

91 移动开发平台支付接入参考

V2.2

目录

前言	3
什么是 91 豆	3
如何选择接入模式.....	3
拥有自己的业务服务器.....	3
没有自己的服务器.....	3
应用内支付接入模式.....	4
1) 同步支付.....	4
2) 异步支付.....	4
3) 我该选择哪种应用内支付接入方式.....	5
代币充值接入模式.....	6
详细接入流程.....	7
1) 纯单机接入.....	7
2) 第三方服务器接入（即业务服务器）	7
3) 代币充值接入.....	9
4) 虚拟商店接入.....	9
5) 漏单处理.....	9
虚拟商店.....	10
介绍.....	10
接入分类.....	10
1) 虚拟商店联机模式	10
2) 虚拟商店本地模式.....	11
支付货币.....	11
商品类型.....	11
1) 非消费型商品.....	11
2) 订阅型商品.....	12
3) 消费型商品.....	12
商品配置.....	12
商品呈现.....	12
商品购买.....	13
商品查询.....	13
1) 非消费型商品.....	13
2) 订阅型商品.....	14
3) 消费型商品.....	14
商品使用.....	14
接入流程（联机模式）	14
1) 非消费型商品接入.....	14
2) 订阅型商品接入.....	15
3) 消费型商品接入.....	17
接入流程（本地模式）	19

前言

该参考手册是开发者在接入 91 移动开发平台(精简版 SDK, 完全版 SDK)过程中, 以帮助开发者根据自己产品的特性, 选择合适的接入模式。手册中, 我们根据不同接入模式, 提供了接入流程。为了保障付费用户的利益, 建议开发者根据流程进行接入。

什么是 91 豆

91 豆是博远无线公司针对移动支付而发行的一种通用代币, 类似 Q 币, Facebook 币一样。具有在 91 平台中流通的特性。用户只要通过 91 帐号及其关联帐号, 就可以使用 91 豆在 91 平台旗下的各个移动产品或者合作伙伴的产品中使用移动支付功能。

1 元 = 1 个 91 豆, 可以支持 0.01 个 91 豆。

开发者将以所在应用中用户消费的 91 豆数额作为应用收入的结算

如何选择接入模式

我们以是否联机应用为标准, 将开发者的应用分为以下两类。在继续阅读下文前, 开发者应该需要明确自己的应用归类, 以及应该选择的接入模式。明确本节所描述的内容将有助于您更好的进行支付功能的接入。

拥有自己的业务服务器

应用拥有自己的业务服务器, 只要在业务服务器上实现接收支付结果通知的相关业务逻辑, 在后台配置了相关的服务器通知地址, 就可以选择接入以下的支付模式。

- **代币充值接入**: 要求应用内必须有自己的虚拟代币 (例如: 金币), 开发者可以通过 91 平台提供的充值渠道, 方便快捷的为开发者的虚拟代币进行充值。这种方式需要业务服务器配合接入。
- **应用内支付接入**: 不论是否拥有虚拟货币都可以进行接入, 区分为同步支付和异步支付。其中同步支付可在客户端完成, 异步支付需要业务服务器配合。具体接入请参考【应用内购买接入模式】。

没有自己的服务器

单机应用, 没有自己的业务服务器, 或者没有计划进行业务服务器接入的开发者, 可以选择以下的接入方式。其中虚拟商店的相关介绍参加后面【虚拟商店】章节。

- **纯单机接入**：使用应用内同步支付模式，开发者将商品信息写死在客户端代码中，玩家购买的物品也是存储在客户端配置文件中。整个支付流程均在客户端完成。开发者需要进行漏单处理。
- **虚拟商店本地模式**：使用应用内同步支付模式，开发者将商品信息配置在虚拟商店服务器，玩家购买的物品存储在客户端，同时开发者需要进行漏单处理。
- **虚拟商店联机模式**：使用应用内异步支付模式，开发者将商品信息配置在虚拟商店服务器，玩家购买的物品存储在虚拟商店服务端，开发者使用前需要向服务端查询，玩家使用的情况需要通知服务端进行消减动作。

同时虚拟商店支持除 91 豆外的自定义虚拟货币，方便开发者更灵活的使用虚拟货币进行活动奖励。

应用内支付接入模式

平台提供两种应用内支付接入模式：同步支付(-NdUniPay:)和异步支付(-NdUniPayAsyn:)。您可以根据自己的需求选择使用其中的一种方式。

1) 同步支付

使用同步购买，用户仅能使用余额进行购买。当余额不足时，无法进行购买，需要进行充值，**用户充值成功后，需要重新发起购买操作**。在购买结束后，平台会立即向客户端应用程序发送购买结果的通知。

2) 异步支付

使用异步购买，支付结果将以消息的方式通知到您的业务服务器或者服务端虚拟商店中。应用接入方需要在开发者后台配置支付回传通知地址。您的业务服务器或者服务端虚拟商店需要处理用户的支付结果通知。具体参见 **SDK** 开发包中的【91 移动开发平台服务端与应用服务端接口规范 1.00.pdf】。

