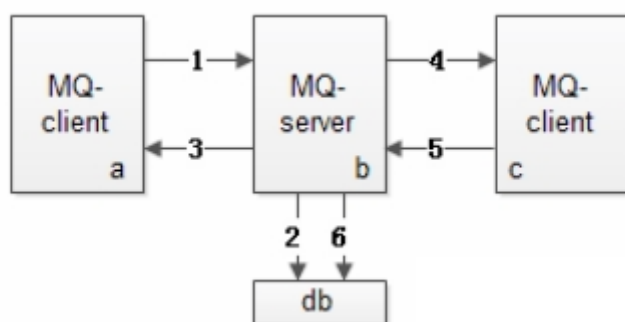


1. MQ消息发送：



1. 发送端MQ-client（消息生产者：Producer）将消息发送给MQ-server
2. MQ-server将消息落地
3. MQ-server回ACK给MQ-client（producer）
4. MQ-server将消息发送给消息接收端MQ-client（消息消费者：Customer）
5. MQ-client（Customer）消费接收到消息后发送ACK给MQ-server
6. MQ-server将落地消息删除。

2. 消息重复发送的原因

为了保证消息必达，MQ使用了消息超时、重传、确认机制。使得消息可能被重复发送，如上图，由于网络不可达原因：3和5中断，可能导致消息重发。消息生产者a收不到MQ-server的ACK，重复向MQ-server发送消息。MQ-server收不到消息消费者b的ACK，重复向消息消费者b发送消息。

3. 消息重复发送产生的后果

举个例子：购买会员卡，上游支付系统负责给用户扣款，下游系统负责给用户发卡，通过MQ异步通知。不管是上半场的ACK丢失，导致MQ收到重复的消息，还是下半场的ACK丢失，导致购卡系统收到重复的购卡通知，都可能出现，上游扣了一次钱，下游发了多张卡。

4. MQ内部如何做到幂等性

对于每条消息，MQ内部生成一个全局唯一、与业务无关的消息ID：inner-msg-id。当MQ-server接收到消息时，先根据inner-msg-id判断消息是否重复发送，再决定是否将消息落地到DB中。这样，有了这个inner-msg-id作为去重的依据就能保证一条消息只能一次落地到DB。

5. 消息消费者应当如何做到幂等性

1. 对于非幂等性业务且要求实现幂等性业务：生成一个唯一ID标记每一条消息，将消息处理成功和去重日志通过事物的形式写入到去重表中。

2. 对于非幂等性业务可不实现幂等性的业务：权衡去重所花的代价决定是否需要实现幂等性，如：购买会员卡成功，向用户发送通知短信，发送一次或者多次影响不大。不做幂等性可以省掉写去重日志的操作。