ配置重试策略](http://jpcloud.jd.com/pages/viewpage.action?pageId=18914245)





一条消息只能在一个broker的一个队列上，一个主题下面的多个消息分散在多个broker里。一个broker上的一个主题的消息会有多个队列，一个队列同一时刻只能被一个线程拉取，加锁，接收到ack后，才会释放锁。超时不会返回ack（超时会重传），异常和正常消费都会返回ack，异常会将消息加入到重试数据库，再次拉取是相当于从不同分组上拉。队列数等于队列数乘以b数量

重拉：



Jmq的阅读文档：

<http://tigcms.jd.com/>

<http://jpcloud.jd.com/display/cloud/JMQ>

这个消息的接收时间



这个消息在background中的处理时间



可能这段时间有积压，导致机器拉取消息晚

消费方的机器消费不及时会造成积压，那么拉取消息就会晚

## 消费配置：



在代码中配置了topic后，启动项目（本地或测试都可以）。

Jmq的测试环境：<http://192.168.179.67/jmq-web/login.html>

在测试环境接入申请主题后，需要开启消费

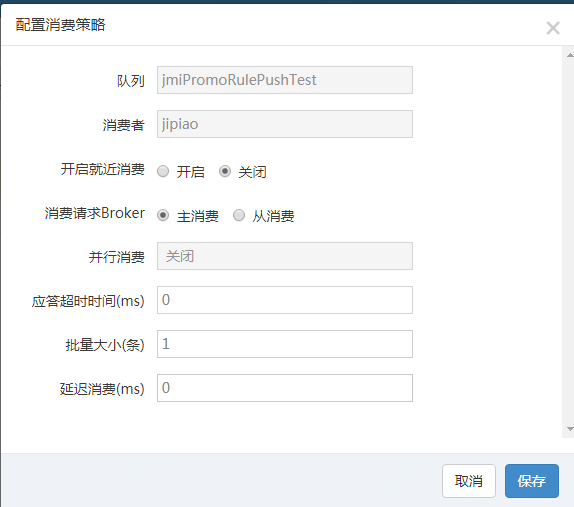


连接数：主Broker数\*客户端的ip数

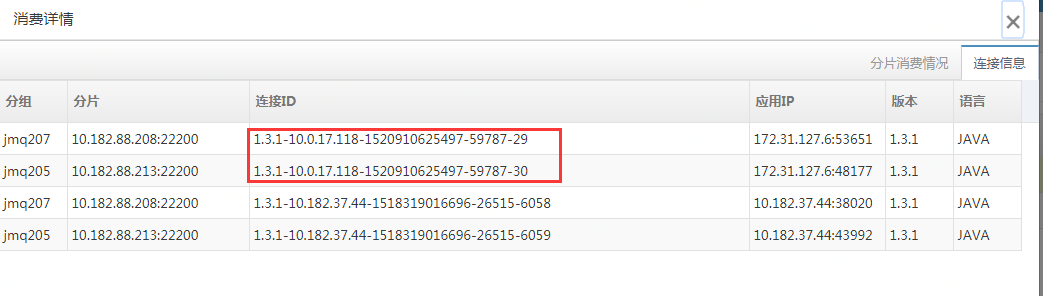
出队数：单个broker上该主题的队列数。

Broker：JMQ的服务端实例

分组：一个分组包含两个broker， 一主一从。主从节点消息完全一致，页面只展示主节点。 分组名一般为jmq\*\*\*

配置消费策略、重试策略等  






## 生产接入与配置

