

本文所描述的对账，非金融机构内部的对账场景。适用于公司自研以及采购的类电商系统，在此种场景下，平台背后支付渠道可能对接微信、支付宝亦或者是其他第三方聚合支付公司的通道。



## 一、支付系统综述

在如下图体系中，对账的目的是保障如下的等式成立：

客户支付的钱-支付渠道手续费=应结账款



有些渠道因为根据结算周期不一样或者其他运营因素，结算需要手续费。此种情况：应结账款-结算手续费=到账金额。

在一个完整的电商交易平台中，还会涉及折扣、积分兑换、满减、余额等支付场景，而且其中又夹杂多商户聚合支付。这种场景会让新入行的小伙伴手足无措。

以上两种场景的逻辑不应该放支付系统（支付模块）处理，而是应该放在交易模块的订单管理子模块处理。

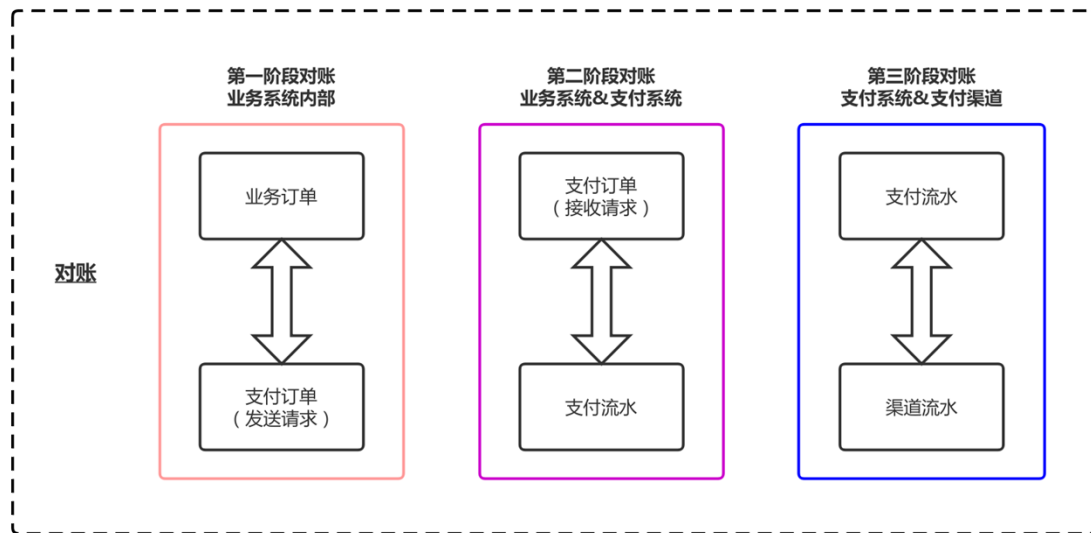
支付系统只关注实付金额的处理，一笔订单的订单金额、折扣等都不是支付系统关心的。在一个复杂的电商系统中（淘宝、京东、亚马逊），交易模块的核心工作之一就是处理好业务订单和支付订单的关系。

业务订单的核心属性：业务订单编号、下单日期、订单金额、订单状态、折扣信息、商户信息、客户信息。

支付订单的核心字段：支付订单编号、业务订单编号、支付时间、支付金额、商户信息、支付状态。

## 二、对账综述

明白了支付系统的定位和分工，在对账阶段所要关注的核心工作也就清楚了。



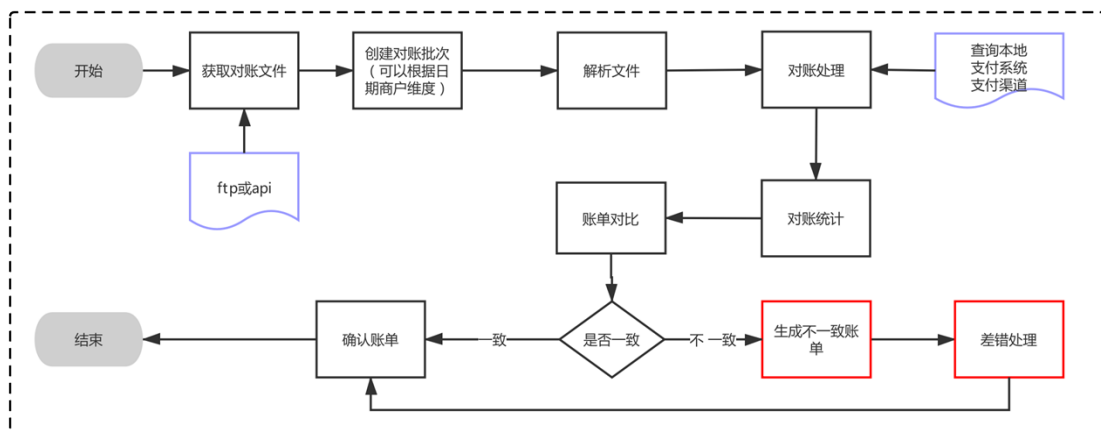
对账主要是保证三项数据的一致性：支付订单、支付流水、渠道流水。

这三项是分别是业务系统、支付系统、支付渠道的支付主数据，保证三者的 ID 关系和状态吻合既保证了支付流程的正确性。而渠道需要保障的渠道流水中的应结金额和实际结算金额的一致性，这个是支付渠道内部对账需要解决的问题。

第一阶段对账中会涉及会员积分的核销、运营折扣的匹配所以后期会专题分享；第二阶段与第三阶段很像，都是根据下游系统生产的对账文件和本地的订单或者流水核对。详细对账流程看下节。

### 三、对账流程

如下图，一般对账文件都是隔日才会生成，因为需要下游系统每日处理完前日的内部对账后才会生成给上游的对账文件。



**获取对账文件：**格式化存储，原始数据所有字段均存储；**创建对账批次：**因为有些系统涉及商户很多或者对接多个支付渠道，所以可以根据实际需求创建对账批次易于分类管理。常见以日期为一批次。但是下游对账文件出问题，可能需要当日需要再重新创建批次亦或是全量覆盖；**对账处理：**从格式化的对账文件中抽取六流水号、类型、状态、金额、商户号等关键字段和本系统的订单匹配；如果 ID+金额+状态 一致，则该笔订单直接核销，打上对账标记。

ID 指发送请求给下游时，下游返回存储在本系统的唯一主键。

对于不一致的场景会有三种情况，分别对应不同处理方案：

**本地有 ID，下游有 ID；**可以匹配但是状态不对；调用状态查询接口同步状态；**本地有 ID，下游无 ID；**根据 ID 查下游订单；**本地无 ID，下游有 ID；**根据 ID 查本地订单，或查询日志。

差错处理这一块，在实际涉及中，建议预留好人工处理的工具，等运行稳定后再根据实际情况把一些频繁出现的场景通过系统自动处理。

## 四、总结

本文中的对账针对场景是类电商系统和支付系统以及支付渠道之间的订单对账，并没有涉及钱包、资金托管模式下的财务对账。

本文提供的方法也主要是从业务需求出发，解决当平台的交易流程比较复杂的情况下，怎么保证订单信息、支付信息一致性的问题。

其他关于资金流水、应结资金、结算资金设计和财务 ERP、企业网银（通过银企互联获取账户流水）的对账，后续文章在——介绍。

### #专栏作家#

侠之大者，微信公众号：侠之大者（ID：xzdzkamil），人人都是产品经理专栏作家。关注互联网金融和企业信息化转型。

题图来自 Unsplash，基于 CC0 协议。