

|  |
| --- |
| Amigo Play  Android平台SDK开发指南（网游版） |
| V3.1.6.b |

# 目录

[目录 2](#_Toc404790589)

[一、 Amigo Play平台介绍 3](#_Toc404790590)

[1.1 Amigo 账号 3](#_Toc404790591)

[1.2 A币系统 3](#_Toc404790592)

[1.3 名词解释 3](#_Toc404790593)

[二、 接入步骤简介 4](#_Toc404790594)

[三、 SDK流程说明 5](#_Toc404790595)

[3.1 登录流程概述 5](#_Toc404790596)

[3.2 支付流程概述 7](#_Toc404790597)

[四、 SDK客户端接入 8](#_Toc404790598)

[4.1 组件和权限声明 8](#_Toc404790599)

[4.1.1 SDK接入所需环境 8](#_Toc404790600)

[4.1.2 添加SDK组件 8](#_Toc404790601)

[4.1.3 添加SDK所需要的权限 8](#_Toc404790602)

[4.1.4 添加SDK组件所需要声明 9](#_Toc404790603)

[4.2 SDK初始化 9](#_Toc404790604)

[4.3 账号登录 10](#_Toc404790605)

[4.4 启用收银台 11](#_Toc404790606)

[4.4.1 调用方式 11](#_Toc404790607)

[4.4.2 OrderInfo 参数说明 12](#_Toc404790608)

[4.4.3 支付失败返回状态码 12](#_Toc404790609)

[五、 服务器接入 12](#_Toc404790610)

[5.1 AmigoToken安全验证 13](#_Toc404790611)

[5.1.1 请求安全验证 13](#_Toc404790612)

[5.1.2 返回验证结果 14](#_Toc404790613)

[5.2 Amigo支付接入 17](#_Toc404790614)

[5.2.1 创建支付订单 17](#_Toc404790615)

[5.2.2 返回订单创建结果 20](#_Toc404790616)

[5.2.3 通知商户支付结果 22](#_Toc404790617)

[5.2.4 商户返回支付结果确认 24](#_Toc404790618)

[六、 其他问题 26](#_Toc404790619)

[6.1 APK打包 26](#_Toc404790620)

[6.2 横竖屏切换 26](#_Toc404790621)

[6.3 代码混淆 26](#_Toc404790622)

# Amigo Play平台介绍

Amigo Play平台是面向所有开发者，为用户提供游戏、阅读、音乐、视频等精品应用的平台。平台内开放了Amigo用户身份资源、付费通道资源和运营推广资源。

## Amigo 账号

Amigo账号是Amigo 平台上用户通行证账号系统。其中以手机号码注册为主的正式账号。

## A币系统

A币是Amigo 平台针对虚拟商品支付而发行的一种通用代币。用户只要通过Amigo的正式账号进行充值（用人民币购买A币），就可以在Amigo平台上的各应用进行支付。

1元=1 A币，A币最小单位0.01 A币。

开发者将用户在所有应用中消费的A币数额作为应用收入的结算依据。

## 名词解释

|  |  |  |
| --- | --- | --- |
| 编号 | 名称 | 说明 |
| 1 | Amigo Play SDK | SDK客户端 |
| 2 | Amigo Play Server | SDK服务器 |
| 3 | Amigo账号 | Amigo平台账号 |
| 4 | 商户APK | 由商户开发的安装于手机上的游戏。 |
| 5 | 商户服务器 | 由商户开发的和商户APK进行交互的服务端系统。 |
| 6 | playerId | 游戏用户的唯一标识 |
| 7 | AmigoToken | 为商户服务器提供的一种账号安全验证token，是JSON字符串 |
| 8 | NotifyURL | 通知商户交易结果的地址，可在Amigo平台设置，也可在创建订单时自定义，并以代码设置优先 |
| 9 | APIKey | 从Amigo平台申请获得，商户APK的唯一标识，用于鉴别应用的身份 |
| 10 | SecretKey | 从Amigo平台申请获得，商户密钥，用于AmigoToken验证时生成签名 |
| 11 | PrivateKey | 从Amigo平台申请获得，商户私钥，用于创建支付订单时数据签名 |
| 12 | PublicKey | 从Amigo平台申请获得，商户公钥，用于商户接收交易结果通知时的签名校验 |

