

## 75\_知己知彼：先看看未引入 MQ 时，大量超时订单的取消逻辑

---

**儒猿架构官网上线**，内有石杉老师架构课最新大纲，儒猿云平台详细介绍，敬请浏览

官网：[www.ruyuan2020.com](http://www.ruyuan2020.com)（建议 PC 端访问）

---

### 1、开篇

上节课作为本周的第一节课，专注于取消订单的讲解，主要讲解了两种取消订单的场景：主动取消订单和超时取消订单，以及取消订单的业务流程。本节课会和大家讲讲，未引入 MQ 时，大量超时订单的取消逻辑。今天课程的内容包括以下几个部分：

- 超时订单的取消逻辑
- 传统取消订单存在的问题

### 2、超时订单的取消逻辑

上节课说了主动取消订单和超时取消订单的场景，主动取消订单好理解，也就在小程序端提供一个“取消订单”的按钮，点击以后完成订单取消的流程。用了优惠券的返还，用了积分的退回，一切看上去都是顺利成章。但是对于超时取消订单来说，并不是消费者主动操作的结果，出发这个条件的前提是一段长时间没有支付。如果我们加入假设条件，30 分钟没有支付的订单系统会自动取消，那么取消订单就要满足两个条件：从订单创建开始计算 30 分钟；同时订单状态为待支付。订单状态容易判断，但是对于时间而言判断起来就麻烦了，于是就引入了 Job 的概念。Job 也是一段应用程序，随着操作系统的启动会常驻其中，一直运行，通过 Job 去不断监听订单的创建时间和当前时间之间的差，一旦超过一定时间，例如：30 分钟，同时订单状态也满足待支付就会对其进行取消操作。顺着这个思路我们对取消订单的流程进行改造，如图 1 所示，我们将订单取消流程分为如下几步：

在“生成订单”以后，消费者可以在订单列表中找到自己创建的订单，并且可以通过点击“取消订单”按钮取消对应的订单，也是我们提到的主动取消订单。

1. 通过设置订单状态和判断时间的条件到 Job 上，Job 则不断“监视”订单的这些状态或者条件，一旦满足就执行取消订单的操作。根据我们的假设，将条件设置为：订单状态=“待支付” && 创建时间<当前时间-30 分钟。解释出来就是订单状态为待支付，并且超时 30 分钟的订单将被取消。
2. 当满足上面条件的时候，就由系统提供的 Job 去执行取消订单的操作。具体执行内容和主动取消订单基本一致，包括订单状态的修改、退回优惠券、返回积分等操作。

