

# 订单

## 订单

---

### 订单是什么？

随着电商行业增速的回落，野蛮生长的阶段已经过去，电商开始进入精细化运作阶段，商家之间竞争从比拼营销和低价，转移到后台的供应链，供应链效率和成本的高低决定了商家的生死。

在电商的供应链管理中，有两块业务是核心内容，即订单处理和库存管理。订单是所有后台业务的源头，在多平台开店已经成为普遍采用的操作手法之后，如何高效、协调地管理来源于不同平台的订单成为影响用户口碑的关键环节。电商消除了购物过程中的空间限制，但作为实体存在的商品需要存放、需要配送，无法消除地理空间上的限制，如何快速、低成本地满足来自不同地域的消费需求，全局库存管理成为必须要良好解决的管理难题。

---

#### 一、多平台订单管理

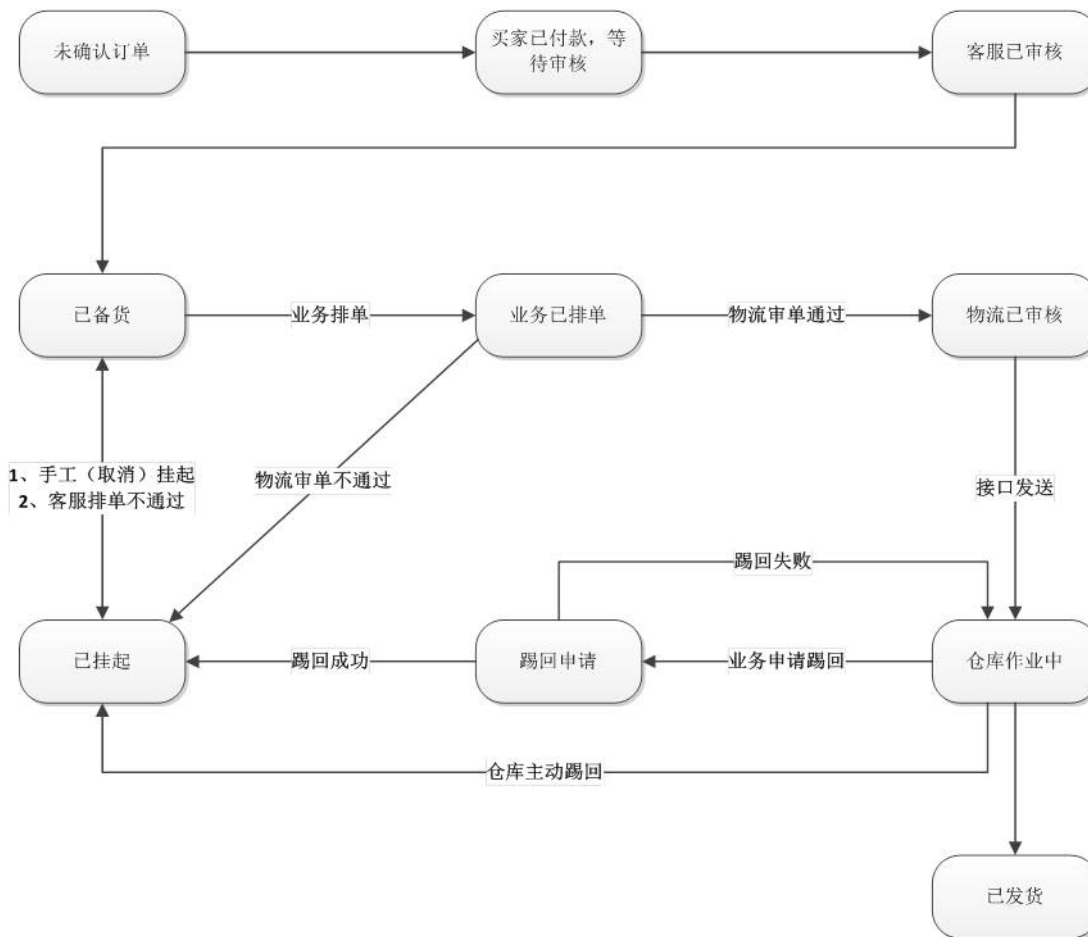
所谓多平台开店，就是在开展电商业务时，不仅仅建立自己的官网，也不仅仅在淘宝平台上开网店，而是和所有主流平台合作，目前，这些平台包括：淘宝、天猫、拍拍商城、京东、苏宁易购、一号店等等，在这些平台上开独立店或通过网络分销模式进行推广。