# 接入步骤简介

**您需要完成以下工作：**

1. **接入前准备**

a)在金立游戏开发者平台<http://dev.game.gionee.com/login/login.html>注册账号。注册流程参见：<http://dev.game.gionee.com/help/page/id/9.html>

b)在游戏开发者平台<http://dev.game.gionee.com/login/login.html>联运游戏中申请获得APIKey、SecretKey、PrivateKey、PublicKey，申请密钥时，游戏PackageName必须以“.am”结尾。

c) 准备金立手机。金立SDK支持安卓4.0.0及以上版本。

金立手机适配：分辨率涵盖800\*480（或者854\*480）/1280\*720/1920\*1080。非金立机型适配：三星/小米/华为/OPPO/VIVO，分辨率涵盖1280\*720/1920\*1080。后续会增加适配机型。

d) 对照readme.txt检查文档是否齐全。SDK下载地址：<http://dev.game.gionee.com/help/sdk.html>

1. **客户端接入**

请按照SDK客户端接入，您需要完成：

1. SDK的初始化
2. 账号登录
3. 启用收银台
4. 处理支付结果
5. **服务端接入**

请按照服务器接入，您需要完成：

1. AmigoToken安全验证
2. Amigo支付接入

# SDK流程说明

## 3.1 登录流程概述

Amigo SDK提供了账号功能有：账户登录->账户安全验证。

账户登录具体实现请参考账号登录，安全验证实现请参考AmigoToken安全验证

**用户登录流程**



1. Amigo账号登录流程图

**Amigo账号安全验证时序**



1. 账号安全验证时序图

注意：对账号安全性要求不高时步骤4-7可选。

## 3.2 支付流程概述

网游版SDK提供的支付功能有：创建支付订单->启动收银台->通知支付结果

其业务时序图如下：



1. 支付时序图

注意：

a. 启动收银台前请确保订单已创建成功。

b. 支付结果请以第15步返回的结果为准。

# SDK客户端接入

请先把随本文档一起分发的Amigo\_Play\_SDK\_Demo程序安装到手机上。该demo完整演示了Amigo Play SDK的工作流程，有助于您快速理解我们 SDK 客户端整个流程。查看Amigo\_Play\_SDK\_Demo代码时请搜索并关注“开发者重点关注”部分。

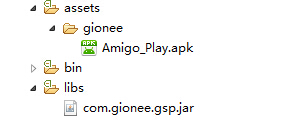
## 组件和权限声明

### 4.1.1 SDK接入所需环境

* 操作系统：Android 4.0.0 及以上版本。**（重要）**
* 开发工具：Eclipse & ADT，建议您将ADT的版本升级到最新的版本。
* 设备支持：Android 4.0及以上系统的手机。

### 4.1.2 添加SDK组件

将 zip 包中“工程组件”目录下的 “assets/gionee/Amigo\_Play.apk”文件拷贝到商户APK工程的 assets/gionee/ 目录下, 将“工程组件”目录下“libs/com.gionee.gsp.jar”包放到工程的 libs 目录下,并在 buildpath 中引用。



### 4.1.3 添加SDK所需要的权限

|  |
| --- |
| <!-- 开始添加Amigo Play SDK需要的权限，开发者重点关注 -->  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  <uses-permission android:name="android.permission.GET\_TASKS" />  <uses-permission android:name="android.permission.INTERNET" />  <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  <uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />  <uses-permission android:name="android.permission.CHANGE\_WIFI\_STATE" />  <uses-permission android:name="android.permission.READ\_PHONE\_STATE" />  <uses-permission android:name="android.permission.INSTALL\_PACKAGES" />  <uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" />  <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  <!-- 结束添加权限 --> |

### 4.1.4 添加SDK组件所需要声明

|  |
| --- |
| <!-- Amigo Play SDK组件声明开始 -->  <service android:name="com.gionee.gsp.floatingwindow.FloatingWindowService" >  <intent-filter>  <action android:name="com.gionee.pay.ic.FloatingWindowService" />  <category android:name="android.intent.category.DEFAULT" />  </intent-filter>  </service>  <activity  android:name="com.gionee.gsp.service.activity.AssistActivity"  android:configChanges="mcc|mnc|orientation|screenSize"  android:theme="@android:style/Theme.Translucent.NoTitleBar" />  <!-- Amigo Play SDK组件声明结束 --> |

