



链滴

消息队列 MQ

作者: [lpp](#)

原文链接: <https://ld246.com/article/1634718324952>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

MQ

使用场景

使用场景

使用场景：解耦合。订单后通知库存。不加入 rabbitmq 时，库存接口访问失败后，导致减库失败，数据不一致。加入后请求后放入 mq 即可，无需等待，rabbitmq 按照顺序请求，某请求第一次失败后会继续排队处理。可以设置容错机制。达到解耦合和容错的机制。

使用场景：异步提高效率，比如 A 系统调 B 系统接口，调完后需要发邮件和短信进行知用户。不加前，需要并行或串行的执行每一步。但是加 mq 后，可以异步进行发送，把需要做的事丢到 mq 中即可，提高异步的效率

可以通过消息队列长度控制请求量；可以缓存短时间内的高并发请求。减少高峰期对服务器压力。

如何保证 RabbitMQ 的消息可靠性，防止消息丢失？

MQ 宕机导致丢失消息

消息持久化：MQ 会将消息持久化磁盘，宕机重启可以恢复消息。

镜像集群：主从备份。

网络故障导致丢失消息

发送者确认模式

消费者丢失消息

消费者确认机制：在处理消息结束后，手动 Acknowledge 回执给 MQ；

MQ 未接受到 Acknowledge 会认为消费失败，消息会保留在 MQ；

如何防止 MQ 消息的重复消费？

原因：网络故障导致生产者确认机制失败，MQ 重新投递消息；

解决思路：保证处理消息接口的幂等性（幂等性指的是多次操作，结果是一致的）。

例如：你有个系统，消费一条消息就往数据库里插入一条数据，要是你一条消息重复次，你不就插入了两条，这数据不就错了？但是你要是消费到第二次的时候，自己判断一下是否已经费过了，若是就直接扔了，这样不就保留了一条数据，从而保证了数据的正确性。

解决消息幂等性：

唯一索引：保证插入的数据只存在一条；

token：每次接口请求前先获取一个 token，在下次请求时，在请求头中加上这个 token，后台进行校验，检验通过删除 token，下次请求再次判断 token；

先查询后判断：首先查询数据库是否存在数据，若存在则表示数据已请求，直接拒绝该求，若不存在，说明是第一次请求，直接放行。

如何确保消息正确地发送至 RabbitMQ--如何确保消息接收方消费了消息？

发送方确认模式

将信道设置成 confirm 模式（发送方确认模式），则所有在信道上发布的消息都会被派一个唯一的 ID。一旦消息被投递到目的队列后，或者消息被写入磁盘后（可持久化的消息），信道

发送一个确认给生产者（包含消息唯一 ID）。

若 MQ 发生内部错误从而导致消息丢失，会发送一条未确认（nack）消息。

发送方确认模式是异步的，生产者应用程序在等待确认的同时，可以继续发送消息。确认消息到达生产者应用程序，生产者应用程序的回调方法就会被触发来处理确认消息。

接收方确认机制

消费者接收每一条消息后都必须进行确认（消息接收和消息确认是两个不同操作）。有消费者确认了消息，MQ 才能安全地把消息从队列中删除。