

全渠道开放服务商白皮书

阿里巴巴集团商家事业部

当前版本：V1.9

更新日期：2016.09.22

目录

| | |
|----------------------------|-----------|
| 全渠道开放服务商白皮书 | 1 |
| 目录 | 2 |
| 行业全渠道开放框架 | 6 |
| 1.1 框架模型 | 6 |
| 1.2 关键模块 | 6 |
| 1.2.1 天猫 B2C | 6 |
| 1.2.2 全渠道行业中台 (星盘) | 6 |
| 1.2.3 商家中台 | 7 |
| 1.2.4 全渠道后台 | 7 |
| 1.3 元数据 | 7 |
| 1.3.1 门店订单状态 | 7 |
| 1.3.2 门店信息 | 8 |
| 1.3.3 库存信息 | 8 |
| 1.3.4 分单策略 | 9 |
| 1.3.5 分账信息 | 9 |
| 1.4 核心场景 (门店发货) | 9 |
| 2 全渠道订单路由 | 11 |
| 2.1 全渠道订单 | 11 |
| 2.2 订单的三级路由 | 11 |
| 2.3 多子订单 | 12 |

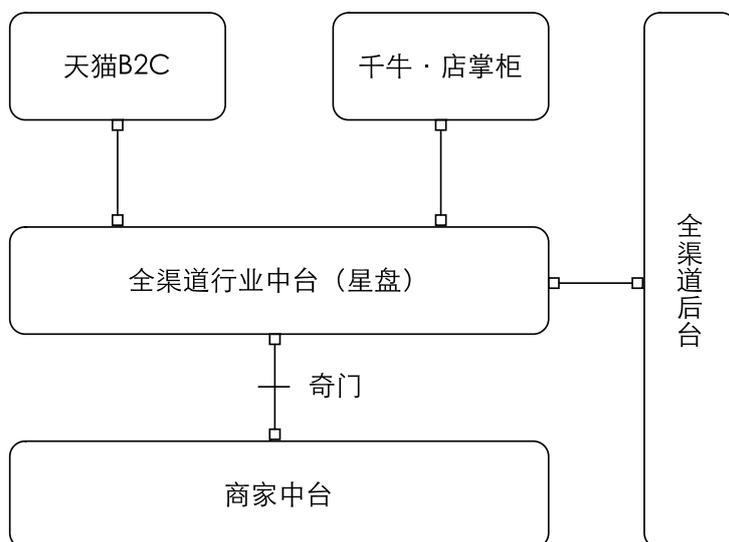
| | | |
|----------|-----------------------|-----------|
| 2.4 | 赠品..... | 13 |
| 3 | 全渠道模型参考实现..... | 13 |
| 3.1 | 星盘分单 + 店掌柜接单..... | 14 |
| 3.2 | ISV 分单 + 店掌柜接单 | 16 |
| 3.3 | 星盘分单 + POS 接单 | 19 |
| 3.4 | ISV 分单 + POS 接单 | 21 |
| 4 | 库存管理要求..... | 22 |
| 4.1 | 库存类别..... | 22 |
| 4.2 | 库存商品关联..... | 23 |
| 4.3 | 避免超卖..... | 24 |
| 5 | 作业系统指标要求 | 24 |
| 5.1 | 正向作业响应指标..... | 24 |
| 5.1.1 | 分单指标..... | 25 |
| 5.1.2 | 接单指标..... | 25 |
| 5.1.3 | 发货指标..... | 25 |
| 5.1.4 | 物流指标..... | 26 |
| 5.2 | 成熟度指标..... | 26 |
| 5.2.1 | 订单接单率..... | 26 |
| 5.2.2 | 门店派单率..... | 26 |
| 5.2.3 | 门店接单率..... | 27 |
| 5.3 | 逆向作业响应指标..... | 27 |

| | | |
|----------|------------------|-----------|
| 5.4 | 关键状态回传时间要求..... | 27 |
| 5.5 | 指标校验..... | 27 |
| 5.5.1 | 状态回传不及时的影响..... | 27 |
| 5.5.2 | 及时回传状态信息..... | 27 |
| 6 | 接口定义..... | 28 |
| 6.1 | 门店上传及商品关联..... | 28 |
| 6.1.1 | 门店新增接口..... | 28 |
| 6.1.2 | 门店类目获取接口..... | 29 |
| 6.1.3 | 门店更新接口..... | 30 |
| 6.1.4 | 门店删除接口..... | 31 |
| 6.1.5 | 门店信息查询接口..... | 32 |
| 6.1.6 | 商品关联绑定接口..... | 33 |
| 6.1.7 | 商品关联门店查询接口..... | 33 |
| 6.1.8 | 门店关联商品查询接口..... | 34 |
| 6.2 | 库存通(后端商品)..... | 35 |
| 6.2.1 | 门店库存初始化接口..... | 35 |
| 6.2.2 | 门店库存更新接口..... | 36 |
| 6.2.3 | 门店库存查询接口..... | 37 |
| 6.2.4 | 库存占用调整接口..... | 39 |
| 6.3 | 库存通 (前端商品)..... | 40 |
| 6.3.1 | 库存初始化接口..... | 40 |
| 6.3.2 | 库存增量更新接口..... | 41 |

| | | |
|----------|------------------------------|-----------|
| 6.3.3 | 库存查询接口..... | 42 |
| 6.3.4 | 库存占用调整接口..... | 44 |
| 6.4 | 订单全链路同步..... | 45 |
| 6.5 | 订单状态更新接口..... | 48 |
| 6.6 | 星盘开放接口 2.0..... | 50 |
| 6.6.1 | 派单/重新分单..... | 50 |
| 6.6.2 | 派单结果同步..... | 51 |
| 6.6.3 | 派单结果查询..... | 54 |
| 6.6.4 | 退款..... | 56 |
| 6.6.5 | 星盘通知 POS 已接单..... | 58 |
| 6.6.6 | 星盘通知 POS 已拒单..... | 59 |
| 6.6.7 | 星盘通知 POS 已发货..... | 60 |
| 6.6.8 | POS 接单通知星盘..... | 62 |
| 6.6.9 | POS 拒单通知星盘..... | 63 |
| 6.6.10 | POS 发货通知星盘..... | 65 |
| 7 | 全渠道 SPI 的接入和部署方式..... | 67 |
| 7.1 | SPI 接入方式..... | 67 |
| 7.2 | 全渠道 SPI 部署到聚石塔 EWS..... | 68 |
| 8 | ISV 系统压测..... | 69 |

行业全渠道开放框架

1.1 框架模型



1.2 关键模块

1.2.1 天猫 B2C

负责全部电商域交易生命周期维护和订单状态流转，并将这些信息实时通知到行业中台。

负责电商域商品、交易、库存等的信息和状态维护。

1.2.2 全渠道行业中台 (星盘)

作为电商域全渠道部分的信息系统中台，主要负责源自天猫 B2C 的全渠道订单二级、三级

路由功能 (详见 2.2 节)，订单状态维护功能以及全渠道规则管理功能。具体来说：

负责全渠道订单的生命周期管理。

负责分单、分账 (触发)、交易控制 (分阶段、大额支付) 等作业系统的控制和执行。

负责静态主数据和动态业务数据的处理和维。

星盘的功能会开放出来由 ISV 参与共建（具体参考第三章，全渠道模型参考实现）。

1.2.3 商家中台

商家自有企业域和作业系统的中台，对接包括 ERP、WMS、POS 等在内的商家系统。

1.2.4 全渠道后台

为商家提供门店、分单策略、分账规则等全渠道基础信息的配置，以及电商域的全渠道报表功能。

为商家提供全渠道分单逻辑服务商的选择配置，以及 WMS 和 POS 作业系统服务商的选择配置。

全渠道后台依赖于商户中心、天猫库存中心和天猫货品中心。

1.3 元数据

1.3.1 门店订单状态

通用状态：

| 描述 | 状态 | 是否必传 |
|-------|---------------------|------|
| 系统已接单 | X_TO_SYSTEM | 是 |
| 已客审 | X_SERVICE_AUDITED | 否 |
| 已财审 | X_FINANCE_AUDITED | 否 |
| 已打拣货单 | X_SORT_PRINTED | 否 |
| 已打发货单 | X_SEND_PRINTED | 否 |
| 已打物流单 | X_LOGISTICS_PRINTED | 否 |

| | | |
|-----|-----------------|---|
| 已拣货 | X_SORTED | 否 |
| 已验货 | X_EXAMINED | 否 |
| 已打包 | X_PACKAGED | 否 |
| 已称重 | X_WEIGHED | 否 |
| 已出库 | X_OUT_WAREHOUSE | 否 |

门店发货场景状态：

| 描述 | 状态 | 是否必传 |
|--------|-----------------------|------|
| 通知门店配货 | X_ALLOCATION_NOTIFIED | 是 |
| 门店已接单 | X_SHOP_ALLOCATION | 是 |

门店自提场景状态：

| 描述 | 状态 | 是否必传 |
|--------|-----------------------|------|
| 通知门店配货 | X_ALLOCATION_NOTIFIED | 是 |
| 门店确认有货 | X_COMMODITY_CONFIRM | 是 |