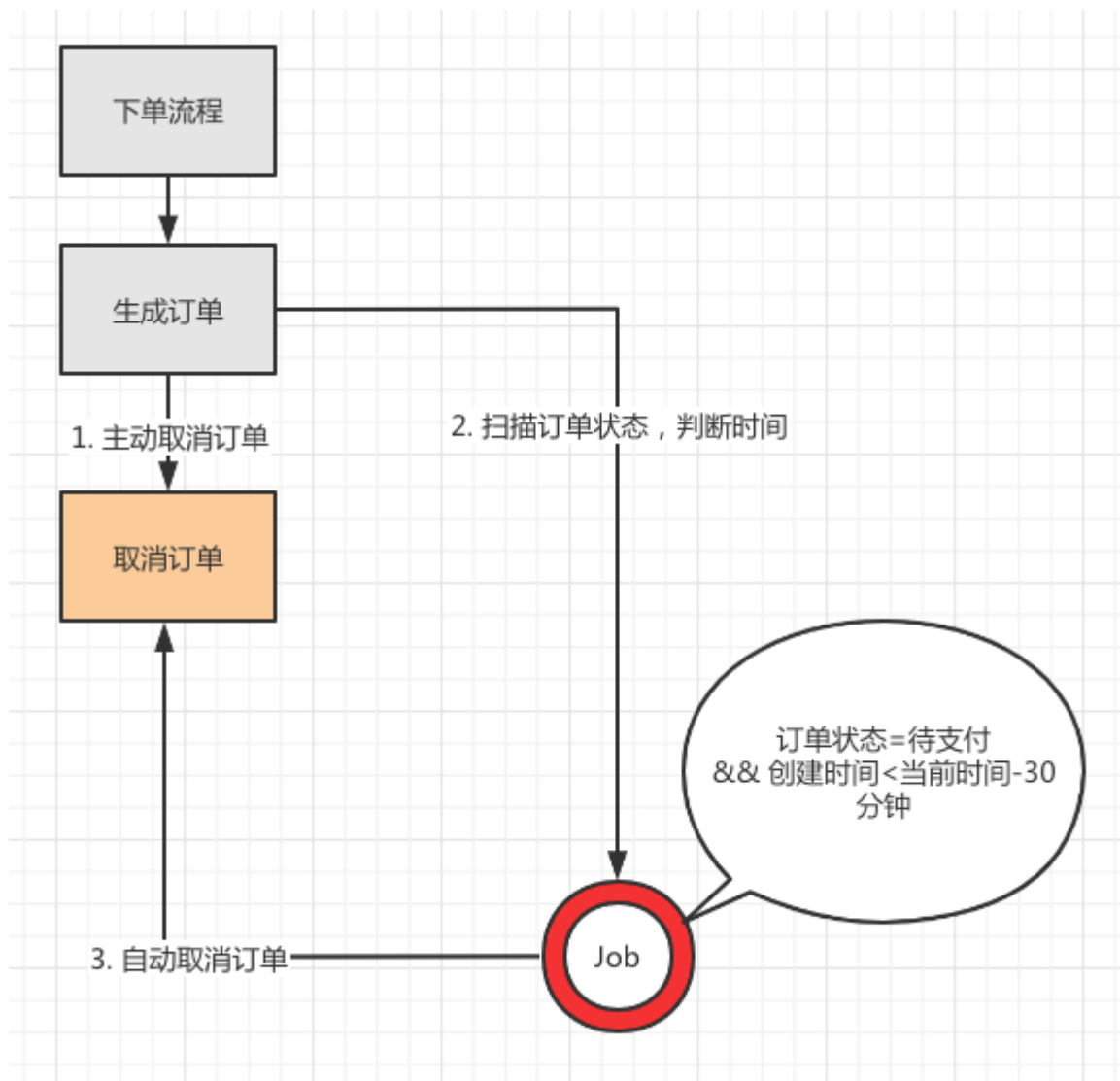


图 1 没有引入 MQ 之前如何取消订单

### 3、传统取消订单存在的问题

上面提到的使用 **Job** 监听的方式对于满足条件的订单进行取消处理，这种方式虽然可行，但是想想还是存在一些问题：

性能问题，为了完成超时取下订单的功能，系统中需要承载大量的 **Job**，**Job** 势必会占用系统的资源。而且这些 **Job** 会定时对数据库的表进行扫描，对数据库资源也是一种占用，从而造成了系统资源的浪费。

维护 **Job**，为了完成订单的业务逻辑需要维护 **Job**，这些 **Job** 虽然和应用系统进行了解耦，但是对于这些 **Job** 的日常维护还是需要很大的成本。

基于上面两个问题，就需要寻求更加优质的解决方案，这个也是我们为什么引入队列的目的，下节课我们会提出 **RocketMQ** 的解决方案，看能否让问题有所改善。

### 4、总结

本节课介绍了在未引入 **MQ** 之前如何完成超时订单取消的功能，通过创建 **Job** 不断监听订单的状态和超时时间，并且在满足设定条件的情况下对订单进行取消。虽然这个方案可行，但是也会遇到维护和性能的问题。下节课来看 **MQ** 如何基于延时消息机制，优化订单取消的问题。下期见，拜拜。