在用户余额充足的情况下，异步购买的表现与同步购买一致，在购买结束后，平台会向客户端应用程序发送购买结果的通知。

异步购买提供了以下两种功能：

好友代付：

如果您接入的是完整版 SDK，在余额不足的界面底部，会有“让好友购买”的功能。您可以选择你的好友，为您进行支付。

注：该功能仅在完整版 SDK 中支持，精简版不支持该功能。

充值直购：

在余额不足的界面右上角，还有一个“91 豆充值”的按钮。使用该功能，可以进行充值，并且，在充值金额到账后，由服务器自动完成该项购买。

由于异步购买功能无法立即知晓购买结果，所以您需要向您的业务服务器或则虚拟商店服务端查询购买结果。

3) 我该选择哪种应用内支付接入方式

同步支付客户端能够马上获知购买结果，所以，一般情况下，同步支付能够满足大多数应用的需求。

异步支付与同步支付最大的区别是：**异步支付能够在余额不足的情况下**，在用户充值成功后，由平台自动购买商品，用户不需要再次发起购买的操作。由于充值可能会带来延时性，会导致无法立即获得购买结果。但是**余额足够的情况下**，跟同步支付一样，立即就可获得支付结果。因此异步支付要求应用有属于自己的服务器或者托管到虚拟商店。

根据不同情况，选择相应的购买模式，我们建议使用如下。

1) 使用同步支付（满足以下条件之一）

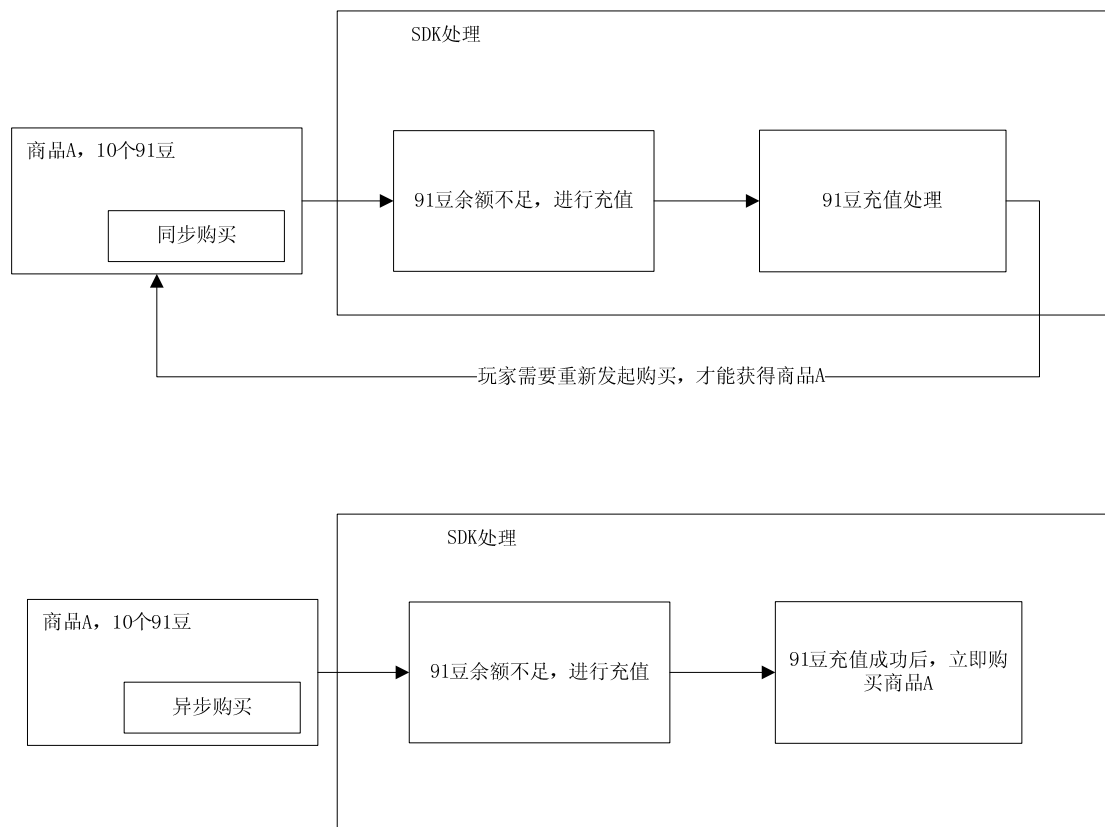
- 物品发货是在客户端进行。例如：购买后，由客户端直接把道具缓存到客户端文件中。
- 用户使用时，不需要联网的。例如：购买后，每次启动，从客户端文件获取用户购买的道具情况，不需要向服务器获取用户购买的道具情况。
- 对道具发货实时性要求很高。例如：某部电影，用户付费后才能观看。

2) 使用异步支付（满足以下全部条件）

- 物品发货是在服务端进行。例如：网络游戏。玩家购买的道具，直接在服务端将道具划入到玩家帐号下的相关数据库中。客户端通过刷新获取服务端的已购买道具情况。
- 用户每次使用时，都向服务端获取已购买道具情况。
- 对道具发货实时性要求不高。例如：玩家账户上的金币。购买后，玩家看到的金币还没到账，过会儿刷新后，金币就到账了。

异步支付简化了用户的购买操作，提供了更好的购买体验。发生余额不足时，用户不需要充值完成后重新购买。异步购买会在用户充值完成后，自动购买。

例如：



最后，需要强调的是，异步接入必须要有自己的业务服务器或者托管到虚拟商店中。

代币充值接入模式

代币是开发者的应用内流通的虚拟货币，代币充值是 91 平台提供的以充值的方式进行购买代币的接入模式。具体流程请参见下面的【代币充值接入】。

接入代币充值需要开发者在后台配置**代币的名称，单位以及同人民币的汇率**。同时需要配置支付结果回传通知地址，以便在发生充值的时候回传充值结果。业务服务器需要处理这个结果通知来选择发货。