1.3.2 门店信息

详见门店信息创建奇门接口：taobao.qimen.store.create 的入参。

1.3.3 库存信息

详见库存初始化奇门接口：taobao.qimen.storeinventory.iteminitial 的入参。

1.3.4 分单策略

暂未开放。

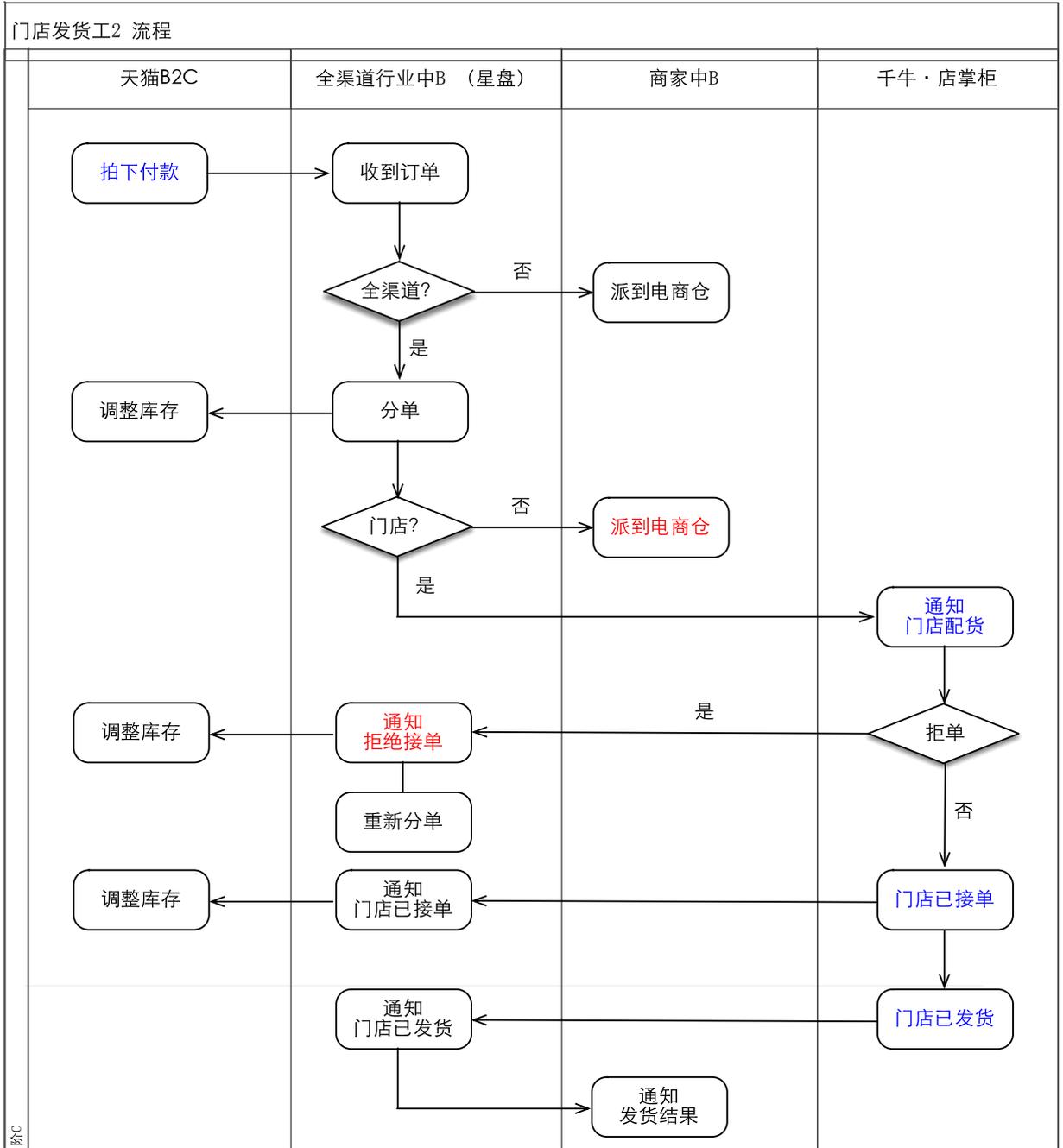
1.3.5 分账信息

暂未开放。

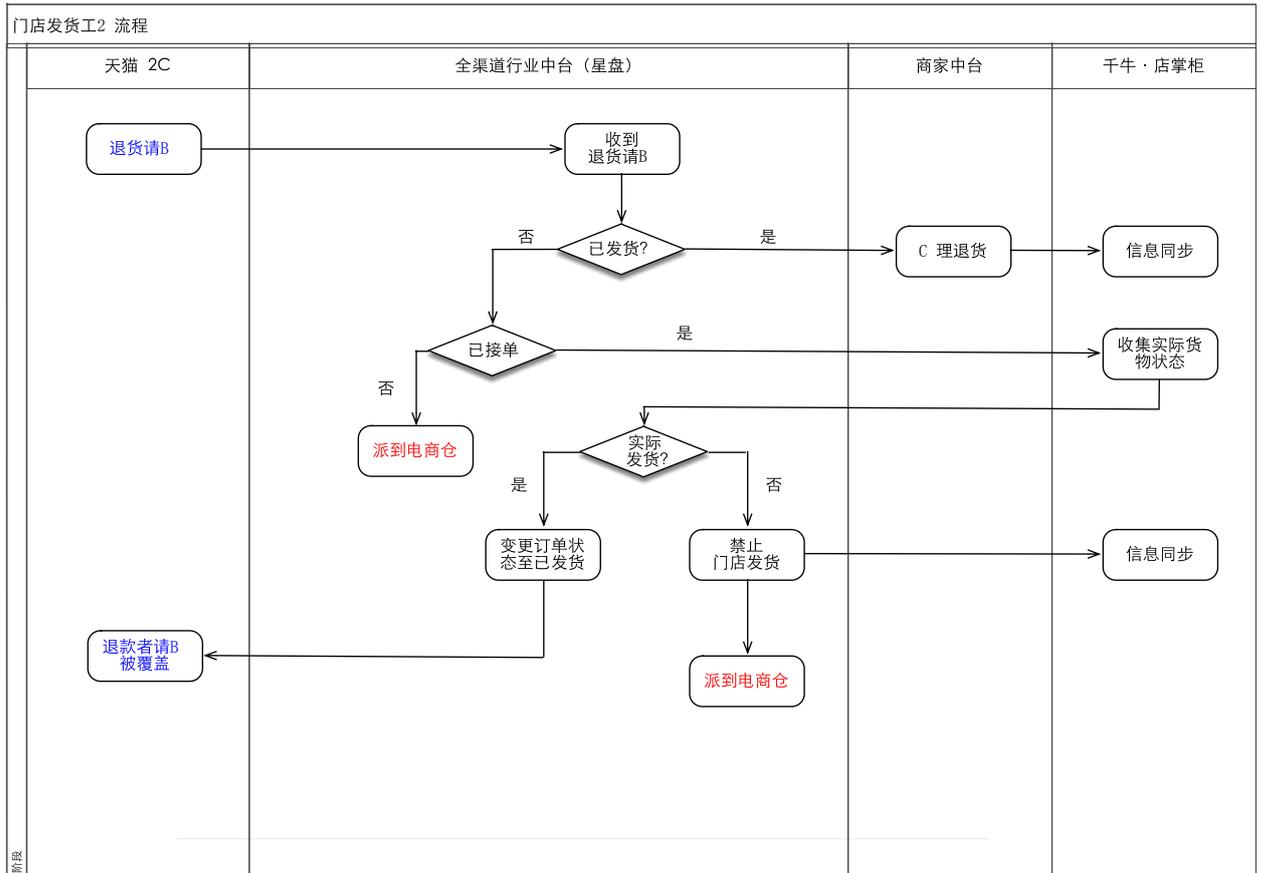
1.4 核心场景（门店发货）

本节以星盘和店掌柜的默认分单和接单实现逻辑为例，描述门店发货的作业流程。

正向流程：



逆向流程：



2 全渠道订单路由

2.1 全渠道订单

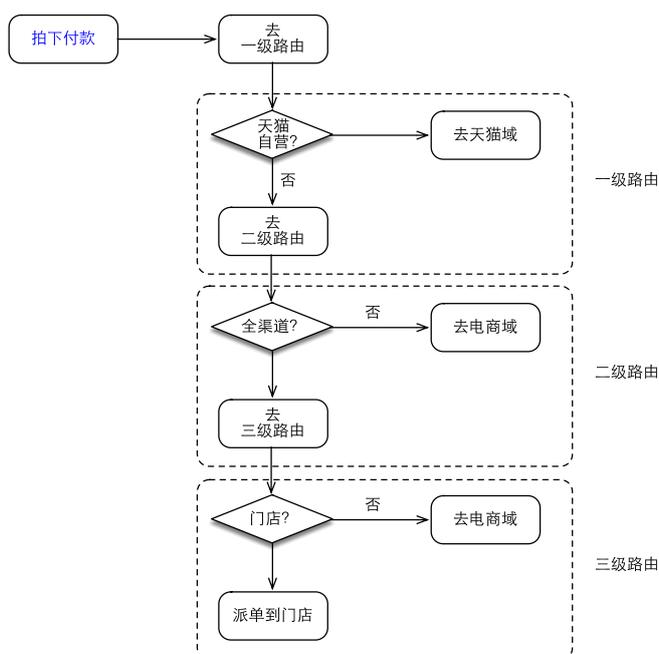
在电商域，所有打了全渠道标的商品属于全渠道商品。如果一笔订单中包含至少一个对应了全渠道商品的子订单，那么该订单属于全渠道订单。

全渠道订单在订单推送时，不会直接给到 RDS 的中间表；而是通过订单的三级路由进行流转，详见下节。

2.2 订单的三级路由

全渠道订单在分派过程中，会经过三次订单流通过程，称为全渠道订单的三级路由。其中第

一级路由，判断是否天猫自营，一级路由由天猫 B2C 完成；二级路由判断订单是否全渠道订单，二级路由由星盘订单网关完成；三级路由判断订单的发货门店或仓，根据商家在全渠道后台的设置，三级路由可以由星盘官方提供，也可以由 ISV 提供的分单应用提供。整个三级路由的示意图如下：



全渠道订单的第三级路由根据商家选择，可以由星盘进行也可以由 ISV 分单应用进行。在第三章全渠道参考实现中，星盘分单或 ISV 分单都是指的第三级路由。

商家选择 ISV 分单后，全渠道订单在流转第三级路由时，该订单会推送到对应的 ISV 的 RDS 中间表中（非全渠道订单处理应用将不会收到该订单，只有在订单状态为已发货的情况下才会收到该订单），同时星盘将调用接口“taobao.qimen.omniorder.allocate”通知 ISV 应用进行第三级路由。在 ISV 应用分单完成后，需将分单结果通过接口“taobao.omniorder.allocatedinfo.sync”同步给星盘。

2.3 多子订单

多子订单，指的是将主订单的多个子订单进行拆分，分别发到不同的门店/仓。

在本白皮书发布时，星盘本身不支持多子订单的处理。ISV 如果实现多子订单逻辑，需要注意：

如果部分子订单被派到了电商仓，部分派到了门店，需确保商家的 WMS 系统可以正确排除掉已发往门店的子订单，而 POS 系统可以正确排除掉发往电商仓的子订单，以避免重复发货。

每个子订单具体的分派情况，由订单结构体中的 omnichannel_param 扩展字段进行标示。

商家的 WMS 系统和 POS 系统需要进行改造，确保理解这些信息。

2.4 赠品

赠品、批次号等是商家在作业系统中经常使用，但是电商域本身不感知的信息。为了使这些信息可以在分单系统、WMS 系统和 POS 系统中进行传递，omnichannel_param 中也定义了相关字段。商家的分单、WMS 和 POS 系统需要进行改造以支持这些字段的写入（通过“taobao.omniorder.allocatedinfo.sync”接口同步给星盘）和解析。

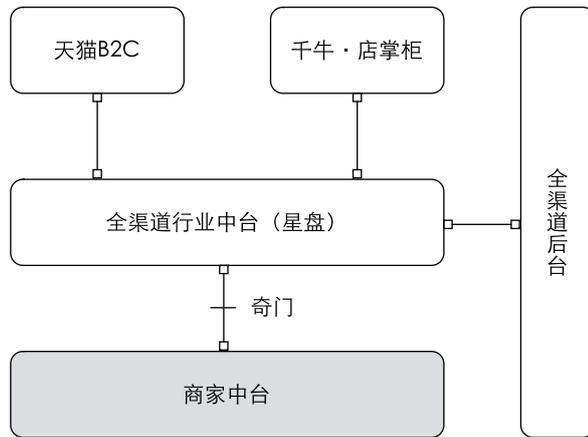
3 全渠道模型参考实现

全渠道电商域的信息状态由星盘进行统一维护，由全渠道后台（全局）和千牛·店掌柜（单个门店）进行统一报表展示。官方提供了分单逻辑和门店作业（接单、发货等）的基础功能，并在此基础上开放给 ISV 共建，以适应更广泛的行业需求。根据分单逻辑和接单逻辑实现的不同，全渠道模型分为下表所示 4 种实现形式，表中同时给出了不同实现的商家参考依据：

| 参考实现 | 是否有线上分单系统 | 是否有门店作业系统 | 是否有行业定制需求 | 接入难度 |
|------|-----------|-----------|-----------|------|
| | 系统 | 系统 | 需求 | 度 |

| | | | | |
|---------------|---|---|---|---------|
| 星盘分单+店掌柜接单 | 否 | 否 | 否 | 容易 |
| ISV 分单+店掌柜接单 | 是 | 否 | 是 | 中 |
| 星盘分单+POS 接单 | 否 | 是 | 否 | 容易 |
| ISV 分单+POS 接单 | 是 | 是 | 是 | 很容 易 |