## SDK初始化

在Application.onCreate方法中调用以下初始化方法

|  |
| --- |
| GamePlatform.init(Context context, String apiKey); |

**方法说明**：

* init方法必须在Application.onCreate方法中调用。

## 账号登录

正式账号登录成功后会返回用户信息，其中包含playerId 、AmigoToken等，商户服务器可以使用AmigoToken信息向账号服务器进行账号安全验证，详见AmigoToken安全验证。

|  |
| --- |
| //第二个参数默认传true表示自动登录，传false即为不自动登录  GamePlatform.loginAccount(mActivity, **true**, **new** LoginListener() {  @Override  **public** **void** onSuccess(AccountInfo accountInfo) {  // 登录成功，处理自己的业务。  // 获取pid  String playerId = accountInfo.mPlayerId;  // 获取amigoToken  String amigoToken = accountInfo.mToken;  }    @Override  **public** **void** onError(Exception e) {  Toast.makeText(mActivity, "登录失败:" + e,  Toast.LENGTH\_SHORT).show();  }    @Override  **public** **void** onCancel() {  Toast.makeText(mActivity, "取消登录",  Toast.LENGTH\_SHORT).show();  }  }); |

**成功登录结果会在onSuccess回调中返回，其结果数据说明如下：**

* playerId：游戏用户唯一标识符playerId；
* amigoToken：json字符串，用于AmigoToken安全验证请求的http body内容。

注意：从2.1.8.d版本开始，登陆成功接口只能获取到playerId。使用老版本SDK接入的游戏（即使用userId作为用户唯一标识的游戏），在升级SDK时一定不能再使用playerId做用户标识，否则会导致老用户数据丢失。SDK升级只需替换com.gionee.gsp.jar和Amigo\_Play.apk，不用修改代码。

## 启用收银台

此接口对应“支付时序图-第6步启动收银台”。

商户APK在收到商户服务器创建支付订单成功的消息后，启动支付收银台。

### 4.4.1 调用方式

注意：启动收银台前必须确认支付订单已创建成功（详见5.2.1 创建支付订单）。

调用接口**GamePayer.pay(OrderInfo orderInfo, GamePayCallback payCallback)**，具体使用可以参见Amigo\_Play\_SDK\_Demo中PayOnlineTestActivity。

参考代码：

|  |
| --- |
| //客户端支付结果回调  private GamePayCallback mGamePayCallback = mGamePayer.**new** GamePayCallback() {    @Override  **public** **void** onPaySuccess() {  // 可以在这里处理自己的业务  }  @Override  **public** **void** onPayCancel() {  // 可以在这里处理自己的业务  }  @Override  **public** **void** onPayFail(String stateCode) {  // 可以在这里处理自己的业务  }  };  //填充订单信息  OrderInfo orderInfo = **new** OrderInfo();  //开发者后台申请的Apikey  orderInfo.setApiKey(apiKey);  //商户订单号，与创建支付订单中的"out\_order\_no"值相同  orderInfo.setOutOrderNo(outOrderNum);  //支付订单提交时间，与创建支付订单中的"submit\_time"值相同  orderInfo.setSubmitTime(submitTime);  //调用支付接口  mGamePayer.pay(orderInfo, mGamePayCallback); |

### 4.4.2 OrderInfo 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属性 | 是否必填 | 说明 |
| mApiKey | 字符串 | 必填 | 创建该支付订单的APIKey |
| mOutOrderNo | 字符串 | 必填 | 商户订单号，与创建支付订单请求时相同 |
| mSubmitTime | 字符串 | 必填 | 订单提交时间，格式为yyyyMMddHHmmss。**与创建支付订单的“submit\_time”字段是同一个值。** |

### 4.4.3 支付失败返回状态码

|  |  |
| --- | --- |
| 状态码 | 状态码含义 |
| 4000 | Amigo支付系统异常 |
| 4001 | 数据格式不正确 |
| 4002 | 空间不足，升级失败 |
| 4003 | 账号安全级别低，升级失败 |
| 4006 | 订单支付失败 |
| 6000 | 支付服务正在进行升级操作或异常 |
| 6002 | 网络连接异常 |