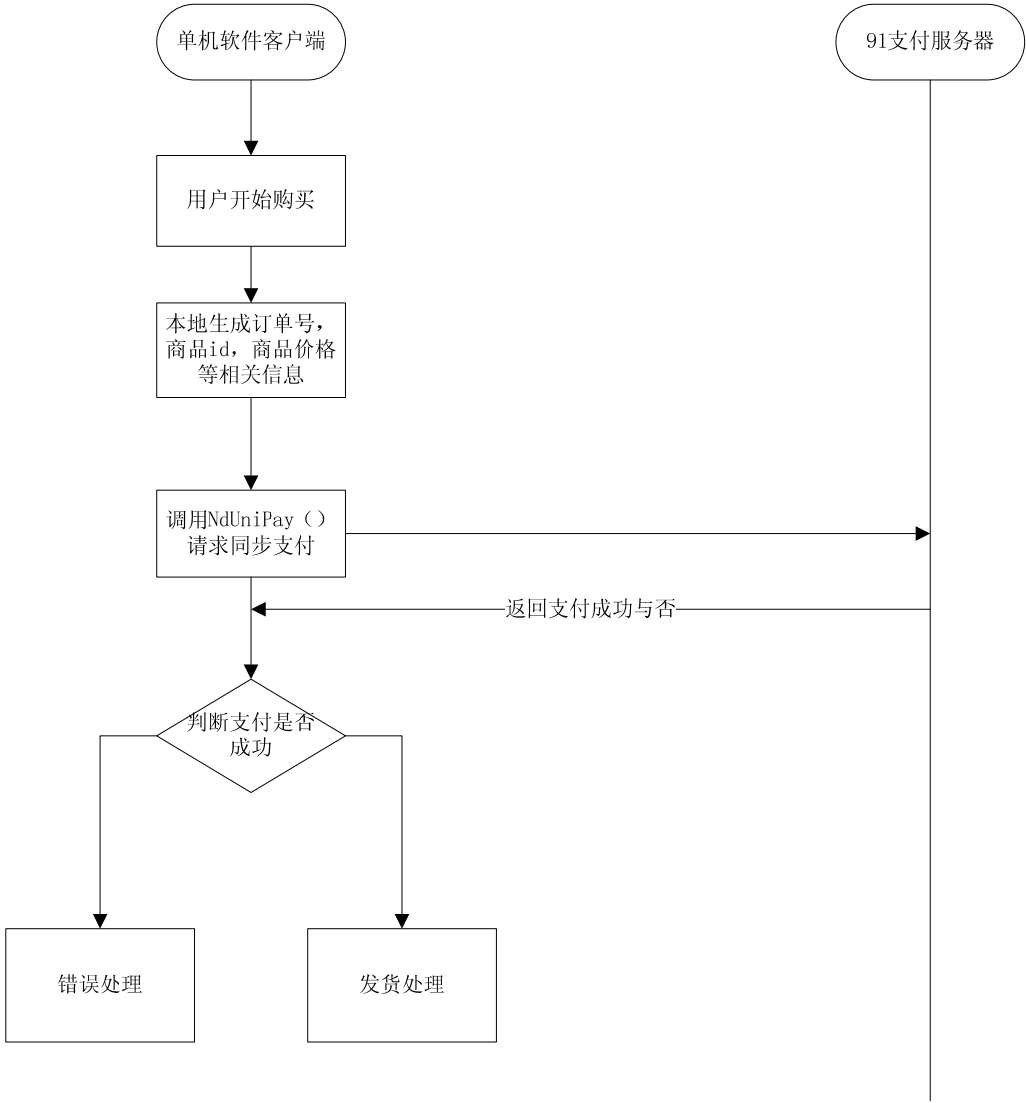
代币充值模式要求开发者必须拥有自己的业务服务器。服务端对接的接口文档，请参见【91 移动开发平台服务端与应用服务端接口规范 1.00.pdf】