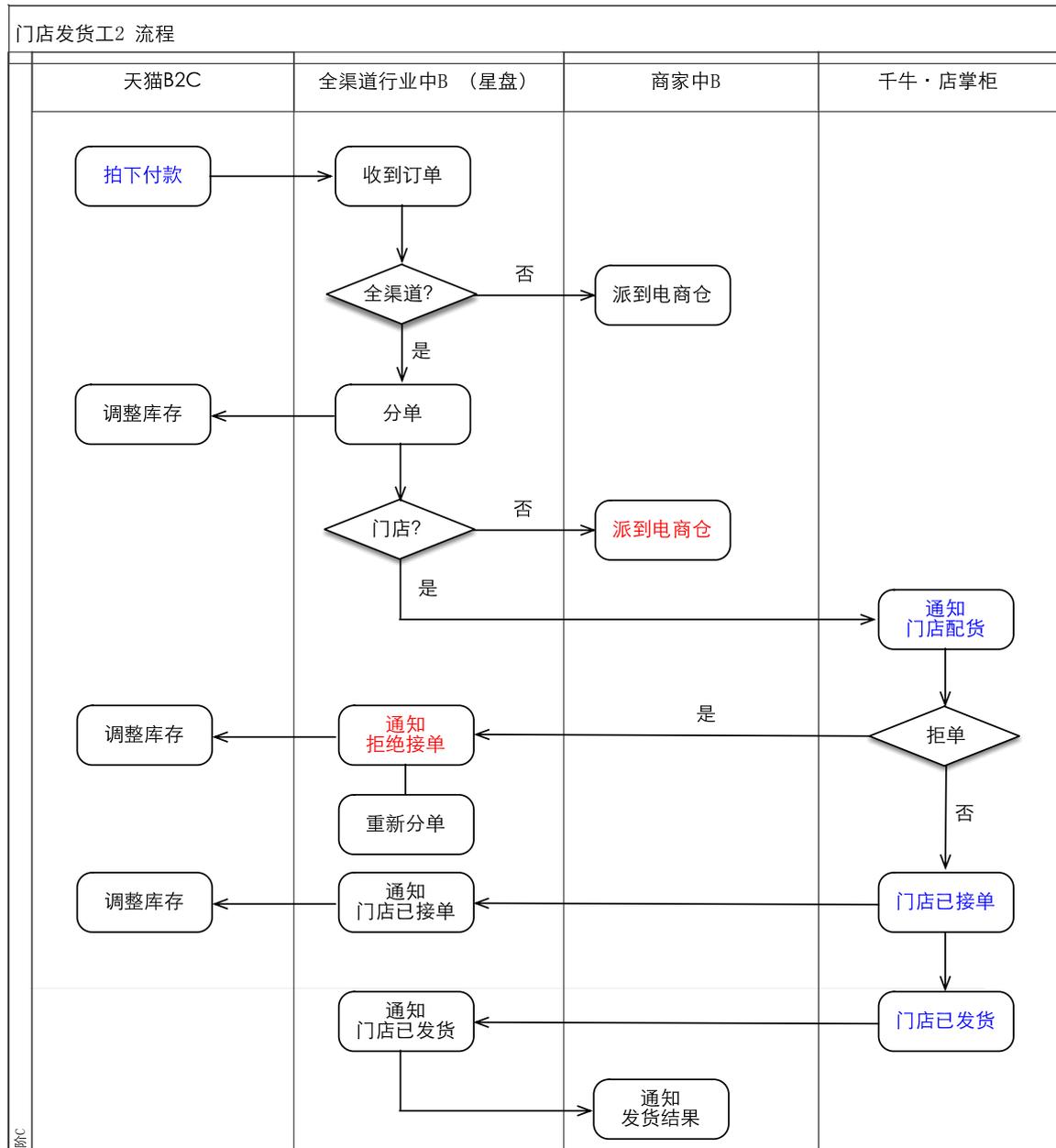
3.1 星盘分单 + 店掌柜接单



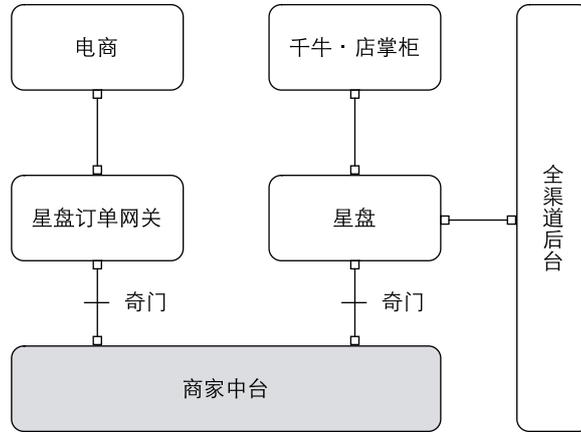
关键接口：

| 接口分类 | 方向 | 接口名 | 描述 |
|------|--------|---|-----------------|
| 门店信息 | 商家->星盘 | taobao.qimen.store.create | 门店新增接口 |
| | 商家->星盘 | taobao.qimen.storecategory.get | 查询门店主营类目信息接口 |
| | 商家->星盘 | taobao.qimen.store.update | 更新门店信息接口 |
| | 商家->星盘 | taobao.qimen.store.delete | 批量删除线下门店数据接口 |
| | 商家->星盘 | taobao.qimen.store.query | 查询门店信息接口 |
| | 商家->星盘 | taobao.qimen.itemstore.banding | 新建/删除商品和门店的绑定关系 |
| | 商家->星盘 | taobao.qimen.itemstore.query | 查询线上商品所关联的门店列表 |
| | 商家->星盘 | taobao.qimen.storeitem.query | 查询门店所关联的线上商品列表 |
| 库存信息 | 商家->星盘 | taobao.qimen.storeinventory.iteminitial | 全量更新电商仓或门店库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemupdate | 增量更新门店或电商仓库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemquery | 查询门店或电商仓库存 |

主要流程：



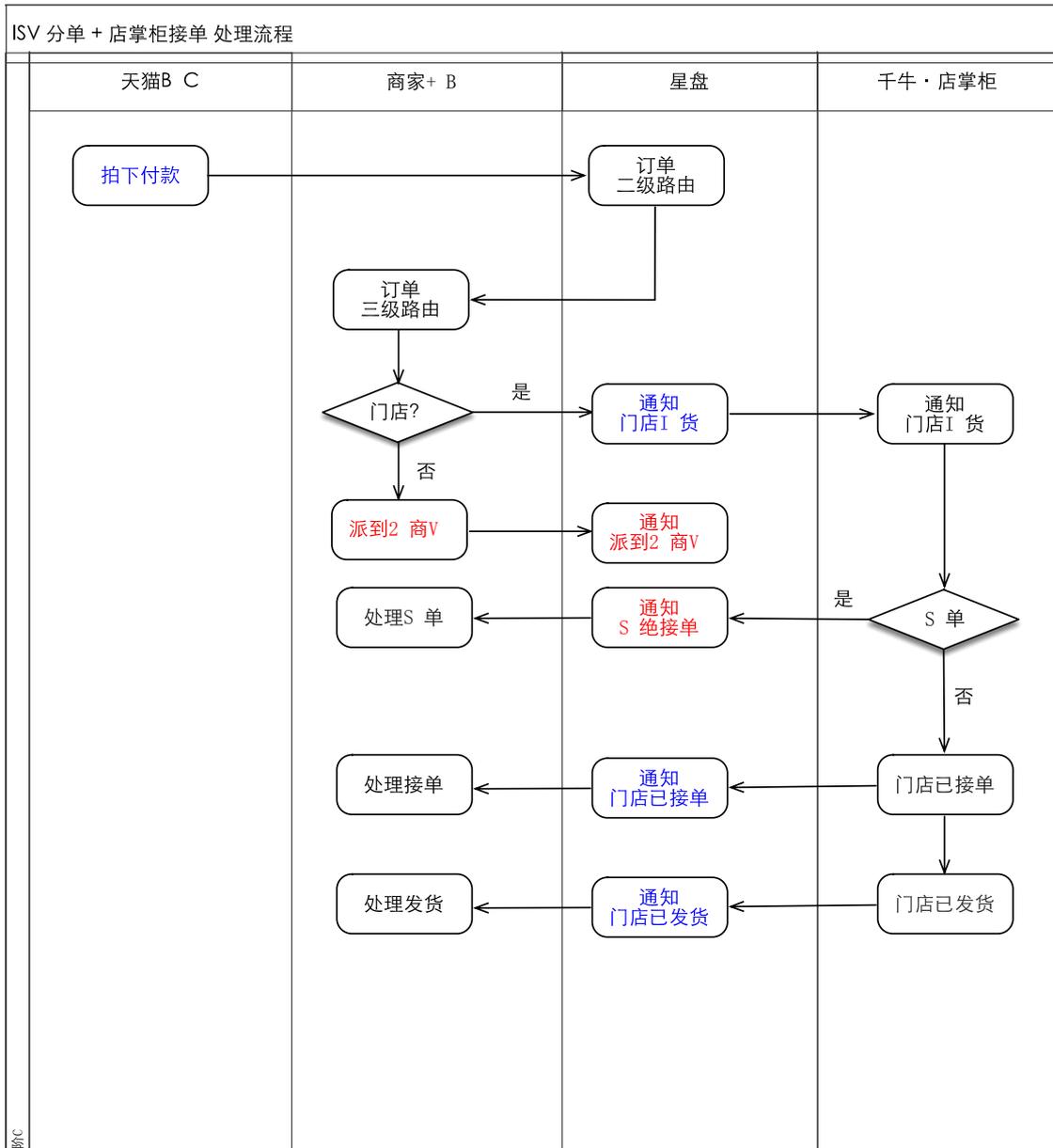
3.2 ISV 分单 + 店掌柜接单



关键接口：

| 接口分类 | 方向 | 接口名 | 描述 |
|------|--------|--|------------------|
| 门店信息 | 商家->星盘 | taobao.qimen.store.create | 门店新增接口 |
| | 商家->星盘 | taobao.qimen.storecategory.get | 查询门店主营类目信息接口 |
| | 商家->星盘 | taobao.qimen.store.update | 更新门店信息接口 |
| | 商家->星盘 | taobao.qimen.store.delete | 批量删除线下门店数据接口 |
| | 商家->星盘 | taobao.qimen.store.query | 查询门店信息接口 |
| | 商家->星盘 | taobao.qimen.itemstore.banding | 新建/删除商品和门店的绑定关系 |
| | 商家->星盘 | taobao.qimen.itemstore.query | 查询线上商品所关联的门店列表 |
| | 商家->星盘 | taobao.qimen.storeitem.query | 查询门店所关联的线上商品列表 |
| 库存信息 | 商家->星盘 | taobao.qimen.storeinventory.iteminitial | 全量更新电商仓或门店库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemupdate | 增量更新门店或电商仓库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemquery | 查询门店或电商仓库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemadjust | 调整所占用的门店/电商仓库存 |
| 作业流转 | 星盘->商家 | taobao.qimen.omniorder.allocate | 启动分单（实时通知）或者重新分单 |
| | 商家->星盘 | taobao.omniorder.allocatedinfo.sync | ISV 同步派单结果到星盘 |
| | 星盘->商家 | taobao.qimen.omniorder.allocatedinfo.get | 派单状态查询 |
| | 星盘->商家 | taobao.qimen.omniorder.store.accpeted | 星盘通知 ISV 已接单状态 |
| | 星盘->商家 | taobao.qimen.omniorder.store.refused | 星盘通知 ISV 已拒单状态 |
| | 星盘->商家 | taobao.qimen.omniorder.store.consigned | 星盘通知 ISV 发货状态 |

主要流程：



订单获取说明：

- 方式一：使用启动分单接口 taobao.qimen.omniorder.allocate 实时获取订单

ISV 实现启动分单 SPI (taobao.qimen.omniorder.allocate), 一旦有全渠道订单进入第三级路由, 星盘会先调用该 SPI 告知 ISV 分单系统有订单产生 (主订单 ID), 并根据 ISV 返回的 Appkey 将订单信息推送到指定 appkey 下的 RDS 中间表中, 然后 ISV 就可以通过查询 RDS 中间表获取订单信息了。

- 方式二：通过定时轮询 RDS 中间表来获取订单信息

订单推送的过程同第一种方式类似,只不过推送的RDS不再由ISV的返回Appkey来确定,而是将订单信息推送到ISV入驻时提交给小二的Appkey下的RDS中间表中。ISV可以不用理会星盘的启动分单调用 (taobao.qimen.omniorder.allocate),而是按自己的需要按时轮询RDS中间表批量获取订单进行处理。

在这种情况下,ISV只需要将启动分单SPI (taobao.qimen.omniorder.allocate) 的实现mock就可以了,星盘也会按流程在推送订单前调用它通知ISV,但不会使用返回的信息。

退款订单获取说明:

- 方式一:使用发起退款接口 taobao.qimen.omniorder.customer.refund 实时获取退款订单

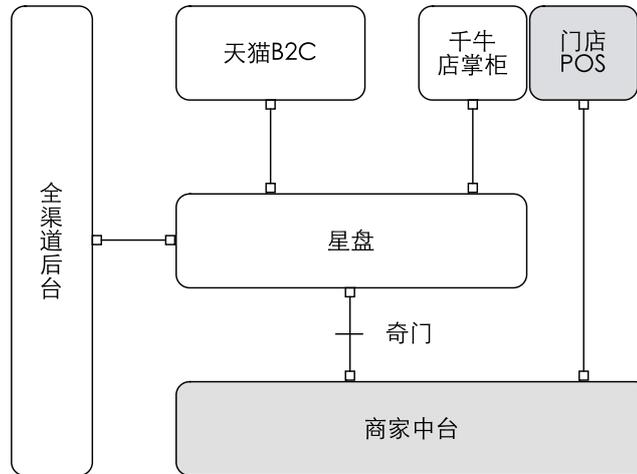
ISV实现退款通知的SPI (taobao.qimen.omniorder.customer.refund),一旦有客户退款申请产生,星盘会先调用该SPI告知ISV退款订单的信息 (主订单ID和子订单ID),然后也会推送到ISV的Appkey下的RDS中间表中,然后ISV就可以通过查询RDS中间表获取退款订单信息了。

- 方式二:通过定时轮询RDS中间表来获取退款订单信息

退款订单推送的过程同第一种方式类似,只不过ISV可以不用理会星盘的退款通知调用 (taobao.qimen.omniorder.customer.refund),而是按自己的需要按时轮询RDS中间表批量获取退款订单进行处理。

在这种情况下,ISV只需将退款订单通知SPI(taobao.qimen.omniorder.customer.refund) 的实现mock就可以了,因为星盘也会按流程在推送退款订单前调用它通知ISV。