# 服务器接入

SDK包中包含了服务器接入demo，其中Amigo\_Play\_Account为AmigoToken验证demo，Amigo\_Play\_Server是支付demo。接入前请先运行服务器demo。

服务器demo帮您实现了您需接入的所有功能，您只需参照本文档了解接口使用，即可快速完成接入。

## AmigoToken安全验证

### 5.1.1 请求安全验证

此接口对应“账号安全验证时序图—步骤5请求Token验证”。

具体实现代码请参照Amigo\_Play\_Account文件中的Account\_Verify\_Demo\_Test.java。

#### 5.1.1.1 请求接口及参数说明

1. 请求接口：https://id.gionee.com/account/verify.do
2. 请求方式：POST
3. 请求类型：application/json
4. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属性 | 是否必填 | 说明 |
| id | 字符串 | 必填 | 商户申请的APIKey（不参与mac签名） |
| ts | 字符串 | 必填 | 是距离1970-01-01 00:00:00 GMT 的秒数。（参与mac签名） |
| nonce | 字符串 | 必填 | 随机字符串。用于HmacSHA1签名算法。（参与mac签名） |
| mac | 字符串 | 必填 | 采取HmacSHA1方式进行mac签名, 此处需用到申请的SecretKey。签名的方法具体可见Amigo\_Play\_Account文件中的Account\_Verify\_Demo\_Test.java |
| AmigoToken  其组成格式:  {  "h":"assoc-handle",  "t":"assoc-timestamp",  "n":"assoc-nonce",  "v":"assoc-signature" } | 字符串 | 必填 | 从客户端登陆返回结果中获取，直接将获取的值作为请求体即可。获取方法详见“账号登录”登录返回结果。（不参与mac签名） |

#### 5.1.1.2 请求参数示例

|  |
| --- |
| POST /account/verify.do HTTP/1.1 Host: id.gionee.com Authorization: MAC id="D4D91AA8707C4201B09A108E6156CB14",  ts="1386832975",  nonce="9444729f",  mac="XorG3EMLBz9VLwwvFEq4lJNop1o=" Content-Type: application/json Content-Length: message-size {  "h":"D534527CF4D94E309B02B88DDF52BF37",  "t":"1386834370",  "n":"70B12025",  "v":"33BB1B199EE3966965BBBE865B018096D2CB8528" }    注意事项：  由字段id、ts、nonce、mac组成的一个明文字符串（不是json串），作为MAC键标示符。本信令必须采用HTTPS。 |

### 5.1.2 返回验证结果

此接口对应“账号安全验证时序图—步骤6返回验证结果”。

验证接口返回的是json格式数据，如果里面不包含“r”参数或“r”值为0，则认为验证成功，否则验证失败。

#### 5.1.2.1 返回参数说明（未说明字段不用关心）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属性 | 是否必填 | 说明 |
| r | 字符串 | 可选 | 账号安全验证返回状态码。失败时才出现，具体含义请见“5.1.2.3返回状态码说明” |
| wid | 字符串 | 可选 | 失败时才出现，为UUID字符串，用于错误跟踪，见 “返回状态码说明” |

#### 5.1.2.2 返回参数示例

|  |
| --- |
| HTTP/1.1 200 OK Content-Type: application/json Content-Length: message-size {  "r":"1011",  "wid":"38c6aa98-b220-49e5-94f5-fab0f8d00838",  "u":"9FBDC73FC6734B0BBCF9AF208A453F13",  "tn":"15311521854",  "na":"Amigo\_你好",  "ptr":" ",  "ul":30,  "sty":0,  "ply":[{  "a":"D4D91AA8707C4201B09A108E6156CB14",  "pid":"877CC9677D8248D1872EC41BCC11B26E",  "na":"Amigo\_21739"   }] } |

