设置->开发者账号->账号详情->许可测试  
填写"可用于测试的 Gmail帐号", 填写gmail, 多个的话用逗号分割, 保存

## 客户端接入注意事项：

1. 确保包名一致(版本号也要一致)
2. 要使用正式签名打包,签名要与上传至Google后台的一致
3. 要使用测试账号测试，管理员要添加测试人员账号（设置->开发者账号->账号详情->许可测试），填写”可用于测试的Gmail账号”，填写gmail,多个的话用逗号分隔，还要发送测试邀请，测试人员通过链接同意邀请之后才能测试（重要），

<https://blog.csdn.net/u014449046/article/details/85307643>





1. 要测试的订阅商品状态为有效
2. 确保 IabHelper 初始化的 base64PublicKey 与Google后台一致

base64PublicKey获取方式：



## 订阅支付流程：

1)支付失败：

①客户端唤起订阅支付（调用SubscriptionBuy）

②用户支付失败

③回调至lua函数 G\_GooglePayFailCallFunc

2)支付成功:

① 客户端唤起订阅支付（调用SubscriptionBuy）

②用户支付成功

③回调至 mPurchaseFinishedListener.onIabPurchaseFinished 执行异步查询商品信息

④查询完毕回调至 mGetInventoryListenerInPurchase.onQueryInventoryFinished 调用consumeAll函数

⑤将具体订单信息回调至lua函数 G\_GoogleConsumerCallFunc

⑥G\_GoogleConsumerCallFunc调用 Utils:PaySuccess 函数向服务器发送本次订单信息

⑦服务器验证完毕发送相应商品给客户端

## Java层新增函数：

GooglePlayHelper中新增函数:

public static void SubscriptionBuy(String sku, String payLoadString)

使用订阅支付时调用此函数,参数与GoogleBuy一致,sku为产品ID，payLoadString为自定义数据，游戏中是服务器生成的订单ID

## Lua层新增函数：

function NativeManager:googleSubscription(goodsStr, orderId) 调用java层SubscriptionBuy函数

## 新增协议：

PushSubscriptionInfoRequest 订阅信息查询

PushSubscriptionInfoReply 订阅信息查询回复

## 订阅测试界面：

app/layer/layers/subscribe/SubscribeTest.lua

使用方式：

require("app.layers.subscribe.SubscribeTest"):create():addToNode(sceneManager:getRunningScene())