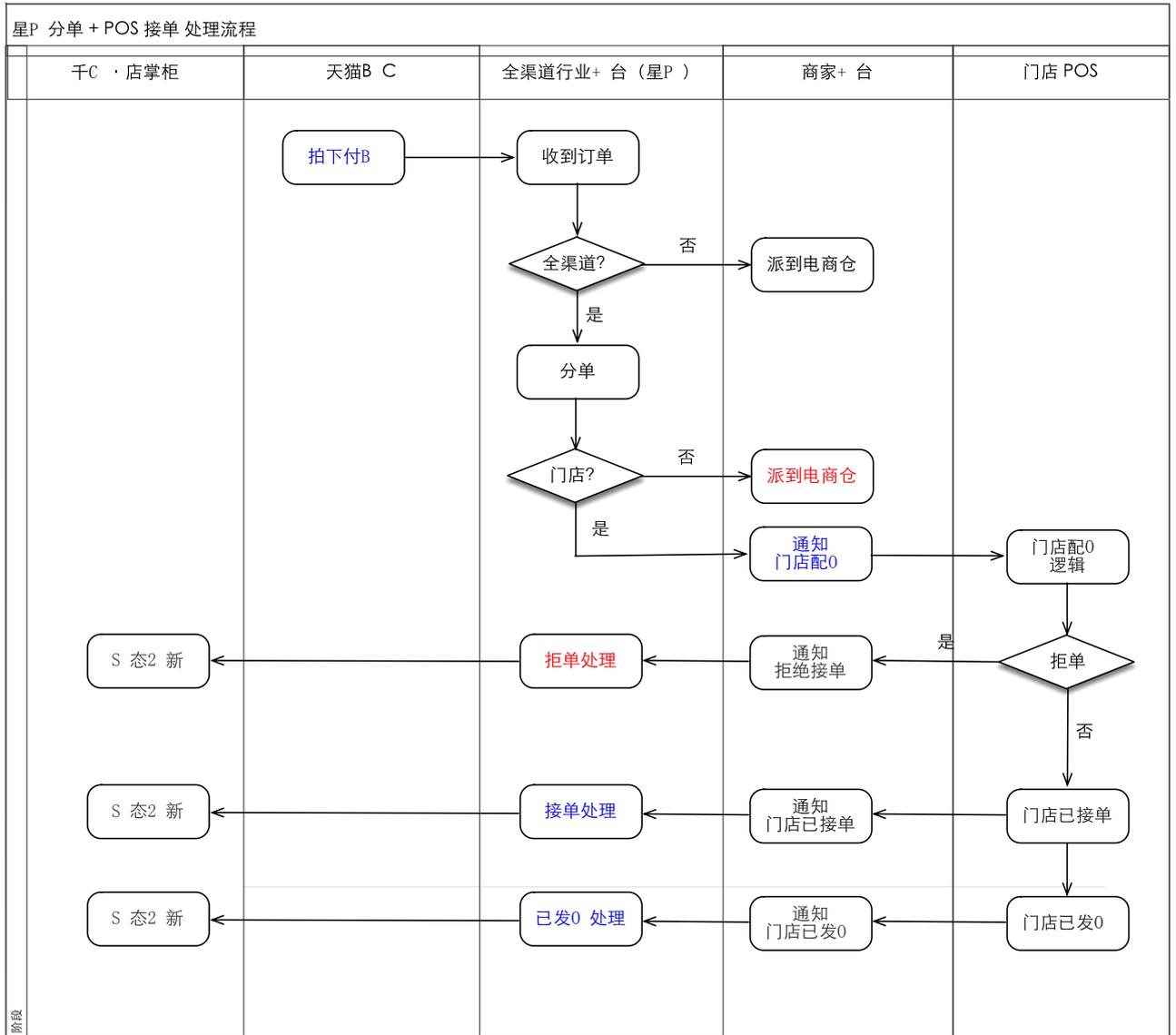
3.3 星盘分单 + POS 接单



关键接口：

| 接口分类 | 方向 | 接口名 | 描述 |
|------|--------|---|---------------------|
| 门店信息 | 商家->星盘 | taobao.qimen.store.create | 门店新增接口 |
| | 商家->星盘 | taobao.qimen.storecategory.get | 查询门店主营类目信息接口 |
| | 商家->星盘 | taobao.qimen.store.update | 更新门店信息接口 |
| | 商家->星盘 | taobao.qimen.store.delete | 批量删除线下门店数据接口 |
| | 商家->星盘 | taobao.qimen.store.query | 查询门店信息接口 |
| | 商家->星盘 | taobao.qimen.itemstore.banding | 新建/删除商品和门店的绑定关系 |
| | 商家->星盘 | taobao.qimen.itemstore.query | 查询线上商品所关联的门店列表 |
| | 商家->星盘 | taobao.qimen.storeitem.query | 查询门店所关联的线上商品列表 |
| 库存信息 | 商家->星盘 | taobao.qimen.storeinventory.iteminitial | 全量更新电商仓或门店库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemupdate | 增量更新门店或电商仓库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemquery | 查询门店或电商仓库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemadjust | 调整所占用的门店/电商仓库存 |
| 作业流转 | 星盘->商家 | taobao.qimen.omniorder.allocatedinfo.sync | 星盘同步派单结果到 POS(商家中台) |
| | 商家->星盘 | taobao.omniorder.store.accpeted | 门店接单 (ISV 门店端) |
| | 商家->星盘 | taobao.omniorder.store.refuse | 门店拒单 (ISV 门店端) |
| | 商家->星盘 | taobao.omniorder.store.consigned | 门店发货 (ISV 门店端) |

关键流程：



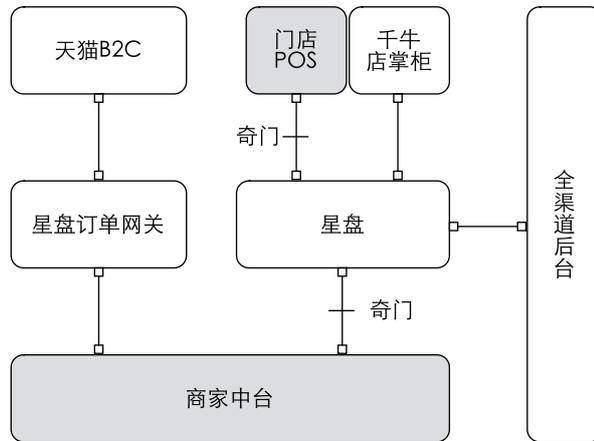
星盘同步派单结果到商家中台：

- 方式一：使用同步派单结果接口 `taobao.qimen.omniorder.allocatedinfo.sync` 获取

一旦有订单在星盘完成分单，星盘会先调用该 SPI 通知 ISV 分单结果，然后还会将分单结果同步到 ISV 的 Appkey 的 RDS 中间表中，ISV 就可以通过查询 RDS 中间表获取分单到门店的信息了。
- 方式二：通过定时轮询 RDS 中间表来获取分单结果

ISV 根据自己的需要按时轮询 RDS 中间表获取派单结果。这种情况下，ISV 无需实现同步派单结果的接口 `taobao.qimen.omniorder.allocatedinfo.sync`。

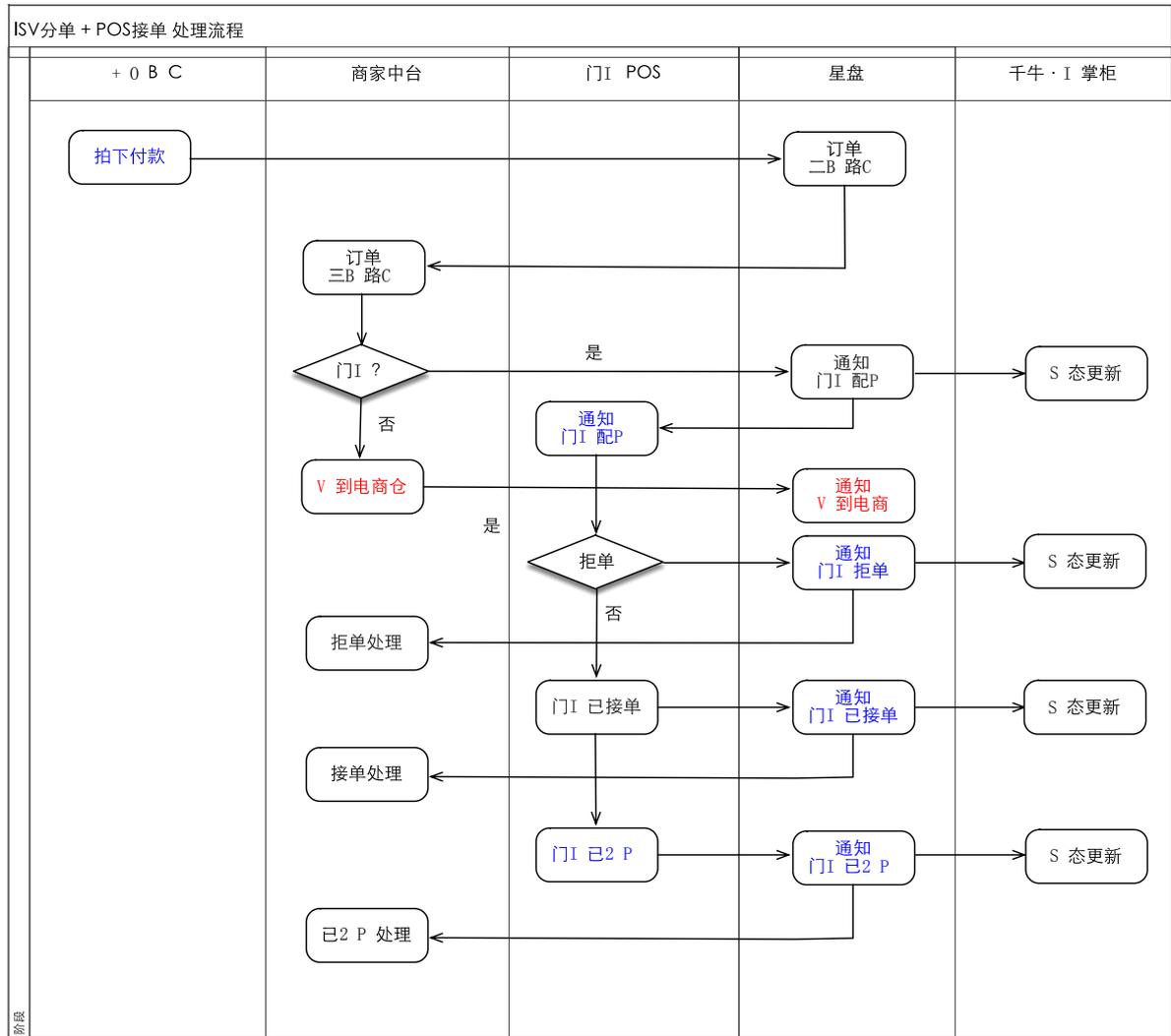
3.4 ISV 分单 + POS 接单



关键接口

| 接口分类 | 方向 | 接口名 | 描述 |
|--------|--|--|-------------------|
| 门店信息 | 商家->星盘 | taobao.qimen.store.create | 门店新增接口 |
| | 商家->星盘 | taobao.qimen.storecategory.get | 查询门店主营类目信息接口 |
| | 商家->星盘 | taobao.qimen.store.update | 更新门店信息接口 |
| | 商家->星盘 | taobao.qimen.store.delete | 批量删除线下门店数据接口 |
| | 商家->星盘 | taobao.qimen.store.query | 查询门店信息接口 |
| | 商家->星盘 | taobao.qimen.itemstore.banding | 新建/删除商品和门店的绑定关系 |
| | 商家->星盘 | taobao.qimen.itemstore.query | 查询线上商品所关联的门店列表 |
| | 商家->星盘 | taobao.qimen.storeitem.query | 查询门店所关联的线上商品列表 |
| 库存信息 | 商家->星盘 | taobao.qimen.storeinventory.iteminitial | 全量更新电商仓或门店库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemupdate | 增量更新门店或电商仓库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemquery | 查询门店或电商仓库存 |
| | 商家->星盘 | taobao.qimen.storeinventory.itemadjust | 调整所占用的门店/电商仓库存 |
| 作业流转 | 星盘->商家 | taobao.qimen.omniorder.allocate | 启动分单或重新分单 |
| | 商家->星盘 | taobao.omniorder.allocatedinfo.sync | ISV 同步派单结果到星盘订单网关 |
| | 星盘->商家 | taobao.qimen.omniorder.allocatedinfo.get | 派单状态查询 |
| | POS->星盘 | taobao.omniorder.store.accpeted | 启动派单 (ISV 门店端) |
| | 星盘->分单 | taobao.qimen.omniorder.store.accpeted | 星盘通知 POS 已接单状态 |
| | POS ->星盘 | taobao.omniorder.store.refuse | 门店拒单 (ISV 门店端) |
| | 星盘->分单 | taobao.qimen.omniorder.store.refused | 星盘通知 POS 已拒单状态 |
| | POS->星盘 | taobao.omniorder.store.consigned | 门店发货 (ISV 门店端) |
| 星盘->分单 | taobao.qimen.omniorder.store.consigned | 星盘通知 POS 发货状态 | |