#### 5.1.2.3 返回状态码说明

|  |  |  |  |
| --- | --- | --- | --- |
| 状态码 | 状态码含义 | err参数 | 详细描述及解决办法 |
| 1007 | 未知错误 | 无 | 提供响应的r和wid值Amigo服务端分析 |
| 1009 | HTTP格式错误 | 必须提供Authorization认证头 | 必须提供Authorization请求头 |
| 1010 | 信令格式错误 | 验证头必须提供MAC标识 | 请确保认证头使用MAC标识 |
| 没有提供令牌 | 检查请求体是否传了令牌 |
| 令牌格式被修改 | 不能对令牌作任何修改 |
| 必须提供id参数 | 提供认证头的id参数 |
| 必须提供ts参数 | 提供认证头的ts参数 |
| 必须提供nonce参数 | 提供认证头的nonce参数 |
| 必须提供mac参数 | 提供认证头的mac参数 |
| 必须提供Authorization认证头 | 提供认证头 |
| 1011 | 认证失败 | 不存在该应用(APIKey) | 调用此接口的第三方应用不存在，请检查验证头的id参数是否为申请的APIKey值 |
| 签名不匹配，请检查你的签名 | 调用方提供的mac签名参数与服务端算出的不一致，请检查生成签名的参数和算法是否正确，详见FQA |
| 不存在对应的用户 | Token对应的用户在系统中不存在，请检查该Token是否为生产环境生成。 |
| 令牌不存在(或者超时被清理) | 被验证的Token已经过期并被清查或者所分配的令牌并不是生产环境生成的。 |
| 验证令牌的应用与生成令牌的应用不是同一个 | 请确保获取和验证Token的应用是同一个 |
| 验证令牌的时间戳与生成令牌时的时间戳不相同 | 令牌的时间戳(ts)可能被修改 |
| 验证令牌的随机数与生成令牌时的随机数不相同 | 令牌的随机数(nonce)可能被修改 |
| 验证令牌的签名与生成令牌时的签名不相同 | 令牌的签名(mac)可能被修改 |
| 验证令牌的次数大于最大限制，一般为一次有效 | 令牌只能被验证一次，不能重复使用 |
| 被验证的令牌已经过期 | 被验证的Token已经过期，过期时间是5分钟 |
| 令牌对应的应用不存在 | 被验证的Token对应的应用并不存在，请检查获取该Token的来源 |
| 令牌本身的签名验证失败，令牌被修改 | 被验证的Token本身签名自检不匹配，Token可能被修改 |
| 1020 | 服务器内部错误 |  | 请提供r和wid参数给Amigo服务端分析 |
| 1031 | 发送超时 |  | 请提供r和wid参数给Amigo服务端分析 |
| 1050 | MAC签名的timestamp已过期 | 无 | 第三应用调用来接口的签名(mac)已经过期，请确保签名有效，签名有效期为5分钟。如遇服务器间时间不同步也可能发生此情况，HTTP响应头的Date参数可以用于纠正时间差 |
| 1051 | MAC签名的nonce重复 | 无 | 二次调用，nonce参数不能重复，请使用随机数 |

## Amigo支付接入

### 5.2.1 创建支付订单

此接口对应“支付时序图—步骤3创建支付订单”。

由商户服务器发起向Amigo Play Server请求创建支付订单，具体接口实现可以参照Amigo\_Play\_Server\_Demo中AmigoPlayDemo类。

#### 5.2.1.1 请求接口及参数说明

1. 请求接口：https://pay.gionee.com/order/create
2. 请求方式：POST
3. 请求类型：application/json
4. 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属性 | 是否必填 | 说明 |
| player\_id | 字符串 | 必填 | 用户唯一标识playerId（不参与签名） |
| api\_key | 字符串 | 必填 | 商户申请的APIKey |
| deal\_price | 字符串 | 必填 | 商品总金额 |
| deliver\_type | 字符串 | 必填 | 付款方式：1为立即付款，2为货到付款  **(目前支付方式1，请选1)** |
| expire\_time | 字符串 | 可选 | 订单过期时间，必须大于订单提交时间submit\_time，格式为yyyyMMddHHmmss(2009年12月27日9点10分10秒表示为20091227091010)，时区为GMT+8 beijing。如果不设，订单默认有效期为10年。**如果有该字段，必须参与签名** |
| notify\_url | 字符串 | 可选 | 自定义服务器通知地址，可以后台配置，也可在创建订单时携带，不能超过1024个字符。(如果商户需要自定义参数，可以在创建订单时以"http://www.partner.com/notifyReceiver?para  m1=value1&param2=value2"的形式定义URL，自定义参数不需要参与签名，支付服务器会在notify通知时原样返回)。**如果有该字段，必须参与签名** |
| out\_order\_no | 字符串 | 必填 | 商户订单号，由商户自定义，数字、字母或"-"、"\_"、"|"构成，64个字符以内。商户需要确保订单号的唯一性，不能重复 |
| subject | 字符串 | 必填 | 商品名称，32个字符以内，不能含有半角“+”、“&”或特殊字符集 |
| submit\_time | 字符串 | 必填 | 订单提交时间，格式为yyyyMMddHHmmss(2009年12月27日9点10分10秒表示为20091227091010)，时区为GMT+8 beijing。该时间需由商户服务器提供 |
| total\_fee | 字符串 | 必填 | 需支付金额，值必须等于商品总金额（deal\_price字段） |
| sign | 字符串 | 必填 | 使用RSA签名算法。签名所需字段不包括sign、player\_id。规则详见下一小节 |

