

# 如何保证RabbitMq消息不丢失

---

题目难度：★★★★

知识点标签：如何保证RabbitMq消息不丢失

学习时长：20分钟

## 题目描述

---

如何保证RabbitMq消息不丢失

## 解题思路

面试官问题可以从几个方面来回答：消息丢失的场景，如何防止消息丢失

## 丢失消息的场景

---

### 1.生产者弄丢了数据

生产者将数据发送到rabbitmq的时候，可能在传输过程中因为网络等问题而将数据弄丢了。

### 2.RabbitMq自己丢失了数据

如果没有开启rabbitmq的持久化，那么rabbitmq一旦重启，那么数据就丢了。所以必须开启持久化将消息持久化到磁盘，这样就算rabbitmq挂了，恢复之后会自动读取之前存储的数据，一般数据不会丢失。除非极其罕见的情况，rabbitmq还没来得及持久化自己就挂了，这样可能导致一部分数据丢失。

### 3.消费端丢失了数据

主要是因为消费者消费时，刚消费到，还没有处理，结果消费者就挂了，这样你重启之后，rabbitmq就认为你已经消费过了，然后就丢了数据。

## 如何防止消息丢失

---

### 1.生产者丢失消息

- 可以选择使用rabbitmq提供的事务功能，就是生产者在发送数据之前开启事务，然后发送消息，如果消息没有成功被rabbitmq接收到，那么生产者会受到异常报错，这时就可以回滚事务，然后尝试重新发送；如果收到了消息，那么就可以提交事务。

缺点：rabbitmq事务已开启，就会变为同步阻塞操作，生产者会阻塞等待是否发送成功，太耗性能会造成吞吐量的下降。

- 可以开启confirm模式。在生产者哪里设置开启了confirm模式之后，每次写的消息都会分配一个唯一的id，然后如何写入了rabbitmq之中，rabbitmq会给你回传一个ack消息，告诉你这个消息发送OK了；如果rabbitmq没能处理这个消息，会回调你一个nack接口，告诉你这个消息失败了，你可以进行重试。而且你可以结合这个机制知道自己在内存里维护每个消息的id，如果超过一定时

间还没接收到这个消息的回调，那么你可以进行重发。

## 2.RabbitMq自己丢失了数据

设置消息持久化到磁盘。设置持久化有两个步骤：

①创建queue的时候将其设置为持久化的，这样就可以保证rabbitmq持久化queue的元数据，但是不会持久化queue里面的数据。

②发送消息的时候讲消息的deliveryMode设置为2，这样消息就会被设为持久化方式，此时rabbitmq就会将消息持久化到磁盘上。

必须要同时开启这两个才可以。

而且持久化可以跟生产的confirm机制配合起来，只有消息持久化到了磁盘之后，才会通知生产者ack，这样就算是在持久化之前rabbitmq挂了，数据丢了，生产者收不到ack回调也会进行消息重发。

## 3.消费者丢失了数据

使用rabbitmq提供的ack机制，首先关闭rabbitmq的自动ack，然后每次在确保处理完这个消息之后，在代码里手动调用ack。这样就可以避免消息还没有处理完就ack。

## 总结

---

1. 设置了队列和消息的持久化，当 RabbitMQ 服务重启之后，消息依旧会存在；
2. 仅设置队列持久化，重启之后消息会丢失；
3. 仅设置消息持久化，重启之后队列会消失，因此消息也就丢失了，所以只设置消息持久化而不设置队列持久化是没有意义的；
4. 将所有的消息都设置为持久化（写入磁盘的速度比写入内存的速度慢的多），可能会影响 RabbitMQ 的性能，对于可靠性不是那么高的消息可以不采用持久化来提高 RabbitMQ 的吞吐量。