关键流程



注意事项：

ISV 在系统实现时，需要确保先调用天猫 B2C 的发货接口，再调用门店发货通知

“taobao.omniorder.store.consigned” 接口。

4 库存管理要求

4.1 库存类别

物理库存：ERP 或 WMS 系统中记录的门店或仓库的实际库存。

渠道确定库存：门店专供指定线上渠道（天猫）的确定销售库存。

渠道共享库存：线上线下共享的可销库存;线上多个渠道共享的可销售库存。

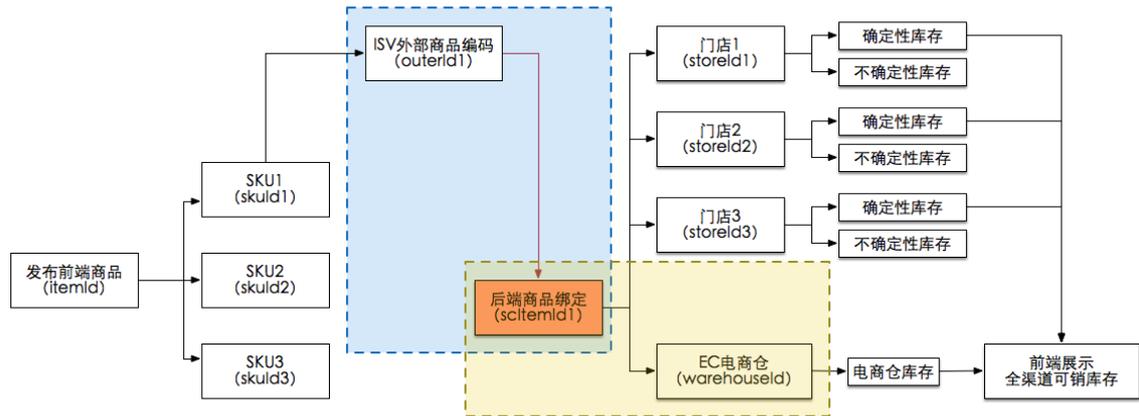
4.2 库存商品关联

第 2 章列出的全渠道参考实现中，给出了奇门库存同步接口，主要包括库存全量更新、库存增量更新、库存查询。这些库存同步接口，提供了关联前端商品（宝贝）和后端商品（货品）两种方式。

通常情况下，库存对应的是货品；因为对于同一个货品，根据销售渠道不同，其宝贝的标示可以是不同的。也就是说，货品与宝贝之间是一对多的关系。这也是为什么，天猫货品中心将库存挂到了货品上，宝贝是无法关联库存的。这就要求，商家需要自己维护宝贝-货品映射关系（调用“taobao.scitem.add”添加货品、同时调用“taobao.scitem.map.add”做宝贝和货品的关联），并且通过货品进行库存上传和更新。这种方式虽然具有很好的灵活性和扩展性，但是这种方式会增加较多的实施成本。

为此，对于宝贝和货品一一对应的商家，可以采用通过宝贝+SKU 同步库存的方式。这种情况下，系统将帮助商家创建货品并建立宝贝和货品的关联关系。这样，商家就可以通过宝贝+SKU 的方式进行库存同步了。

这种情况下，商家系统的改造成本比较低，但是，需要确保宝贝和货品的一一对应关系。这样将对货品售卖的灵活性造成较大影响，例如无法做到多个店铺在售宝贝的库存共享。宝贝-库存映射关系参见下图：



4.3 避免超卖

对于使用确定性库存而言，超卖问题并不突出，只要做到库存变更时及时调用库存更新接口更新即可。

对于非确定性库存，存在门店线下售卖与电商线上售卖资源竞争的问题。在这种情况下，要求全渠道中台采取有效的机制避免超卖的发生。举例来说，星盘采用非确定库存和确定性库存双占的方式避免因非确定库存占用失效导致的超卖。在非确定性库存被线上真正售卖后（接单确认），才会释放对确定性库存的占用。

在默认情况下，天猫会扣减具有最大确定性库存的仓或门店中的库存。星盘和 ISV 在分单时，如果分配到的仓或门店与天猫所扣减的仓或门店不一致，需调用库存调整接口（`taobao.qimen.storeinventory.itemadjust`）进行库存的调整，以确保库存的准确性。

5 作业系统指标要求

5.1 正向作业响应指标

拍下付款 — [分单效率](#) — 派单到门店 — [接单效率](#) — 门店已接单 — [发货效率](#) — 门店已发货 —

5.1.1 分单指标

平均分单时间：统计周期内所有子订单分单时间平均值（分单时间 = 子订单完成派单时间戳 - 拍下付款时间戳）。

最大分单时长：在统计周期内最大分单时间。

分单效率：是衡量分单作业系统效率的核心指标。分单效率通过平均分单时间衡量，平均分单时间越长，分单效率越低。

分单作业要求：待定。

分单效率要求：待定。

5.1.2 接单指标

平均接单时间：统计周期内所有接单操作完成的平均值（接单时间 = 门店已接单 - 派单到门店）。

最大接单时长：在统计周期内最大接单时间。

接单效率：是衡量门店接单响应度的核心指标。接单效率通过平均接单时间衡量，平均接单时间越长，接单效率越低。

接单效率要求：待定。

5.1.3 发货指标

平均发货时间：统计周期内所有发货操作完成的平均值（发货时间 = 门店已发货时间戳 - 门店已接单时间戳）。

最大发货时长：在统计周期内最大等待发货时间。

发货效率：是衡量门店打标发货速度的核心指标。发货效率通过平均发货时间衡量，平均发货时间越长，发货效率越低。

发货效率要求：待定。

5.1.4 物流指标

平均送货时间：统计周期内所有物流时间的平均值（物流时间 = 签收时间戳 - 门店已发货时间戳）。

最大物流时长：在统计周期内最大送达时间。

物流效率：是衡量物流质量的核心指标。物流效率通过平均送货时间衡量，平均送货时间越长，物流效率越低。

物流效率要求：待定。

5.2 成熟度指标

5.2.1 订单接单率

门店接单次数占所有派送至门店订单的百分比。门店接单率越高，说明门店发货整体成熟度越高。

5.2.2 门店派单率

分单到门店的全渠道订单占所有全渠道订单的百分比。门店派单率越高，说明整体门店发货能力越强。

5.2.3 门店接单率

接单门店占有派单门店的比率。门店接单率越高，说明门店发货能力越均衡。

5.3 逆向作业响应指标

5.4 关键状态回传时间要求

| 接口分类 | 状态 | 接口名 | 要求时间 |
|------|----------------------------|-------------------------------------|------|
| 状态信息 | X_ALLOCATION_NOTIFIED 通知配货 | taobao.qimen.orderStatus.sync | <5m |
| | X_SHOP_ALLOCATION 已接单 | taobao.qimen.orderStatus.sync | <5m |
| | X_ALLOCATION_NOTIFIED 通知配货 | taobao.omniorder.allocatedinfo.sync | <5m |
| | X_SHOP_ALLOCATION 已接单 | taobao.omniorder.store.accpeted | <5m |
| | X_SHOP_HANDLED 已发货 | taobao.omniorder.store.consigned | <5m |
| | X_SHOP_DENY 已拒单 | taobao.omniorder.store.refuse | <5m |

5.5 指标校验

5.5.1 状态回传不及时的影响

状态信息回传不及时会造成以下影响：

- 1、无法及时展示门店发货物流信息。
- 2、逆向流程（退款退货场景）会存在一定问题。
- 3、长期信息回传不及时将会导致全渠道打标和加权失效。

5.5.2 及时回传状态信息

系统提供了全链路压测工具，用来帮助商家评估和提升状态信息的回传效率。

6 接口定义

6.1 门店上传及商品关联

6.1.1 门店新增接口

接口说明：

商家在 ERP 等系统中调用该接口，新建线下门店，并将其与线上店铺关联。新建门店成功后，会返回一个门店 ID，是一个全局 ID，后续所有的接口都用此 ID 通信。其中门店主营类目字段，需要通过 4.1.2 的“门店类目获取接口”获取所有类目候选项，然后上传对应的类目编号。

API 名称：taobao.qimen.store.create。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
<storeCode>ERP 系统中 门店编码，string ( 50 )，必填</storeCode>
<shopId>需要关联的线上店铺 ID，long</shopId>
<companyName>商户名称（例如，甘其食），string ( 50 )，必填</companyName>
<storeName>门店名称（例如，文一西路淘宝店），string ( 50 )，必填</storeName>
<storeType>类型，string ( 50 )，NORMAL=普通门店，MALL = 商城，MALL_SHOP = 店中店，
LIGHT_SHOP = 淘小铺，HOSPITAL = 阿里健康（医院），DEPARTMENT = 阿里健康（医院科室），
WAREHOUSE=仓库，必填</storeType>
<mainCategory>门店主营类目，int，必填</mainCategory>
<startTime>开始营业时间，string (19)，HH:MM，必填</startTime>
<endTime>关闭营业时间，string (19)，HH:MM，必填</endTime>
<address>
  <region>区域（如华南区），string(50)</region>
    <province>省份，string ( 50 )，必填</province>
    <city>城市，string ( 50 )，必填</city>
  <area>区域，string ( 50 )，必填</area>
  <town>村镇，string ( 50 ) </town>
  <detailAddress>详细地址，string ( 50 )，必填</detailAddress>
  <countryCode>国家二字码，string ( 50 ) </countryCode>
</address>
```

```
<storeStatus>门店状态，string ( 50 ) ， HOLD=暂停营业 ， CLOSE=关店 ， NORMAL=正常
</storeStatus >
<storeDescription>商户介绍，string ( 500 ) </storeDescription>
<storeKeeper>
  <name>姓名，string ( 50 ) </name>
  <tel>固定电话，string ( 50 ) ， 必填</tel>
  <mobile>移动电话，string ( 50 ) ， 必填</mobile>
  <fax>传真，string ( 50 ) </fax>
  <zipCode>邮编，string ( 50 ) </zipCode>
</storeKeeper>
  <remark>备注，string ( 500 ) </remark>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <storeId>门店 ID<storeId>
  <code>响应码</code>
  <message>响应信息</message>
</response>
```

6.1.2 门店类目获取接口

接口说明：

调用该接口，获取门店类目（如需可视化，然后设计成如下样式），其中每个最末端类目对应一个 id，每当商家选定一个主营类目时，ISV 只需将对应的 ID 传给奇门即可。**建议：**

每次新增门店时，实时拉取下门店类目，保证类目的同步。

* 主营类目

API 名称：taobao.qimen.storecategory.get。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <remark>备注，string ( 500 ) </remark>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <storeCategory>类目的 JSON 格式文件</storeCategory>
  <code>响应码</code>
  <message>响应信息</message>
</response>
```

6.1.3 门店更新接口

接口说明：

调用该接口，更新门店信息

API 名称：taobao.qimen.store.update。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <storeId>门店 ID，long，必填</storeId>
  <storeCode>ERP 系统中 门店编码，string ( 50 )，必填</storeCode>
  <shopId>需要关联的线上店铺 ID，long，必填</shopId>
  <companyName>商户名称，string ( 50 )，必填</companyName>
  <storeName>门店名称，string ( 50 )，必填</storeName>
  <storeType>类型，string ( 50 )，NORMAL=普通门店，MALL = 商城，MALL_SHOP = 店中店，
  LIGHT_SHOP = 淘小铺，HOSPITAL = 阿里健康（医院），DEPARTMENT = 阿里健康（医院科室），
  WAREHOUSE=仓库，必填</storeType>
  <mainCategory>门店主营类目，int，必填</mainCategory>
  <startTime>开始营业时间，string (19)，HH:MM，必填</startTime>
  <endTime>关闭营业时间，string (19)，HH:MM，必填</endTime>
  <address>
    <region>区域（如华南区），string(50)</region>
    <province>省份，string ( 50 )，必填</province>
    <city>城市，string ( 50 )，必填</city>
    <area>区域，string ( 50 )，必填</area>
    <town>村镇，string ( 50 )</town>
    <detailAddress>详细地址，string ( 50 )，必填</detailAddress>
    <countryCode>国家二字码，string ( 50 )</countryCode>
  </address>
  <storeStatus>门店状态，string ( 50 )，HOLD=暂停营业，CLOSE=关店，NORMAL=正常
</storeStatus >
```

```
<storeDescription>商户介绍，string ( 500 ) </storeDescription>
<storeKeeper>
  <name>姓名，string ( 50 ) </name>
  <tel>固定电话，string ( 50 ) ， 必填</tel>
  <mobile>移动电话，string ( 50 ) ， 必填</mobile>
  <fax>传真，string ( 50 ) </fax>
  <zipCode>邮编，string ( 50 ) </zipCode>
</storeKeeper>
<remark>备注，string ( 500 ) </remark>
</request>
```

出参规范:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
</response>
```

