# 客户端计费流程

本文档主要介绍在游戏内部的计费流程，不涉及到SDK，具体的SDK计费雷神知道，这个文档内就不写了；

## 计费基础设计

游戏中的计费点采用一对多的形式，下面介绍一对一和一对多两种形式的优劣；

**一对一形式：**就是游戏中每一个计费点，在苹果或者Google都申请一个计费点，好处是一对一处理，坏处是每次新增计费点都需要后台操作，而且苹果审核要打开所有计费点，不方便提审，同时不好做自定义礼包；

**一对多形式：**苹果和Google后台只申请单个价格的计费点，比如0.99的，申请一个，然后游戏中所有的计费点配置表，关联一个价格计费点，比如游戏中卖一个新人礼包和工人礼包都是0.99，他们在我们业务层的计费点id不同，一个是worker，一个是new，但是对应的苹果或者Google后台的计费点ID都一样，都是比如price\_099这样；好处是每次提审，只需要给苹果看钻石计费点一个界面就可以，而且可以随时配置新的计费点，坏处是容易出错，需要额外的数据关联；

## 代码对象

GoldExchangeItem：计费点对象，由服务器下发的，包含了计费点的很多信息，比如名称、图标、价格、道具内容等；

PayController：计费管理类，所有的计费流程都通过这个类的单例去操作；

## 通用计费流程

PayController.getInstance().callPayment(GoldExchangeItem purchasingItem…)

客户端所有的计费调用，都通过上面这个接口去调用

1. 发送PayRecordCommand协议，记录玩家的计费打点（原游戏打点）；
2. 发送PaymentBeforeCheckCommand协议，用来判断玩家是否可以购买该计费点，以及构建一个计费的订单号；
3. 判断计费点类型（下述计费点类型都是为了统一游戏逻辑而做，这样不管是免费的，还是钻石的，都可以走计费逻辑，客户端方便很多）：
   1. Free计费点：发送PaymentFreeCommand协议
   2. goldBrick（金砖）计费点：发送PaymentGoldBrickCommand协议；
   3. diamond（钻石）计费点：发送PaymentDiamondCommand协议；
   4. 常规计费点（收钱的那种）：
      1. 没有开启SDK的情况（测试模式）：发送PaymentTestCommand协议；
      2. 安卓正式环境：callOnlinePaymentAndroid；
      3. iOS正式环境：callOnlinePaymentIOS；
      4. GS包正式环境：callOnlinePaymentWinGS；
4. 计费成功之后，客户端会调用onCallPaymentServerCallback函数；
5. 最后调用CCCommonUtil中的payParseData函数，解析计费结果，获取道具展示计费成功消息；