代币充值的发货行为必须在服务端进行，客户端通过获取或刷新机制，向业务服务器请求充值结果。

例如： 开发者定义自己的游戏币为：元宝， 单位为个， 比例为：1 元人民币=10 个元宝。

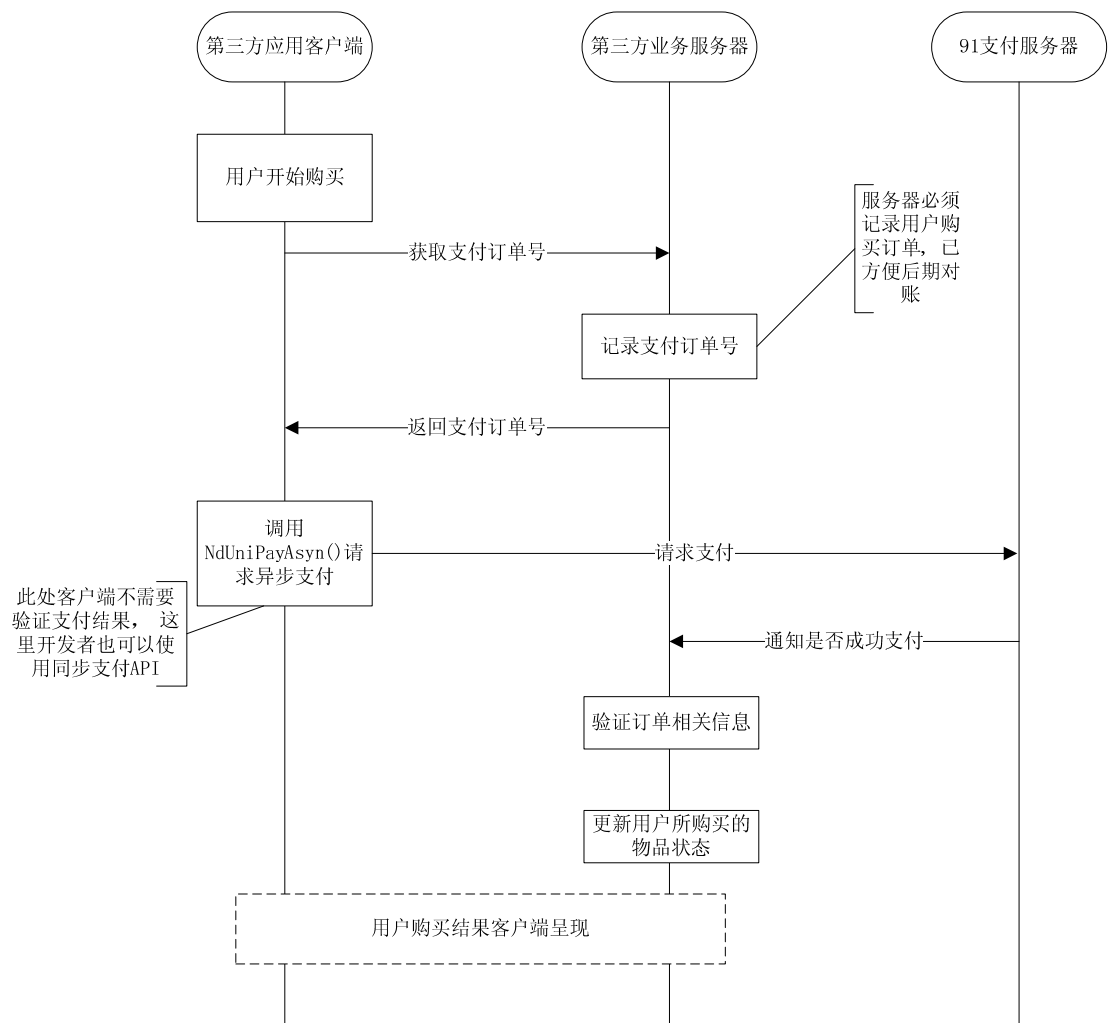
详细接入流程

1) 纯单机接入

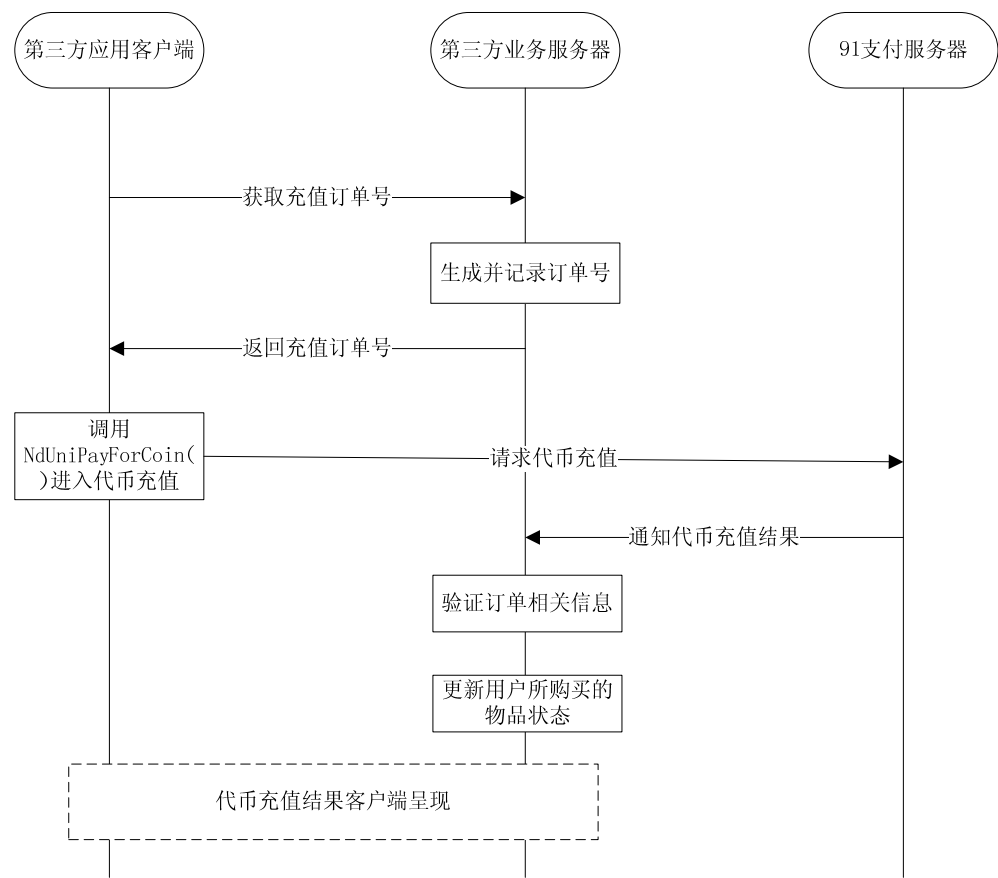


注：由于纯单机应用，由于网络原因，会导致支付结果没收到，又或者发货时，程序退出，导致用户没有收到物品，开发者需要进行漏单处理，请参见后面的【漏单处理】。

2) 第三方服务器接入（即业务服务器）



3) 代币充值接入



4) 虚拟商店接入

参见本文后面的虚拟商店

5) 漏单处理

在购买过程中,可能出现以下情况:购买已经完成,而应用程序却没有收到购买结果的通知,例如,购买过程中,用户退出了应用程序,或者网络出现了问题,导致购买成功的消息无法到达客户端。