#### 5.2.1.2 订单签名规则

首先，将参数值按照参数名（不包括sign、player\_id字段）按字母顺序排序（如果第一个字母相同，按第二个字母排序，依次类推），再将参数值直接连接，得到待签名字符串。

然后，使用PrivateKey（商户私钥）对待签名字符串进行签名，即为我们所需的sign字段值。

最后，将其写入sign 字段即可。

示例：

|  |
| --- |
| 1. 待签名原始数据：   api\_key=FF20A4B3BCD44F1380DEECADFC6657D3  deal\_price=0.01  deliver\_type=1  out\_order\_no=201311290000005  subject=元宝充值  submit\_time=20131210154013  total\_fee=0.01   1. 拼装待签名内容signContent：   “api\_key值”+“deal\_price值”+“deliver\_type值”+“out\_order\_no值”+“subject值”+“submit\_time值”+“total\_fee值”  规则：将参数值按照参数名（不包括sign、player\_id字段）按字母顺序排序，再将参数值直接连接，得到以下待签名内容signContent：  FF20A4B3BCD44F1380DEECADFC6657D30.011201311290000005元宝充值201312101540130.01   1. 生成签名sign字段（具体方法见Amigo\_Play\_Server\_Demo中PayUtil.java）：   sign = RSASignature.*sign*(signContent.toString(), *private\_key*, CharEncoding.*UTF\_8*); |

注意：拼装待签名内容signContent时，无需&连接，并且只是参数值连接无需参数名。

#### 5.2.1.3 请求参数示例

|  |
| --- |
| {"sign":"PnKNWYuXc2uZxde3IRgYzHId\/CZ3sfTBmnuhXtEb3qzkMpnc7rscIBo\/aOgN9SBkpj0bbC5GkXsGn8irTWdtyENMq5ZoS1HJZo2PkeNb1AN5hm1bTwbdWWMgoWMZxxY8IoCeolS8vvXUha6LwNrJyjPtxxe\/sAz\/QbSuYmcP6Gg=","deliver\_type":"1","total\_fee":"0.01","subject":"testPaymobile","api\_key":"DDFDAEC3DBF544DD99EB9F508B429905","out\_order\_no":"50b36e69c5ff1386670127535","player\_id":"amigo\_player001","submit\_time":"20140403172737","deal\_price":"0.01"} |

注意：所有参数都是以字符串类型提交。特别注意total\_fee、deal\_price、deliver\_type、submit\_time、expire\_time字段值都需转换为String类型。

### 5.2.2 返回订单创建结果

此接口对应“支付时序图—步骤4返回订单创建结果”。

#### 5.2.2.1 返回参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属性 | 是否必填 | 说明 |
| status | 字符串 | 必填 | 返回的状态码，详见“5.2.2.3返回状态码说明” |
| description | 字符串 | 必填 | 描述信息，与状态码对应 |
| order\_no | 字符串 | 必填 | 支付订单号，创建失败时返回空，  请注意：**这个是后续对账用的，现在暂时没用到，这个参数不是调起收银台的订单参数。** |
| api\_key | 字符串 | 必填 | 创建该支付订单的APIKey |
| out\_order\_no | 字符串 | 必填 | 商户订单号，与创建支付订单请求时相同，**请以这个订单号来调起收银台。** |
| submit\_time | 字符串 | 必填 | 订单提交时间，格式为yyyyMMddHHmmss(2009年12月27日9点10分10秒表示为20091227091010)，时区为GMT+8 beijing。该返回值同请求创建支付订单的“submit\_time”字段值一致 |

