

# 消息的发送和接收

2021年11月1日 23:46

## 发送消息的方式

### 同步的

发送方发出消息之后，会处于等待状态，知道接收方发出确认消息，接收到响应之后才会发送下一条消息

### 异步的

发送方发出消息之后，不用等待接收方的回应，会给接收方提供一个回调函数，当接收方成功消费消息之后就发回一个成功的响应，失败则发回一个失败的响应。是基于回调的方式来处理消费的消费结果的

### 单向的

不需要回调，发送方发出消息之后就不用管了。对于简单的业务，或者我们认为消息服务器一定可以成功处理的业务，我们就可以采用单向发送的方式

## 消费/接收消息的方式

### 拉取型消费（主动消费）

消费者主动从服务器获取消息，当获取到消息之后，再启动消费的过程

### 推送型消费

服务器主动推送消息到消费者。这种方式下，消息服务器不会考虑消费者的消费能力。这种方式用的比较少

## 消费的顺序

### 顺序消费

消息在消费的时候必须按照发送的顺序来逐一消费，这样的方式会让消息服务器中只维持一个消息队列，在这个消息队列中只有一个消费者来承担消费，对性能的影响较大

### 并行消费

发送到Topic中的消息是由多个消费者来并行处理的，不能保证先发过来的消息会被先消费

## 消息的类型

### 普通消息

前面提到的所有消息

### 延时消息

生产者在发送这个消息到消息服务器RocketMQ时，它不希望消费者立即去消费这条消息，而

是可以指定一个延时，即在一个特定的时间点之后再消费

常用的场景是电商平台中的支付问题，因为电商平台中的支付是由第三方支付平台来完成的，我们的系统并不知道订单是否支付成功，要等第三方支付平台回调我们的接口才知道支付成功，这就需要等待。也可能是用户下了一个订单之后没有立刻支付

所以我们一般会定义一个时间窗口，一般是30分钟，下了订单之后三十分钟没有支付，这个订单就会被取消

这种场景用延时消息去实现是很容易的，比如我们可以发一个延迟30分钟消费的消息，当消费者在三十分钟之后收到这条消息，再去检查订单，如果这个订单仍然是未支付状态（或者说没有收到第三方的回调函数），我们就取消订单

我们要注意延时消息并不是制定要在某个时间点去消费，在RocketMQ中延迟消费是有固定的时间间隔的，分别是1s、5s、10s、30s等一直到分钟、小时等。

当我们指定延时发送的时候，我们是指定一个整数，代表上面间隔里面的第几级

因为RocketMQ的延时消息是利用消息消费中的重发机制实现的，消息服务器会把没有发送成功或没有处理成功的消息放到重发队列中，队列中就是时间间隔，延时消费就是用这些时间间隔来实现的

所以延时消息不是放到当前消费者队列中，而是放到重发队列中，当到达我们指定的时间间隔才放到消费者队列中

## 事务消息

事务消费是和数据库事务关联在一起的消息

当我们在一个事务中发送消息时，消息服务器并不会把消息立刻放到消费者的队列中，而是通过回调函数来查看在生产者这边的事务执行情况，如果事务执行成功，消费者才能看到这个消息，之后在进行消费