如果你的 App 中,将用户购买后的发货是在客户端进行,虚拟商品是存储在客户端配置文件中。那么你需要在你的程序建议按照下面做法处理漏单情况。

在发起购买请求后,立即记录下该请求的订单号,在购买结果到达时,删除该条记录。而在每次应用程序启动时,检查是否有未收到购买结果的订单,如果有,向服务器发起验证,并根据验证结果处理该记录。

虚拟商店

介绍

虚拟商店即虚拟商品管理平台，是 91 平台为方便开发者提高软件开发效率以及方便后期运营，提供的虚拟商品管理平台。开发者可以根据运营的需要，动态的管理虚拟商品。

其功能涵盖如下：

1. 虚拟商品的配置，上架，下架
2. 商品促销消息公告
3. 商品打折销售
4. 限量版商品销售
5. 商品限时段抢购
6. 商品限数量抢购
7. 商品订阅销售
8. 自定义虚拟币

开发者可以通过虚拟商品管理平台，灵活的运用以上功能，将会促进所在应用的软件销售。另外虚拟商店管理玩家目前在该应用中所拥有的虚拟商品的数量及状态。

虚拟商店的针对那些单机版应用，而没有提供自己的业务服务器。或者开发者可以不用再开发一套虚拟商品管理系统，直接使用 91 平台提供的虚拟商店功能，进一步提高效率。

虚拟商店加入了促销通知，打折销售，限量版销售，限时抢购，商品赠送等运营机制。将进一步提高软件收益。另外开发者还可以自定义虚拟货币，将单机软件变成网游模式。

使用虚拟商店接入， 开发者需要在后台配置所要销售的商品名称，商品图标，商品价格等相关信息

接入分类

1) 虚拟商店联机模式

玩家购买的虚拟物品是存储在虚拟商店服务端，开发者需要在玩家登录后向虚拟商店服务器获取玩家已购买的虚拟物品。当玩家使用虚拟道具时，需要通知虚拟物品服务端进行消减操作。

2) 虚拟商店本地模式

虚拟商店中的所配置的虚拟商品仅作为配置属性的作用。玩家购买的虚拟物品是存储在本地客户端，当玩家使用虚拟道具时，由开发者自行在本地进行处理。

支付货币

开发者在使用虚拟商店时，需要预先选择使用哪种支付货币在虚拟商店中使用。默认是使用 91 豆。

1) 使用 91 豆进行支付

所有虚拟商品的购买都是以 91 豆作为支付货币。由于 91 豆是所有应用间的统一流通货币，也是与开发者收益结算的货币，不能随便对用户进行 91 豆奖励。

2) 自定义虚拟币

开发者可以在虚拟商店中预先定义一种虚拟币，以及与 91 豆的兑换比例后，所有虚拟商品的购买都是通过自定义虚拟币来作为支付货币。这样开发者不需要单独开发虚拟币。

例如：开发者自己定义一个虚拟币叫“元宝”，作为游戏内使用。它与 91 豆的兑换比率为：1 个 91 豆 = 100 个元宝。玩家先充值元宝，然后再用元宝购买虚拟道具

这样的好处是，开发者可以不受财务等相关政策的限制，使用自定义虚拟币作为活动奖励。

商品类型

虚拟商品根据不同的业务特性，主要分为以下三类类型。开发者可根据自己软件的特性，选择其中一种商品类型。商品类型是互斥，开发者只能选中一种类型。

1) 非消费型商品

用户只需要购买一次，用户一旦购买就与 91 账号关联，没有购买数量的要求，通常用于关卡类、试用版转正式版类消费。此类商品主要适合于单机类的应用。通常这类只需要配置商品 ID 和商品价格即可

注：开发者可以在后台设置该非消费型商品，跟购买设备绑定，即只能在所购买的那个设备上使用，其他设备上不能使用。

2) 订阅型商品

该商品没有购买数量的要求，但有使用次数与使用周期的要求，通常用于订阅服务类消费。例如：某视频软件中，用户可以包月看视频。某工具软件，用户可以使用该工具的次数。此类商品主要适合需要订阅的应用软件。通常这类商品只需要配置商品 ID，商品价格，购买后允许可以使用的时间周期，或者购买后允许可以使用的次数。

注：开发者可以在后台设置该订阅型商品（按使用周期），跟购买设备绑定，即只能在所购买的那个设备上使用，其他设备上不能使用。

3) 消费型商品

该商品就是我们在网络游戏中常见的道具类商品。用户可以针对该商品，购买 1 个或 1 个以上商品，有购买数量的要求。平台维护玩家购买及使用后的商品数量及状态情况。通过这种方式，开发者很容易将单机版的道具类游戏转换为弱网游游戏。引入运营的概念。

消费型商品可设定的属性较多。开发者可以根据运营的需要，灵活配置。主要设置如下

1. 降价销售
2. 限量版商品销售（开发者可以推出价格高昂的限量版商品，只有有限个玩家拥有）
3. 限时抢购（例如：开发者可配置某个商品，仅在晚上 10:00-12:00 销售）
4. 限量抢购（例如：开发者可配置某个商品，玩家只允许购买有限数量的商品）

商品配置

开发者可以在 dev.91.com 的开发者管理后台中的【应用管理】下的【虚拟商店】中配置所需类型的商品相关信息。

商品呈现

91SDK 已经内置了通用的虚拟商店的手机端呈现，如下图：



开发者只需要调用打开虚拟商店的 **API**，即可完成商品的呈现，购买功能。不需要额外的操作。

另外若开发者需要自己呈现商品信息，也可以利用平台对外提供的商品数据 **API**，开发者可以根据这些 **API**，获取开发者后台所配置的所有商品信息，然后根据这些商品信息，自己制作虚拟商店的 **UI** 层现。