#### 5.2.2.2 返回参数示例

|  |
| --- |
| {"status":"200010000","description":"成功创建订单","out\_order\_no":"201311290000005","api\_key":"DDFDAEC3DBF544DD99EB9F508B429905","submit\_time":"20131210154013","order\_no":"52a690d10cf223aa543968e0"} |

#### 5.2.2.3 返回状态码说明

|  |  |  |
| --- | --- | --- |
| 状态码 | 状态码含义 | 详细描述及解决办法 |
| 200010000 | 成功创建订单 | 属于正常状态 |
| 400000000 | 请求参数格式错误 | 订单JSON格式错误，检查字段是否已转换为string类型 |
| 400010001 | 参数格式错误，subject为空 |  |
| 400010002 | 含有非法字符+& | 检查subject字段是否含有非法字符 |
| 400010003 | 无效的订单，没有交易额 | 检查total\_fee字段是否是有效值 |
| 400010004 | 签名错误，信息校验失败 | 解决办法详见FQA |
| 500000000 | 服务器繁忙，请稍后再试 |  |
| 400010006 | 商户已被冻结暂无法创建订单 |  |
| 400010007 | 参数格式错误，subject长度大于32个字符 |  |
| 400010008 | 参数格式错误，api\_key为空 |  |
| 400010009 | 参数格式错误，sign为空 |  |
| 400010010 | 参数格式错误，绑定账号订单，user\_id为空 |  |
| 400010012 | 参数格式错误，out\_order\_no格式不正确 |  |
| 400010013 | 参数格式错误，out\_order\_no的长度大于32个字符 |  |
| 400010014 | 时间格式错误，submit\_time格式不正确 |  |
| 400010015 | 参数值非法，deliver\_type值不支持 |  |
| 400010016 | 参数格式错误，total\_fee格式不正确 |  |
| 400010017 | 参数格式错误，deal\_price格式不正确 |  |
| 400010019 | 时间格式错误，expire\_time格式不正确 |  |
| 400010020 | 时间值无效，expire\_time需要大于submit\_time |  |
| 400010022 | 订单金额不匹配，total\_fee与deal\_price不相等 |  |
| 400010023 | 参数格式错误，notify\_url长度大于1024个字符 |  |
| 400010024 | 参数格式错误，deal\_price需要大于0 |  |
| 400010025 | 参数格式错误，deliver\_type为空 |  |
| 400010026 | 参数格式错误，不存在该商户 | 检查api\_key传值是否正确 |
| 400010100 | 订单号已经存在 |  |

### 5.2.3 通知商户支付结果

此接口对应“支付时序图—步骤15后台通知商户支付结果”。

交易成功后，Amigo Play Serve会将交易结果以POST 方式传递至商户NotifyURL，商户服务器需要维护并校验该订单的状态。

注意：Amigo Play Server只将支付成功的订单进行返回，未成功订单不会有通知；商户服务器需要确保NotifyURL 稳定可靠。

具体使用方法可参照Amigo\_Play\_Server\_Demo中OrderNotify类。

#### 5.2.3.1 接口介绍

1. 通知接口：商户提供NotifyURL (例如：<http://www.partner.com/notifyReceiver>)
2. 通知方式：POST
3. 通知类型：application/x-www-form-urlencoded
4. 通知参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 属性 | 是否必填 | 说明 |
| api\_key | 字符串 | 必填 | 商户APIKey |
| close\_time | 字符串 | 必填 | 支付订单关闭时间，格式为yyyyMMddHHmmss(2009年12月27日9点10分10秒表示为20091227091010)，时区为GMT+8 beijing。 |
| create\_time | 字符串 | 必填 | 支付订单创建时间，格式为yyyyMMddHHmmss(2009年12月27日9点10分10秒表示为20091227091010)，时区为GMT+8 beijing。 |
| deal\_price | 浮点数 | 必填 | 商品总金额 |
| out\_order\_no | 字符串 | 必填 | 商户订单号 |
| pay\_channel | 整数 | 必填 | 用户支付方式(A币支付：100) |
| submit\_time | 字符串 | 必填 | 商户提交订单时间，格式为yyyyMMddHHmmss(2009年12月27日9点10分10秒表示为20091227091010)，时区为GMT+8 beijing。该返回值与请求创建支付订单的“submit\_time”字段值一致 |
| user\_id | 字符串 | 必填 | 返回null |
| sign | 字符串 | 必填 | 以上字段按照字母顺序都参与签名，同订单创建时签名。此字段需签名验证，具体详见5.2.4.1 签名验证。 |