6.1.4 门店删除接口

接口说明：

调用该接口，批量删除线下门店数据。

API 名称：taobao.qimen.store.delete。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <storeId>门店 ID，long， 必填<storeId>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
</response>
```

6.1.5 门店信息查询接口

接口说明：

调用该接口，查询门店相关信息。

API 名称：taobao.qimen.store.query。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <storeId>门店 ID，long，必填<storeId>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
<storeCode>ERP 系统中 门店编码，string ( 50 ) </storeCode>
<shopId>需要关联的线上店铺 ID，long</shopId>
<companyName>商户名称，string ( 50 ) </companyName>
<storeName>门店名称，string ( 50 ) </storeName>
<storeType>类型，string ( 50 )，NORMAL=普通门店，MALL = 商城，MALL_SHOP = 店中店，
LIGHT_SHOP = 淘小铺，HOSPITAL = 阿里健康（医院），DEPARTMENT = 阿里健康（医院科室），
WAREHOUSE=仓库 </storeType>
<mainCategory>门店主营类目，int </mainCategory>
<startTime>开始营业时间，string (19)，HH:MM </startTime>
<endTime>关闭营业时间，string (19)，HH:MM </endTime>
<address>
  <region>区域（如华南区），string(50)</region>
  <province>省份，string ( 50 ) </province>
  <city>城市，string ( 50 ) </city>
  <area>区域，string ( 50 ) </area>
  <town>村镇，string ( 50 ) </town>
  <detailAddress>详细地址，string ( 50 ) </detailAddress>
  <countryCode>国家二字码，string ( 50 ) </countryCode>
</address>
<storeStatus>门店状态，string ( 50 )，HOLD=暂停营业，CLOSE=关店，NORMAL=正常
</storeStatus >
<storeDescription>商户介绍，string ( 500 ) </storeDescription>
<storeKeeper>
  <name>姓名，string ( 50 ) </name>
  <tel>固定电话，string ( 50 ) </tel>
```

```
<mobile>移动电话, string ( 50 ) </mobile>
<fax>传真, string ( 50 ) </fax>
<zipCode>邮编, string ( 50 ) </zipCode>
</storeKeeper>
<remark>备注, string ( 500 ) </remark>
</response>
```

6.1.6 商品关联绑定接口

接口说明：

调用该接口，将线上商品和线下门店“新建/删除”关联。这里的线上。每次只能单个商品关联多个门店，门店上限 200。

API 名称：taobao.qimen.itemstore.banding。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
<actionType>操作类型, string ( 50 ), ADD=新建, DELETE=删除, 必填</actionType>
<itemId>线上商品 ID, long, 必填</itemId>
<storeIds>
    <storeId>门店 ID, long, 必填<storeId>
    ...
</storeIds>
<remark>备注, string ( 500 ) </remark>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
    <flag>success|failure</flag>
    <code>响应码</code>
    <message>响应信息</message>
</response>
```

6.1.7 商品关联门店查询接口

接口说明：

调用该接口，查询线上商品所关联的门店列表。

API 名称：taobao.qimen.itemstore.query。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <itemId>线上商品 ID , long , 必填</itemId>
  <page>当前页数 ,long ,非必填 ,默认值为 1 ,返回第一页数据 ,每页默认记录 100 条</page>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success</flag>
  <code>响应码</code>
  <storeIds>
    <storeId>门店 ID , long <storeId>
    ...
  </storeIds>
  <message>响应信息</message>
</response>
```

6.1.8 门店关联商品查询接口

接口说明：

调用该接口，查询某门店所关联的线上商品列表。

API 名称：taobao.qimen.storeitem.query。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <storeId>门店 ID , long , 必填</storeId>
  <page>当前页数 ,long ,非必填 ,默认值为 1 ,返回第一页数据 ,每页默认记录 100 条</page>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success</flag>
```

```
<code>响应码</code>
<itemIds>
  <itemId>线上商品 ID , long</itemId>

  ...
</ itemIds >
<message>响应信息</message>
</response>
```

6.2 库存通(后端商品)

6.2.1 门店库存初始化接口

接口说明：

全量初始化门店库存，该接口一次可以初始化多个门店(store)的多个商品(item/sku)的多种类型(type)库存。**确定性的库存只能初始化一次**

API 名称：taobao.qimen.storeinventory.initial

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <userId>淘宝账号 ID , long , 必填</userId>
  <operationTime>操作时间 , string (19) , YYYY-MM-DD HH:MM:SS , 必填</operationTime>
  <stores>
    <store>
      <warehouseType>库存来源的类型 , string(50) , WAREHOUSE=电商仓 , STORE=门店 , 必填</warehouseType>
      <warehouseId>门店 ID( 商户中心 ) 或 电商仓 ID , string(50) , 必填</warehouseId>

      <storeInventories>
        <storeInventory>
          <billNum>流水号 , 用于幂等操作 , string(50) , 必填</billNum>
          <scItemId>淘宝后端商品 id , long , 必填</scItemId>
          <outerId>ISV 系统中商品编码 , string(50)</outerId>
          <inventoryType>库存类型 , string(50) , CERTAINTY=确定性库存 , UNCERTAINTY=不确定性库存 , 必填</inventoryType>
          <quantity>对应类型的库存数量 , long , 必填</quantity>
```

```
        </storeInventory>
        ...
    </storeInventories>
</store>
...
</stores>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
  <errorDescriptions>
    <error>
      <description>错误描述，String(50)</description>
      <failedBillNum>处理失败的流水号（有多个时，用逗号分隔），string(50)</failedBillNum>
    </error>
  </errorDescriptions>
</response>
```

6.2.2 门店库存更新接口

接口说明：

增量更新门店库存，该接口一次可以同时更新多个门店(store)的多个商品(item/sku)的多种类型(type)库存。

API 名称：taobao.qimen.storeinventory.update

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <userId>淘宝账号 ID，long，必填</userId>
  <operationTime>操作时间，string (19)，YYYY-MM-DD HH:MM:SS，必填</operationTime>
  <stores>
    <store>
      <warehouseType>库存来源的类型，string(50)，WAREHOUSE=电商仓，STORE=门店，必填</warehouseType>
      <warehouseId>门店 ID(商户中心) 或 电商仓 ID，string(50)，必填</warehouseId>
    </store>
  </stores>
</request>
```

```

    <storeInventory>
      <billNum>流水号, 用于幂等操作, string(50) , 必填</billNum>
      <billType>单据类型, String(50), TAOBAO_INCLUDED=包括淘系销售及其他渠道销售,
      TAOBAO_EXCLUDED=只含其他渠道销售, 不包括淘系销售, BALANCE=盘点(包括入库、出
      库、盘亏盘盈), 必填</billType>
      <scItemId>淘宝后端商品 id, long, 必填</scItemId>
      <outerId>ISV 系统中商品编码, string(50)</outerId>
      <inventoryType>库存类型, string(50), CERTAINTY=确定性库存,
      UNCERTAINTY=不确定性库存, 必填</inventoryType>
      <quantity>本次调整库存数量(新增时填正数, 扣减时填负数), int, 必填</quantity>
      <finalQuantity>调整后库存数量, long</finalQuantity>
    </storeInventory>
    ...
  </storeInventories>
</store>
...
</stores>
</request>

```

出参规范

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
  <errorDescriptions>
    <error>
      <description>错误描述, String(50)</description>
      <failedBillNum>处理失败的流水号(有多个时, 用逗号分隔), string(50)</failedBillNum>
    </error>
  </errorDescriptions>
</response>

```

6.2.3 门店库存查询接口

接口说明：

查询门店库存, 该接口一次可以同时查询多个门店(store)的多个商品(item/sku)的多种类型(type)的库存。

API 名称：taobao.qimen.storeinventory.query

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <userId>淘宝账号 ID , long , 必填</userId>
  <stores>
    <store>
      <warehouseType>库存来源 , string(50) , WAREHOUSE=电商仓 , STORE=门店 , 必填</warehouseType>
      <warehouseId>门店 ID(商户中心) 或 电商仓 ID , string(50) , 必填</warehouseId>
      <storeInventories>
        <storeInventory>
          <scItemId>淘宝后端商品 id , long , 必填</scItemId>
          <outerId>ISV 系统中商品编码 , string(50)</outerId>
        </storeInventory>
        ...
      </storeInventories>
    </store>
    ...
  </stores>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
  <stores>
    <store>
      <warehouseType>库存来源 , string(50) , WAREHOUSE=电商仓 , STORE=门店
    </warehouseType>
      <warehouseId>门店 ID(商户中心) 或 电商仓 ID , string(50) </warehouseId>
      <storeInventories>
        <storeInventory>
          <scItemId>淘宝后端商品 id , long</scItemId>
          <outerId>ISV 系统中商品编码 , string(50)</outerId>
          <quantityDetails>
            <quantityDetail>
              <inventoryType>库存类型 , string(50) , CERTAINTY=确定性库存 ,
              UNCERTAINTY=不确定性库存</inventoryType>
              <quantity>当前类型库存数量 , long</quantity>
            </quantityDetail>
          </quantityDetails>
        </storeInventory>
      </storeInventories>
    </store>
  </stores>
</response>
```

```
        </storeInventory>
        ...
    </storeInventories>
</store>
...
</stores>
</response>
```

6.2.4 库存占用调整接口

接口说明：

当第三方系统出现分单结果和天猫货品中心分单结果不一致时，需要调用此接口同步分单消息给天猫货品中心，调整之前占用的门店/电商仓库存。

API 名称：taobao.qimen.storeinventory.adjust

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <userId>淘宝账号 ID，long，必填</userId>
  <operationTime>操作时间，string(19)，YYYY-MM-DD HH:MM:SS，必填</operationTime>
  <inventoryAdjustInfo>
    <billNum>流水号，用于幂等操作，string(50)，必填</billNum>
    <tbTradeOrder>淘宝订单号，string(50)，必填</tbTradeOrder>
    <tbSubTradeOrder>淘宝子订单号，string(50)，必填</tbSubTradeOrder>
    <targetWarehouseType>调整到的目标仓储类型，string(50)，STORE=门店，
    WAREHOUSE=电商仓，必填</targetWarehouseType>
    <targetWarehouseId>需要调整到的目标门店 ID(商户中心) 或电商仓 ID，
    string(50)，必填</targetWarehouseId>
    <scItemId>淘宝后端商品 id，long，必填</scItemId>
    <outerId>ISV 系统中商品编码，string(50)</outerId>
    <inventoryType>库存类型，string(50)，CERTAINTY=确定性库存，UNCERTAINTY=不确定性库存，必填</inventoryType>
    <quantity>需调整的库存数量，long，必填</quantity>
  </inventoryAdjustInfo>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
</response>
```

6.3 库存通（前端商品）

6.3.1 库存初始化接口

接口说明：

全量更新电商仓或门店库存，该接口一次可以初始化多个门店(或电商仓)的多个商品的多种类型库存。确定性的库存只能初始化一次

API 名称：taobao.qimen.storeinventory.iteminitial

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <userId>淘宝账号 ID，long，必填</userId>
  <operationTime>操作时间，string(19)，YYYY-MM-DD HH:MM:SS，必填</operationTime>
  <stores>
    <store>
      <warehouseType>库存来源的类型，string(50)，WAREHOUSE=电商仓，STORE=门店，必填</warehouseType>
      <warehouseId>门店 ID( 商户中心 ) 或 电商仓 ID，string(50)，必填</warehouseId>
      <storeInventories>
        <storeInventory>
          <billNum>单据流水号，用于幂等操作，string(50)，必填</billNum>
          <itemId>淘宝前端商品 id，long，必填</itemId>
          <outerId>ISV 系统中商品编码，string(50)，必填</outerId>
          <skuId>商品的 SKU 编码，long，有 sku 的必填</skuId>
          <inventoryType>库存类型，string(50)，CERTAINTY=确定性库存，UNCERTAINTY=不确定性库存，必填</inventoryType>
          <quantity>对应类型的库存数量(正数)，int，必填</quantity>
        </storeInventory>
      </storeInventories>
    </store>
  </stores>
</request>
```

```
...
    </storeInventories>
</store>
...
</stores>
</request>
```

出参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
  <errorDescriptions>
    <error>
      <description>错误描述，String(50)</description>
      <failedBillNum>处理失败的流水号（有多个时，用逗号分隔），string(50)</failedBillNum>
    </error>
  </errorDescriptions>
</response>
```

6.3.2 库存增量更新接口

接口说明：

增量更新门店或电商仓库存,该接口一次可以同时增量更新多个门店的多个商品的非确定性库存.