通常消费型商品，需要有专门界面表现商店列表。非消费型商品，订阅型商品，由于这两类通常商品数量较少，建议开发者自己通过获取商品信息接口，自己做商品呈现。

商品购买

- 1) 对于使用虚拟商店界面的，商品购买处理已经涵盖， 开发者不需要另外处理。
- 2) 对于开发者自己制作虚拟商店界面的， 则需要单独调用购买 **API**，进行购买。

考虑支付的安全性，以及避免断网，死机造成玩家不能正常收取虚拟物品。所有购买的虚拟商品，都放在虚拟商店服务端，不另外单独返回购买成功与否。 开发者需要单独调用接口，获取玩家已购买的商品信息。

商品查询

对于用户已经购买的商品， 虚拟商店服务端都会进行登记，并维护商品的数量以及状态。 根据不同商品的类型，使用不同的 **API** 查询商品的数量或状态

1) 非消费型商品

使用 `NdProductIsPaid(String productId)` **API** 判断该用户是否已经购买了该商品。

2) 订阅型商品

使用 `NdProductIsExpired(String productId)` API 判断该用户所购买的该商品是否已经使用时间到期或者使用次数到期，又或者尚未购买该商品。

3) 消费型商品

使用 `NdGetUserProduct()` API 或 `NdGetUserProductsList()` API 获取当前用户在虚拟商店中指定商品或者所有商品的库存状态。

注：以上 API 没有标明详细参数，及回调函数，具体需要参见各平台的使用手册中的相关 API 描述

商品使用

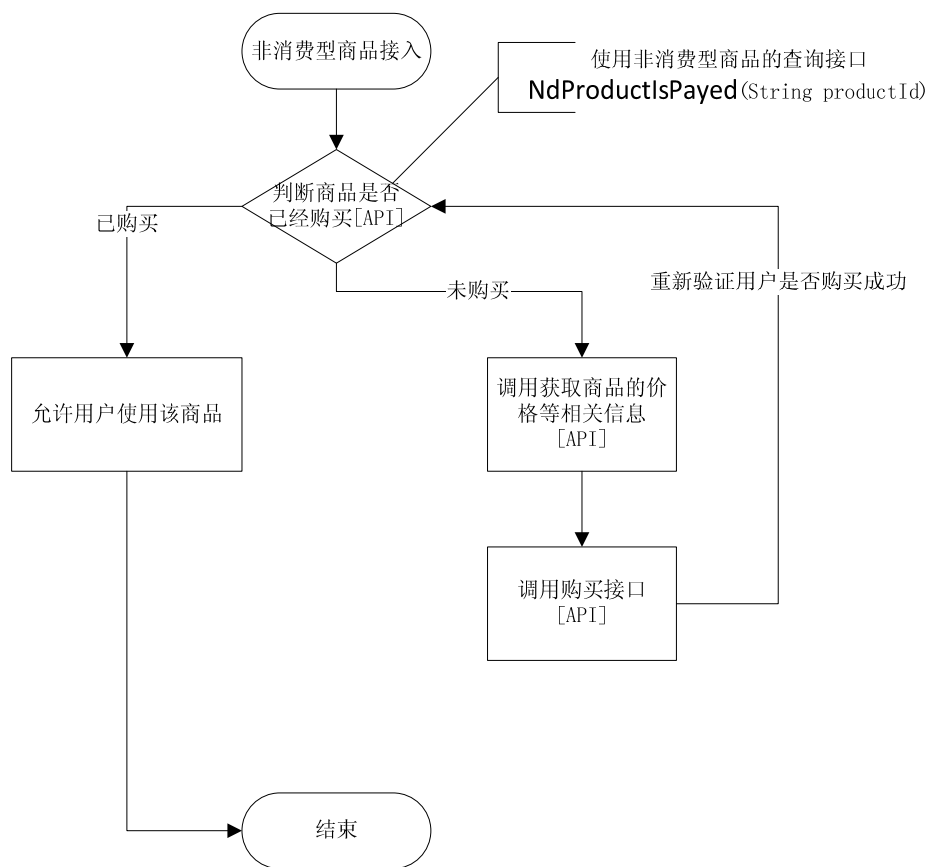
对于订阅型商品（使用次数订阅模式），以及消费型商品，需要调用使用商品 API，用于表示用户使用该商品。通过该 API，将通知 91 服务端，更新商品的使用次数，或者减少用户拥有该商品的数量。