##### （一）如何统一不同平台上的订单

很明显，由于不同平台的设计思路的差异，每个平台的订单状态都是不同的，比如淘宝平台订单状态有：“没有创建支付宝交易”、“等待买家付款”、“买家已付款”、“卖家已发货”、“买家已签收”、“交易成功”、“交易自动关闭”、“卖家或买家主动关闭交易”。而京东平台的订单状态是：“等待出库”、“等待发货”、“已发货”、“买家已收货”、“已取消”。

这些状态中，有不少状态对平台管理有用，而对后续订单的处理是无用的；比如淘宝平台的“没有创建支付宝交易”状态，这个状态是淘宝网为了控制订单的支付过程中异常而设计的，而在海鼎系统（以下简称“系统”）中，没有支付成功的订单都不需要处理，去掉这些无用的状态，不同平台的订单状态就大同小异了，我们根据业务处理的需要，将订单统一成以下各种状态。

这些状态之间的关系如下：



注：（图、来自互联网禁止转载）

当订单从各个平台上下载到后台系统之后，系统根据对应规则，将所有订单的状态都统一到平台的订单状态中，以便后续的订单处理作业按照一致的方式完成。

（二）订单管理所用到的表：

商品扩展分类表	退货单	到货通知表	订单商品表	订单日志表	产品分类表	订单表	商品价格表	用户信息表	配送方式表
category_extend	returns_doc	notify_registry	order_goods	order_log	category	order	countsum	member	delivery

（三）订单的高并指的是什么？如何解决？

解决方案有三种

方法一：使用队列来实现(MemcacheQ等这样的消息队列)eg: 比如有100张票可供用户抢，那么就可以把这100张票放到缓存中，读写时不要加锁。当并发量大的时候，可能有500人左右抢票成功，这样对于500后面的请求可以直接转到活动结束的静态页面。进去的500个人中有400个人是不可能获得商品的。所以可以根据进入队列的先后顺序只能前100个人购买成功。后面400个人就直接转到活动结束页面。当然进去500个人只是举个例子，至于多少可以自己调整。而活动结束页面一定要用静态页面，不要用数据库。这样就减轻了数据库的压力

方法二：假设有m张票，有n台产品服务器接收请求，有x个请求路由服务器随机转发直接给每台产品服务器分配  $m/n$  张票每台产品服务器内存做计数器，比如允许  $m/n * (1+0.1)$  个人进来。当内存计数器已满：后面进的人，直接跳到到活动结束的静态页面，通知路由服务器，不在路由到这台服务器(这个值得商讨)。所有产品服务器进来的  $m/n * (1+0.1)$  个人再全部转发到一台付款服务器上，进入付款环节，看谁手快了，这时候人少，加锁什么的就简单的。

方法三：使用队列来实现(MemcacheQ等这样的消息队列)eg: 比如有100张票可供用户抢，那么就可以把这100张票放到缓存中，读写

时不要加锁。当并发量大的时候，可能有500人左右抢票成功，这样对于500后面的请求可以直接转到活动结束的静态页面。进去的500个人中有400个人是不可能获得商品的。所以可以根据进入队列的先后顺序只能前100个人购买成功。后面400个人就直接转到活动结束页面。当然进去500个人只是举个例子，至于多少可以自己调整。而活动结束页面一定要用静态页面，不要用数据库。这样就减轻了数据库的压力。

高并发是指在同一个时间点，有很多用户同时的访问URL地址，比如：淘宝的双11，双12，就会产生高并发，如贴吧的刷爆，就是恶意的高并发请求，也就是DDOS攻击。