API 名称：taobao.qimen.storeinventory.itemupdate。

入参规范：

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <userId>淘宝账号 ID，long，必填</userId>
  <operationTime>操作时间，string (19)，YYYY-MM-DD HH:MM:SS，必填</operationTime>
  <stores>
    <store>
      <warehouseType>库存来源的类型，string(50)，WAREHOUSE=电商仓，STORE=门店，必填</warehouseType>
      <warehouseId>门店 ID(商户中心) 或 电商仓 ID，string(50)，必填</warehouseId>
```

```

    <storeInventories>
    <storeInventory>
    <billNum>单据流水号，用于幂等操作，string(50)，必填</billNum>
    <billType>单据类型，String(50)，TAOBAO_INCLUDED=包括淘系销售及其他
    渠道销售，TAOBAO_EXCLUDED=只含其他渠道销售，不包括淘系销售，
    BALANCE=盘点（包括入库、出库、盘亏盘盈），必填</billType>
    <itemId>淘宝前端商品 id，long，必填</itemId>
    <outerId>ISV 系统中商品编码，string(50)，必填</outerId>
    <skuId>商品的 SKU 编码，long，有 sku 的必填</skuId>
    <inventoryType>库存类型，string(50)，CERTAINTY=确定性库存，
    UNCERTAINTY=不确定性库存，必填</inventoryType>
    <quantity>本次调整库存数量（新增时填正数，扣减时填负数），int，必填</quantity>
    <finalQuantity>调整后库存数量，int</finalQuantity>
    </storeInventory>
    ...
    </storeInventories>
  </store>
  ...
</stores>
</request>

```

出参规范

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
  <errorDescriptions>
    <error>
      <description>错误描述，String(50)</description>
      <failedBillNum>处理失败的流水号（有多个时，用逗号分隔），string(50)</failedBillNum>
    </error>
  </errorDescriptions>
</response>

```

6.3.3 库存查询接口

接口说明

查询门店或电商仓库存，该接口一次可以同时查询多个门店或电商仓的多个商品的多种类型的库存。

API 名称 : taobao.qimen.storeinventory.itemquery。

入参规范

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <userId>淘宝账号 ID , long , 必填</userId>
  <stores>
    <store>
      <warehouseType>库存来源 , string(50) , WAREHOUSE=电商仓 , STORE=门店 , 必填</warehouseType>
      <warehouseId>门店 ID(商户中心) 或 电商仓 ID , string(50) , 必填</warehouseId>
      <storeInventories>
        <storeInventory>
          <itemId>淘宝前端商品 id , long , 必填</itemId>
          <outerId>ISV 系统中商品编码 , string(50) , 必填</outerId>
          <skuId>商品的 SKU 编码 , long , 有 sku 的必填</skuId>
        </storeInventory>
        ...
      </storeInventories>
    </store>
    ...
  </stores>
</request>
```

出参规范 :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
  <stores>
    <store>
      <warehouseType>库存来源 , string(50) , WAREHOUSE=电商仓 , STORE=门店
    </warehouseType>
    <warehouseId>门店 ID(商户中心) 或 电商仓 ID , string(50) </warehouseId>
    <storeInventories>
      <storeInventory>
        <itemId>淘宝前端商品 id , long</itemId>
        <outerId>ISV 系统中商品编码 , string(50)</outerId>
        <skuId>商品的 SKU 编码 , long</skuId>
      <quantityDetails>
        <quantityDetail>
```

```

        <inventoryType>库存类型，string(50)，CERTAINTY=确定性库存，
        UNCERTAINTY=不确定性库存</inventoryType>
        <quantity>当前类型库存数量，int</quantity>
    </quantityDetail>
</quantityDetails>
</storeInventory>
...
</storeInventories>
</store>
...
</stores>
</response>

```

6.3.4 库存占用调整接口

接口说明：

当第三方系统出现分单结果和天猫货品中心分单结果不一致时，需要调用此接口同步分单消息给天猫货品中心，调整之前占用的门店/电商仓库存。

API 名称：taobao.qimen.storeinventory.itemadjust。

入参规范：

```

<?xml version="1.0" encoding="utf-8"?>
<request>
    <userId>淘宝账号 ID，long，必填</userId>
    <operationTime>操作时间，string(19)，YYYY-MM-DD HH:MM:SS，必填</operationTime>
    <inventoryAdjustInfo>
        <billNum>流水号，用于幂等操作，string(50)，必填</billNum>
        <tbTradeOrder>淘宝订单号，string(50)，必填</tbTradeOrder>
        <tbSubTradeOrder>淘宝子订单号，string(50)，必填</tbSubTradeOrder>

        <targetWarehouseType>调整到的目标仓储类型，string(50)，STORE=门店，
        WAREHOUSE=电商仓，必填</targetWarehouseType>

        <targetWarehouseId>需要调整到的目标门店 ID(商户中心) 或电商仓 ID，
        string(50)，必填</targetWarehouseId>

        <itemId>淘宝前端商品 id，long，必填</itemId>
        <outerId>ISV 系统中商品编码，string(50)，必填</outerId>

```

```

    <skuId>商品的 SKU 编码，long, 有 sku 的必填</skuId>
    <inventoryType>库存类型，string(50)，CERTAINTY=确定性库存，UNCERTAINTY=不确定性库存，必填</inventoryType>
    <quantity>需调整的库存数量，long，必填</quantity>
  </inventoryAdjustInfo>
</request>

```

出参规范：

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
</response>

```

PS：4.2 库存通（后端商品）方案 1 与 4.3 库存通（前端商品）方案 2 的区别；

对接前，一定要理解 scitem 和 item

> 两种方案对比

| 序号 | 商家自己维护“商品-货品”，通过“货品”来同步库存（方案1） | 平台代理商商家维护“商品-货品”，通过“商品”来同步库存（方案2） |
|-------|---|---|
| 优点 | 后续管理库存会比较方便，而且可以扩展很多其他的业务 | 商家和ISV实施成本会比较低 |
| 缺点及风险 | 商家和ISV实施成本会比较高 | 商家自己的货品编码必须保障唯一性，如果有因为不唯一导致前台宝贝单号售卖，只能由商家自行承担后果； 后续只能通过前台宝贝下的SKU去维护库存，无法进行后端货品分仓库的库存维护； 无法通过货品作为天猫和商家之间全渠道货品销售预测类协同； 无法做期货货品&期货订单管理； 无法做同一后端货品在阿里体系上多市场多店铺对应的前台多个宝贝的库存共享； 无法做单品与组合商品之间库存共享的组合商品发布管理； |
| 改造事项 | 1.到天猫库存中心，创建“电商仓”，拿到电商仓ID。电商仓库存支持页面后台管理（全新的后台），门店库存暂不支持页面后台管理，只能通过接口更新 2.为所有的“商品”创建对应的“货品”，并维护“商品-货品”映射关系 3.门店/电商仓库存都需按“货品”的维度上传和更新 | 1.到天猫库存中心，创建“电商仓”，拿到电商仓ID。电商仓库存支持页面后台管理（全新的后台），门店库存暂不支持页面后台管理，只能通过接口更新 2.门店/电商仓库存都按“商品”的维度上传和更新 |

6.4 订单全链路同步

接口说明：

商家中台，将订单下行分单情况同步到线上。**此接口应用于商家系统自己分单（非阿里**

星盘分单）的模式下更新订单状态。

API名称：taobao.qimen.orderStatus.sync。

```

<?xml version=" 1.0" encoding=" utf-8" ?>

```

```
<request>

  <sellerCode>卖家淘宝 ID , String , 必填</sellerCode>

  <parentOrderCode>淘系主订单号 , Long , 必填 ( 单订单情况下主子订单号相
同 ) </parentOrderCode>

  <orderCodes>

    <orderCode>淘系子订单号 , Long , 必填</orderCode>

  </orderCodes>

  <storeId>目标门店的商户中心门店编码, Long , 条件必填 ( 门店发货、门店自提
相关状态场景下 ) </storeId>

  <type>业务类型 , String , 条件必填( 门店发货、门店自提相关状态下 )( 枚举值 :
FAHUO、ZITI ) </type>

  <status>订单状态 , String , 必填</status>

  <operator>操作人 , String , 必填</operator>

  <actionTime> 事件发生时间 , String ( yyyy-MM-dd HH:mm:ss ) , 必填

</actionTime>

</request>
```

出参规范 :

```
<?xml version=" 1.0" encoding=" utf-8" ?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
</response>
```

状态说明 :

通用状态 :

| 描述 | 状态 | 是否必传 |
|-------|---------------------|------|
| 系统已接单 | X_TO_SYSTEM | 是 |
| 已客审 | X_SERVICE_AUDITED | 否 |
| 已财审 | X_FINANCE_AUDITED | 否 |
| 已打拣货单 | X_SORT_PRINTED | 否 |
| 已打发货单 | X_SEND_PRINTED | 否 |
| 已打物流单 | X_LOGISTICS_PRINTED | 否 |
| 已拣货 | X_SORTED | 否 |
| 已验货 | X_EXAMINED | 否 |
| 已打包 | X_PACKAGED | 否 |
| 已称重 | X_WEIGHED | 否 |
| 已出库 | X_OUT_WAREHOUSE | 否 |

门店发货场景状态：

| 描述 | 状态 | 是否必传 |
|--------|-----------------------|------|
| 通知门店配货 | X_ALLOCATION_NOTIFIED | 是 |
| 门店已接单 | X_SHOP_ALLOCATION | 是 |

门店自提场景状态：

| 描述 | 状态 | 是否必传 |
|--------|-----------------------|------|
| 通知门店配货 | X_ALLOCATION_NOTIFIED | 是 |
| 门店确认有货 | X_COMMODITY_CONFIRM | 是 |

6.5 订单状态更新接口

接口说明：

星盘和 ISV，可以通过此接口，来更新订单状态。**此接口应用于使用阿里星盘分单，且使用商家系统（非阿里掌柜）接单/拒单的模式下更新订单状态。**

API 名称：taobao.qimen.orderStatus.update。