对于非消费型商品，以及订阅型商品（使用时间订阅模式），则不需要调用使用 API，直接通过相应类型的查询接口即可。不需要更新服务端的状态。

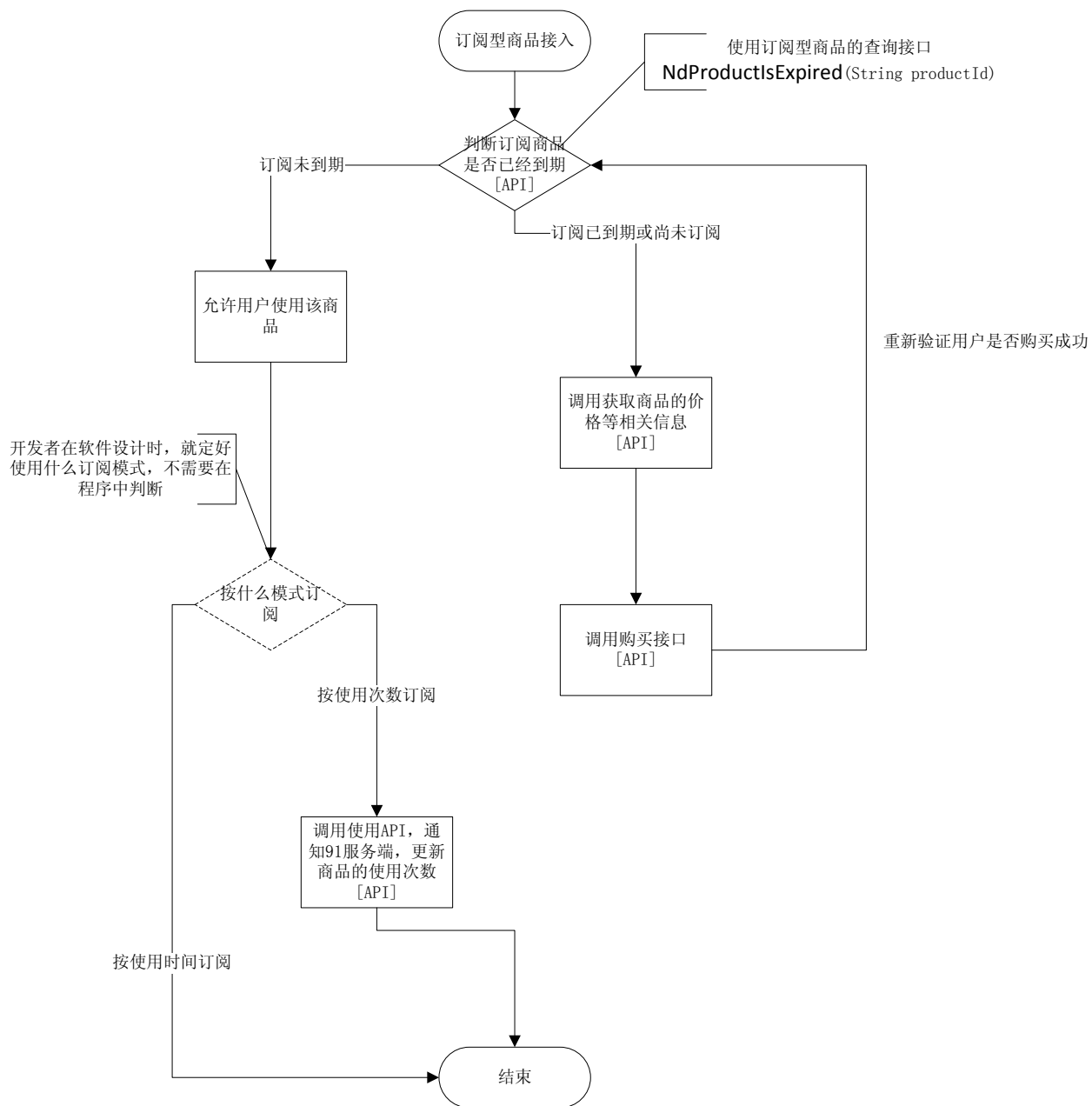
接入流程（联机模式）

所有类型的虚拟商品，开发者都要事先在开发者后台配置好相关的虚拟商品

1) 非消费型商品接入

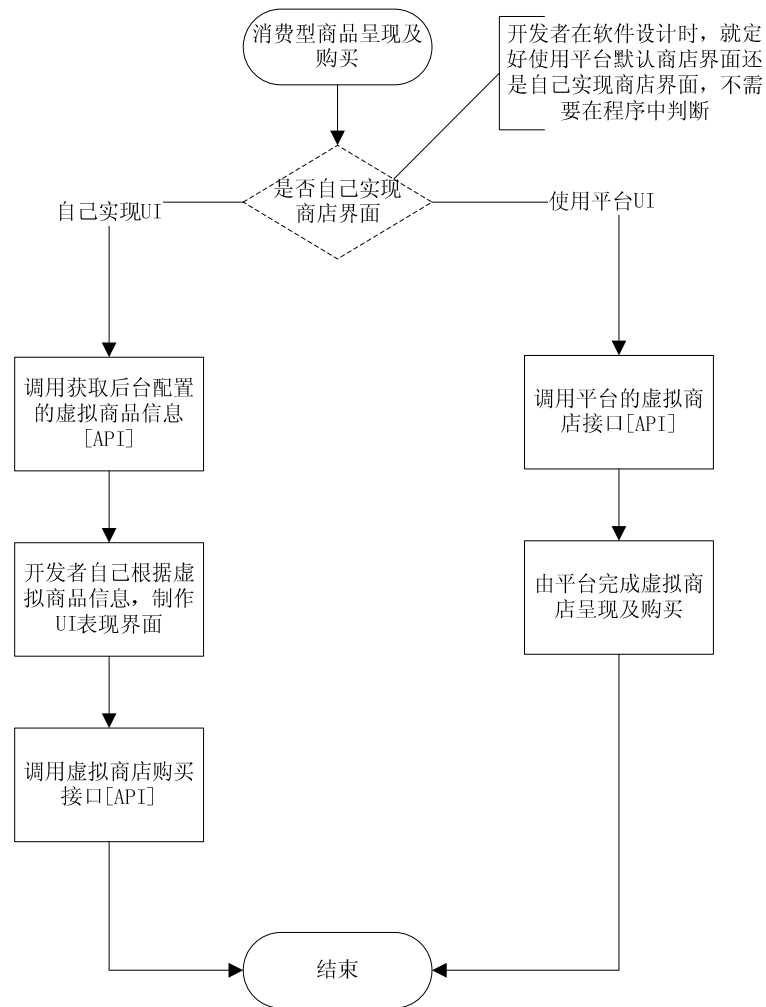


2) 订阅型商品接入