#### 5.2.3.2 通知参数示例

|  |
| --- |
| close\_time=20140403174902&out\_order\_no=50b36e69c5ff1386670127535&sign=A1nItIRYJoKyfL35Evq1mgTd%2FhRLwxF2mMKijwXCqjz1PYR%2F7Wq8l0pQxSMAL%2B%2BHF5RF5i865u2JCI0xv7GX4mGMDbaZGe8Z9wBbfrYR5MjGDOmtHLWBM1WnKeyKSPXs1yIdPXNzSB4NEwjq1VFijh1FahDXJJUBexTBqamBzgE%3D&api\_key=DDFDAEC3DBF544DD99EB9F508B429905&create\_time=20140403172917&user\_id=null&pay\_channel=100&submit\_time=20140403172737&deal\_price=0.01 |

### 5.2.4 商户返回支付结果确认

商户服务器收到支付结果通知后，需进行签名验证并给Amigo服务器响应，示例代码请参见Amigo\_Play\_Server\_Demo的OrderNotify.java中doPost函数实现。

#### 5.2.4.1 签名验证

1. 签名验证规则

先将接收到的参数key-value对按照key（不包括sign 字段）按照字符升序排序，以“key=value”形式用“&”拼接(sign字段不参与)，得到待签名字符串。然后使用PublicKey对待签名字符串进行验签。

示例：

|  |
| --- |
| 1. 获取原始数据：   api\_key=FF20A4B3BCD44F1380DEECADFC6657D3  close\_time=20131210154043  create\_time=20131210154015  deal\_price=0.01  out\_order\_no=201311290000005  pay\_channel=100  submit\_time=20131210154013  user\_id=null   1. 拼装待验签内容checkContent：   “api\_key键=api\_key值”+“&”+“close\_time键=close\_time值”+“&”+“create\_time键=create\_time值”+“&”+“deal\_price键=deal\_price值”+“&”+“out\_order\_no键=out\_order\_no值”+“&”+“pay\_channel键=pay\_channel值”+“&”+“submit\_time键=submit\_time值”+“&”+“user\_id键=user\_id值”  规则：按照key的ASCII升序排序，以“key=value”形式用“&”拼接，得到以下待验签内容checkContent：  api\_key=FF20A4B3BCD44F1380DEECADFC6657D3&close\_time=20131210154043 &create\_time=20131210154015&deal\_price=1.01&out\_order\_no=2013112900000054317&pay\_channel=100&submit\_time=20131210154013&user\_id=null   1. 取sign字段值：   String sign = req.getParameter("sign");   1. 验证签名（具体方法见Amigo\_Play\_Server\_Demo中OrderNotify.java）：   RSASignature.*doCheck*(checkContent, sign, *public\_key*, CharEncoding.*UTF\_8*); |

1. 注意事项

拼凑原始值，不要人为引入空格。

#### 5.2.4.2 响应确认结果

http响应返回的body内容为纯字符串success（注意不能包含html等其他任何语言标记、空格或换行符）。

应答参数示例：

|  |
| --- |
| success |

注意：notify 通知最多发送7次，间隔2分钟，在收到商户success应答后即停止。

# 其他问题

## 6.1 APK打包

1．游戏的PackageName必须以“.am”结尾，必须和申请密钥的包名一致。若忘记，可查看密钥文档中的包名，或者去开发者平台中查看申请的包名。

## 6.2 横竖屏切换

当应用的某个activity 调用Amigo Play SDK时，窗口会根据当前应用的屏幕状态自动适配，无需应用做任何操作。

## 6.3 代码混淆

如果你的apk有代码混淆，请添加以下忽略信息：

-keep public class \* extends android.os.IInterface

-keep public class \* extends android.os.Binder

-keep class com.youju.statistics.\*\*{\*;}