```
<?xml version=" 1.0" encoding=" utf-8" ?>
```

```
<request>
```

```
    <sellerCode>卖家淘宝 ID，String，必填</sellerCode>
```

```
    <parentOrderCode>淘系主订单号，Long，必填（单订单情况下主子订单号相同）</parentOrderCode>
```

```
    <orderCodes>
```

```
        <orderCode>淘系子订单号，Long，必填</orderCode>
```

```
    </orderCodes>
```

```
    <storeId>目标门店的商户中心门店编码，Long，条件必填（门店发货、门店自提相关状态场景下）</storeId>
```

```
    <type>业务类型，String，条件必填（门店发货、门店自提相关状态下）（枚举值：FAHUO、ZITI）</type>
```

```
    <status>订单状态，String，必填</status>
```

```
    <operator>操作人，String，必填</operator>
```

```
    <actionTime>事件发生时间，String（yyyy-MM-dd HH:mm:ss），必填
```

```
</actionTime>
```

```
</request>
```

出参规范：

```
<?xml version=" 1.0" encoding=" utf-8" ?>
<response>
  <flag>success|failure</flag>
  <code>响应码</code>
  <message>响应信息</message>
</response>
```

门店发货状态 JST_ASTROLABE_STORE_DELIVER_ORDER_STATE 中各状态值含义（黑色

字体表示由阿里星盘更新的状态，绿色字体表示由商家系统更新的状态）：

| 发起方 | 场景 | 状态 | Value | 描述 |
|-----|------|-------------|-----------------------|---|
| 星盘 | 正向流程 | 通知门店/仓配货 | X_ALLOCATION_NOTIFIED | |
| ISV | | 门店已接单 | X_SHOP_ALLOCATION | |
| ISV | | 门店拒绝接单 | X_SHOP_DENY | |
| ISV | | 已发货 | X_SHOP_HANDLED | 与订单状态同步，已发货 |
| 星盘 | 售中退 | 发起退款/退货申请 | X_SHOP_CANCEL | 向门店发起售中退货请求 |
| ISV | | 货未发出 | X_SHOP_CANCEL_CONFIRM | 门店店员会将货品退回货架，线上客服可以退款（售中退） |
| ISV | | 货已发出 | X_SHOP_CANCEL_DENIED | 门店还未将 www 物流信息填入系统，但货品已发出,此时线上不能退款(售中退) |
| 星盘 | 售后退 | 发起售后退款/退货申请 | X_HANDLED_CANCEL | 售后退请求发给门店 |
| ISV | | 门店收到退货 | X_MATCHED | 售后退货已收到 |

门店自提状态 JST_ASTROLABE_STORE_DELIVER_ORDER_STATE 中各状态值含义：

| 发起方 | 场景 | 状态 | Value | 描述 |
|-----|------|--------|-----------------------|----------|
| 星盘 | 正向流程 | 通知门店配货 | X_ALLOCATION_NOTIFIED | 等待门店店员接单 |
| ISV | | 门店确认有货 | X_COMMODITY_CONFIRM | |

| | | | | |
|-----|-----|----------------|-----------------------|----------------|
| ISV | | 启动无货调拨（以后版本支持） | X_COMMODITY_TRANSER | 门店拒单，启动无货调拨流程 |
| ISV | | 确认调货成功（以后版本支持） | X_TRANSFER_SUCCESS | 门店收到调货 |
| 星盘 | 售中退 | 通知取消发货 | X_SHOP_CANCEL | 售中退已通知门店 |
| ISV | | 门店确认退货请求 | X_SHOP_CANCEL_CONFIRM | 店员阿里掌柜点击‘我知道了’ |
| 星盘 | 售后退 | 发起售后退款/退货申请 | X_HANDLED_CANCEL | 售后退请求发给门店 |
| ISV | | 门店收到退货 | X_MATCHED | 售后退货已收到 |

6.6 星盘开放接口 2.0

6.6.1 派单/重新分单

taobao.qimen.omniorder.allocate

API 基本信息

- API 名称: 全渠道分单接口/重新分单
- API 描述:

如果该订单是第一次分单，则星盘调用 ISV 分单接口进行分单。ISV 返回的 AppKey 列表(逗号分隔)表示希望订单推送系统需要写入的 RDS。

如果该订单是已经调用过该接口进行过分单，则会进行重新分单，需要注意的是在这种情况下，如果改订单状态为已经发货，则接口将会返回报错。

API 请求入参

| 名称 | 类型 | 是否必 | 描述 | 示例值 |
|----|----|-----|----|-----|
|----|----|-----|----|-----|

| | | 须 | | |
|-------------|--------|---|------------|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| seller_id | number | 是 | 卖家的 ID | 21230 |
| seller_nick | string | 否 | 卖家的昵称 | xxx 旗舰店 |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |
| appKey | string | 否 | 希望写入 RDS 对应的 AppKey, 以逗号分隔 | 123543,89765 |

6.6.2 派单结果同步

taobao.omniorder.allocatedinfo.sync

API 基本信息

- API 名称: 分单结果同步给星盘
- API 描述: ISV 分单完成, 将分单结果同步给星盘.

API 请求入参

| 名称 | 类型 | 是否 必须 | 描述 | 示例值 |
|----------------------|--------|----------|--|------------------|
| tid | number | 是 | 淘宝交易主订 单 id | 2004574256073903 |
| status | string | 是 | 订单状态, 可选 值: Waiting(仍 在 派单中) Allocated(派 单成功) AllocateFail(派 单失败) | Allocated |
| message | string | 否 | 错误的内 容描 述 | internal error |
| store_allocated_list | struct | 是 | 门店分单列表 | |
| store_id | string | 是 | 门店 Id | 123 |
| store_name | string | 否 | 门店名称 | XXX 门店 |
| tid | number | 是 | 淘宝交易主订 单 id | 2004574256073903 |
| sub_oid | number | 是 | 淘宝交易子订 单 id | 2004574256073903 |
| type | string | 是 | 发货类型, 可选 | Store |

| | | | | |
|-------------|--------|---|--|-----------|
| | | | 值: Store(门店发 货) Warehouse(电 商仓) | |
| status | string | 是 | 订单分单状态, 可选值: Waiting(仍在 派单中) Allocated(派 单成功) AllocateFail(派 单失败) | Allocated |
| description | string | 否 | 描述 | |
| attributes | struct | 否 | 扩展属性字段 | |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" :接单成功 ;其他 表示操作失败 , 请返 回错误 message 信 息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

6.6.3 派单结果查询

说明：为避免派单结果同步异常，增加星盘主动查询派单状态接口

taobao.qimen.omniorder.allocatedinfo.get

API 基本信息

- API 名称: 根据订单 ID 获取分单信息
- API 描述: 星盘调用 ISV 接口，获取门店分单信息.

API 请求入参

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-----|--------|------|------------|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|--------|--------|------|---------------------------|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| status | string | 是 | 订单状态, 可选值: Waiting(仍在派 | Allocated |

| | | | | |
|----------------------|--------|---|--|------------------|
| | | | 单中) Allocated(派单成功) AllocateFail(派单失败) | |
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |
| store_allocated_list | struct | 是 | 门店分单列表 | |
| store_id | string | 是 | 门店 Id | 123 |
| store_name | string | 否 | 门店名称 | XXX 门店 |
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| sub_oid | number | 是 | 淘宝交易子订单 id | 2004574256073903 |
| type | string | 是 | 发货类型, 可选值: Store(门店发货) Warehouse(电商仓) | Store |

| | | | | |
|-------------|--------|---|--|-----------|
| status | string | 是 | 订单分单状态, 可选值: Waiting(仍在派 单中) Allocated(派单 成功) AllocateFail(派 单失败) | Allocated |
| description | string | 否 | 描述 | |

6.6.4 退款

taobao.qimen.omniorder.customer.refund

API 基本信息

- API 名称: 客户退款通知 ISV
- API 描述: 客户退款 API , 星盘通知 ISV 客户发起退款.

API 请求入参

| 名称 | 类型 | 是 否 必 须 | 描述 | 示例值 |
|----|----|------------------|----|-----|
| | | | | |

| | | | | |
|----------------|--------|---|---|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| sub_order_list | struct | 是 | 退款子订单列表 | |
| sub_oid | number | 是 | 子订单 Id | 2004574256073903 |
| status | string | 是 | 订单状态, 可选值: STATUS_NOT_PAY(等待买家付款) STATUS_PAID (等待卖家 发 货) STATUS_REFUNDED (交 易 关 闭) STATUS_TRANSFERR ED (交易成功) | STATUS_PAID |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

6.6.5 星盘通知 POS 已接单

taobao.qimen.omniorder.store.accpeted

API 基本信息

- API 名称: POS 接单
- API 描述: 星盘将 POS 接单消息通知 ISV.

API 请求入参

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-----------|--------|------|------------|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| seller_id | number | 是 | 卖家的淘宝 id | 12345 |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

6.6.6 星盘通知 POS 已拒单

taobao.qimen.omniorder.store.refused

API 基本信息

- API 名称: POS 拒单
- API 描述: POS 拒单, 星盘通知 ISV.

API 请求入参

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-----------|--------|------|------------|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| seller_id | number | 是 | 卖家的淘宝 id | 12345 |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|---|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败, 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

6.6.7 星盘通知 POS 已发货

taobao.qimen.omniorder.store.consigned

API 基本信息

- API 名称: POS 发货成功
- API 描述: POS 发货成功, 星盘通知 ISV.

API 请求入参

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|-----------|--------|------|------------|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| seller_id | number | 是 | 卖家的淘宝 id | 12345 |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|---|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败, 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

| | | | | |
|----------|--------|---|-----|----|
| operator | string | 是 | 操作员 | 小王 |
|----------|--------|---|-----|----|

| | | | | |
|----------------|--------|---|---|------------------|
| sub_order_list | struct | 是 | 退款子订单列表 | |
| sub_oid | number | 是 | 子订单 Id | 2004574256073903 |
| status | string | 是 | 订单状态, 可选值: STATUS_NOT_PAY(等待买家付款) STATUS_PAID (等待卖家 发 货) STATUS_REFUNDED (交 易 关 闭) STATUS_TRANSFERRING D (交易成功) | STATUS_PAID |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

6.6.8 POS 接单通知星盘

taobao.omniorder.store.accpeted

API 基本信息

- API 名称: POS 端接单接口
- API 描述: ISV POS 端接单, 通知星盘.

API 请求入参

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|----------------|--------|------|---|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| store_id | string | 是 | 门店 Id | 1342 |
| type | string | 是 | 发货类型, 可选值: Store(门店发货) Warehouse(电商仓) | Store |
| operator | string | 是 | 操作员 | 小王 |
| sub_order_list | struct | 是 | 退款子订单列表 | |
| sub_oid | number | 是 | 子订单 Id | 2004574256073903 |

| | | | | |
|--------|--------|---|---|-------------|
| status | string | 是 | 订单状态, 可选值: STATUS_NOT_PAY(等待买家付款) STATUS_PAID (等待卖家发货) STATUS_REFUNDED (交易 关 闭) STATUS_TRANSFERE D (交易成功) | STATUS_PAID |
|--------|--------|---|---|-------------|

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" :接单成功 ;其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

6.6.9 POS 拒单通知星盘

taobao.omniorder.store.refuse

API 基本信息

- API 名称: POS 端拒单
- API 描述: ISV POS 端拒单, 通知星盘.

API 请求入参

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|----------------|--------|------|---|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| store_id | string | 是 | 门店 Id | 1342 |
| type | string | 是 | 发货类型, 可选值: Store(门店发货) Warehouse(电商仓) | Store |
| operator | string | 是 | 操作员 | 小王 |
| sub_order_list | struct | 是 | 退款子订单列表 | |
| sub_oid | number | 是 | 子订单 Id | 2004574256073903 |
| status | string | 是 | 订单状态, 可选值: STATUS_NOT_PAY(等待买家付款) | STATUS_PAID |

| | | | | |
|--|--|--|--------------------------|--|
| | | | STATUS_PAID (等待卖家发货) | |
| | | | STATUS_REFUNDED (交易关闭) | |
| | | | STATUS_TRANSFERED (交易成功) | |

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

6.6.10 POS 发货通知星盘

taobao.omniorder.store.consigned

API 基本信息

- API 名称: POS 端发货

■ API 描述: ISV POS 端发货, 通知星盘.

API 请求入参

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|------------------|--------|------|---|------------------|
| tid | number | 是 | 淘宝交易主订单 id | 2004574256073903 |
| store_id | string | 是 | 门店 Id | 1342 |
| type | string | 是 | 发货类型, 可选值: Store(门店发货) Warehouse(电商仓) | Store |
| operator | string | 是 | 操作员 | 小王 |
| logistic_id | string | 是 | 物流 ID | 233324 |
| logistic_company | string | 是 | 物流公司 | 顺丰 |
| mail_no | string | 是 | 物流单号 | 2323232 |
| description | string | 否 | 发货描述 | 已发送 |
| sub_order_list | struct | 是 | 退款子订单列表 | |
| sub_oid | number | 是 | 子订单 Id | 2004574256073903 |
| status | string | 是 | 订单状态, 可选值: | STATUS_PAID |

| | | | | |
|--|--|--|---|--|
| | | | STATUS_NOT_PAY(等待买家付款) STATUS_PAID (等待卖家发货) STATUS_REFUNDED (交易关闭) STATUS_TRANSFERE D (交易成功) | |
|--|--|--|---|--|

API 返回结果

| 名称 | 类型 | 是否必须 | 描述 | 示例值 |
|---------|--------|------|--|----------------|
| code | string | 是 | "0" : 接单成功 ; 其他表示操作失败 , 请返回错误 message 信息 | 500 |
| message | string | 否 | 错误的内容描述 | internal error |

7 全渠道 SPI 的接入和部署方式

SPI 是由淘宝制定接口标准 , 由第三方的 ISV 负责实现服务并将调用地址绑定到标准接口的

API。在全渠道场景中 , SPI 主要应用于星盘调用商家系统完成启动分单、状态通知等操作。

7.1 SPI 接入方式

<http://open.taobao.com/doc2/detail.htm?spm=a219a.7386653.6.3.lRwrnp&treeId>

[=1&articleId=103784&docType=1](#)

7.2 全渠道 SPI 部署到聚石塔 EWS

SPI 服务必须部署在聚石塔内，请开发商使用聚石塔 EWS 平台进行部署和管理，以提高开发运维效率。在 SPI 的接口配置的线上环境和测试环境地址都必须是 EWS 的服务地址。

聚石塔 EWS 平台使用步骤如下：

第一步：购买 ECS 资源

需要先在聚石塔中购买 ECS 云服务器，[购买方式介绍](#)

EWS 容器对有云服务器的操作系统有限制。目前支持 Ubuntu 和 CentOS 两种系统，同时对操作系统的版本也有要求，云服务器启动后还需要安装管理 Agent 和 Docker 环境，就要操作说明如下：

[Ubuntu 操作系统说明](#)

[CentOS 操作系统说明](#)

按步骤执行完成后，这些 ECS 云服务就成功注册为 EWS 的宿主机，可供部署 EWS 容器使用了。宿主机的容器所在目录及容器内部的目录结构[说明文档](#)

第二步：创建 EWS 服务

创建完成 EWS 宿主机后，就可以创建 EWS 服务了，EWS 服务就是全渠道商家中台系统的应用程序。

EWS 提供给了多种语言的应用服务器镜像供开发商选择，开发者也可以创建自定义的镜像；创建的 EWS 服务后，系统会分配给服务一个以 hz.taeapp.com 为域名后缀的地址（前缀由开发商自定义），该地址就是全渠道商家中台系统的访问地址。

在配置 SPI 绑定接口时所填写的地址就是该域名下的地址，建议开发商创建两个服务，一个

供线上环境使用，一个供测试环境使用。

[具体创建流程说明](#)

第三步：部署应用程序

创建好服务后，就可以部署应用程序了，EWS 提供了多种部署方式供开发商选择：

[单个服务部署](#)

[多个服务编排部署](#)

第四步：EWS 服务的日常监控

[AliAPM 性能监控平台](#)

8 ISV 系统压测

[商品通压测指南](#)