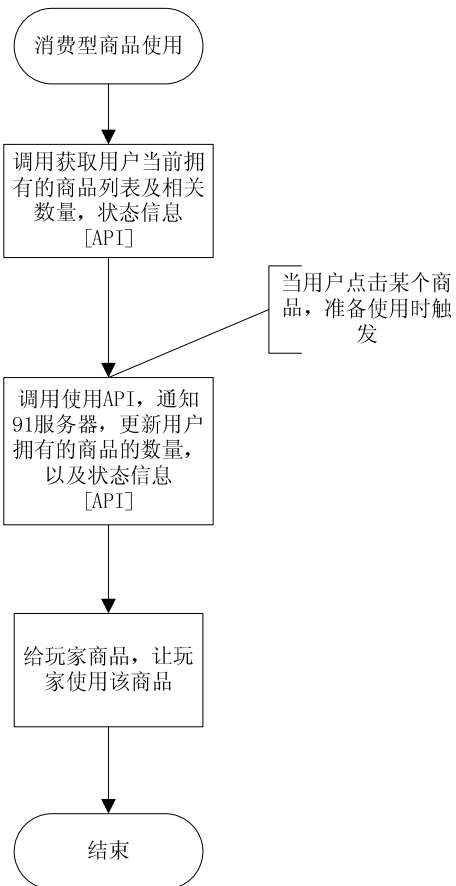


3) 消费型商品接入

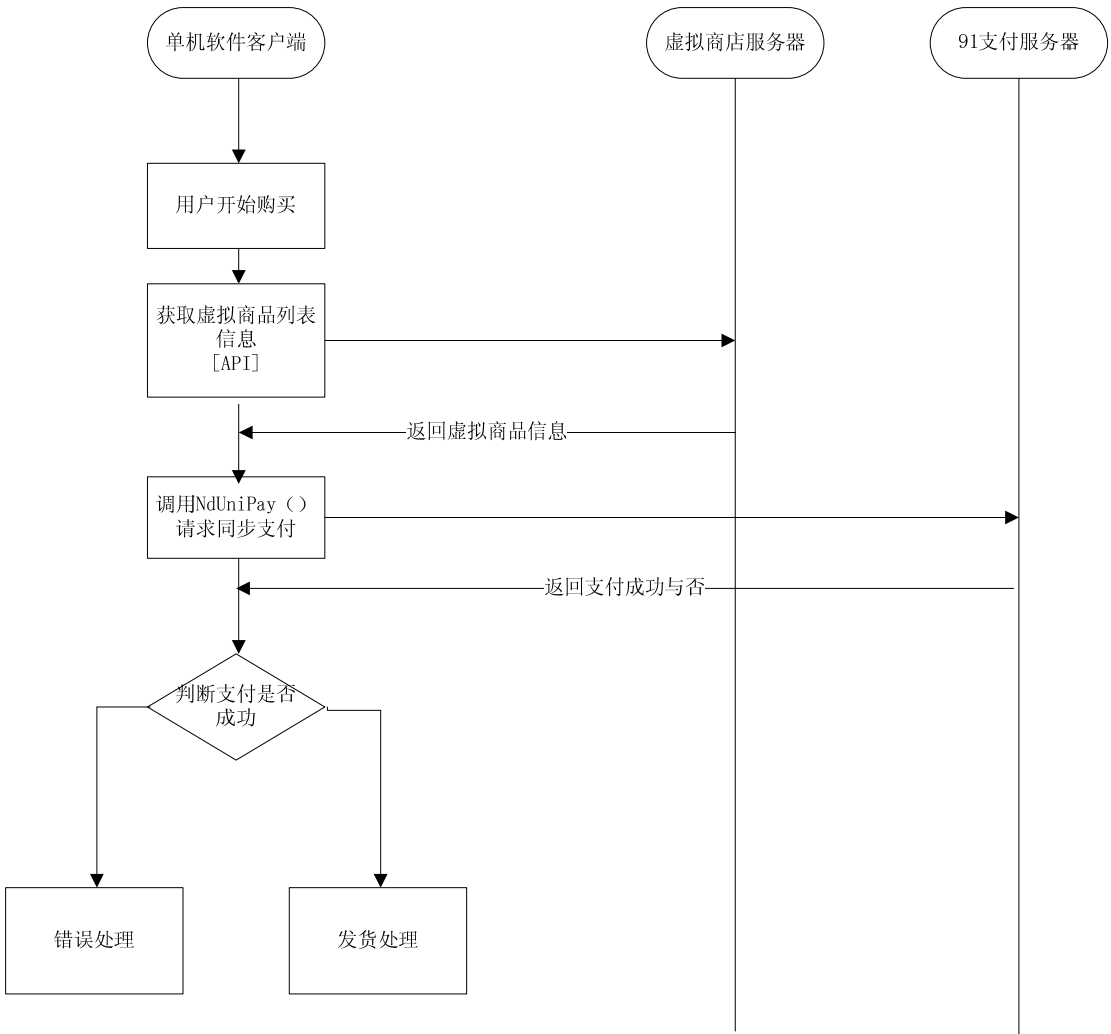
<1> 商品呈现及购买



<2> 商品使用



接入流程（本地模式）



注：由于虚拟商店本地模式，使用的是同步支付接口，由于网络的原因，某些情况会导致支付结果没收到，又或者发货时，程序退出，导致用户没有收到物品，开发者需要进行漏单处理，请参见【漏单处理】