

# 小伙伴游戏平台——游戏接入文档

## 名词定义

- 小伙伴游戏平台 在后续描述中统一简称 平台，

## 1. 接入流程

### 1.1 申请游戏开发商

请向 平台 商务人员提交游戏相关信息，并获得相应的game\_key和 game\_secret。

请咨询和您对接的 平台 人员，拿到 平台 提供的游戏URL地址, 地址形式如下：`http://gc.hgame.com/home/game/appid/xxx/gameid/xxx/review/1`

只有在拿到这个地址后，才可以进行后续的操作。

### 1.2 初始化操作

在游戏调用接口前需要初始化 平台 SDK。

#### 1.2.1 加载js-sdk

下载 `hgame-sdk.js`，将其加载到游戏页面上。

示例：

```
<script type="text/javascript" src="http://f.hgame.com/js/hgame-sdk.js"></script>
```

#### 1.2.2 初始化设置

功能说明

设定game\_key

参数说明：

- game\_key**：这里是 平台 提供的game\_key

示例：

```
<script language="javascript">
    var hGame = new hGame({
        "game_key": 'demo-game-2'
    });
</script>
```

## 2. 接口说明

### 2.1 认证

获取 平台 当前登录的用户

步骤：

- 平台 调用游戏首页，在打开游戏首页的过程中，ticket等参数会直接附加在游戏URL地址里面
- 游戏在拿到ticket等参数后，向 平台 验证ticket有效性，并获取登录用户ID( `open_id` )

#### 2.1.1 初始化游戏首页时， ticket参数等说明

参数说明：

- game\_key**：这里是 平台 提供的game\_key（required）
- timestamp**：时间戳（required）
- nonce**：随机字符串（required）
- login\_type**：始终为1，表示ticket认证（required）
- ticket**：认证票据，用于获取平台唯一用户，只能使用一次（required）
- signature**：签名（required）

在拿到上述参数后，按照文档中的签名机制用game\_secret进行签名，然后验证签名是否正确即可。

示例：

```
请求URL： http://gamedemo.com/index.php?game_key=demo-game-2&timestamp=14332332&nonce=rand232&login_type=1&ticket=afd333&signature=222
```

2.1.2 验证登录ticket，并获取openId

步骤

- 1. 游戏服务端向小伙伴游戏平台服务端验证登录票据，并获得平台唯一用户ID

请求地址：

```
http://gc.hgame.com/user/getticketuserinfo
```

参数说明：

- **game\_key**： 这里是游戏中心提供的game\_key（required）
- **timestamp**： 游戏方生成的时间戳（required）
- **nonce**： 游戏方生成的随机字符串（required）
- **login\_type**： 始终为1，表示ticket认证（required）
- **login\_ticket**： 注意：与获取时ticket参数名称不同，认证票据，用于获取平台唯一用户，只能使用一次（required）。值就是上一步里面获取的ticket
- **signature**： 签名（required）

上述参数按照文档中的签名机制用game\_secret进行签名

示例：

```
请求URL： http://gc.hgame.com/user/getticketuserinfo?game_key=demo-game-2&timestamp=14332332&nonce=rand232&login_type=1&login_ticket=afd333&signature=222
```

返回格式：

```
1 {
2   "code":0,           // 0=成功=0 非0=失败
3   "message":"操作成功", // 错误描述
4   "data":{"
5     "open_id":"aaa33dd32dd2dadd"
6   }
7 }
```

返回Code说明：

Code	说明
0	成功
-1	ticket不存在或过期
-2	禁止用户登录
-3	游戏已下架
验证错误	请参照签名说明

2.2 购买

购买游戏内的道具或金币。

步骤：

- 1. 由游戏服务器生成购买订单，并调用hGame.pay方法，发起订单
- 2. 用户通过平台完成付费
- 3. 平台 回调游戏提供的notify\_url，通知游戏付费成功

2.2.1 发起购买订单

功能说明：

调用小伙伴游戏平台的接口购买游戏内道具

参数说明：

- **pay\_data**：支付信息

支付信息**pay\_data**参数说明(游戏方服务端生成):

- **game\_key**：这里是游戏中心提供的game\_key（required）
- **open\_id**：小伙伴游戏平台提供的用户id（required）
- **total\_fee**：道具支付金额（单位元，精确到小数点后两位，（required）
- **game\_orderno**：游戏生成的订单号（required，唯一）
- **subject**：游戏道具名称（required）
- **description**：游戏道具描述（option）
- **notify\_url**：支付完成后通知URL（required）
- **timestamp**：时间戳（required）
- **nonce**：随机字符串（required）
- **game\_area**: 用户所在的游戏区（option）
- **game\_group**: 用户所在的游戏服（option）
- **game\_level**: 用户在游戏等级中的等级（option）
- **game\_role\_id**: 用户的角色Id（option）
- **signature**：签名（required）

**payType**说明:

**payType** 是指游戏指定的优先支付方式，可以为空值";

由于客户端操作系统（IOS需要使用苹果的支付系统）的不同，以及是否安装应用（比如支付宝，微信等）的差异，客户端可用的支付方式是有限的。这里的优先指对不同的客户端实际可用的支付方式中优先选择游戏方指定的。

**PayType** 合法值：

说明	值
支付宝移动端网页支付方式	alipay_wap
支付宝PC端网页支付方式	alipay_web
支付宝SDK支付方式（安卓）	alipay_sdk
IOS支付	pay_apple

上述参数准备完成后，按照文档中的签名机制用game\_secret进行签名

示例:

PHP完成数据签名组装, 输出为js的json对象:

```
<?php
$data = array(
    "game_key"           => 'demo-game-2',
    "open_id"            => 'abcdefg123456',
    "total_fee"          => '1.00',
    "game_orderno"       => 'order_1233445',
    "subject"            => '道具名称',
    "description"        => '道具描述',
    "notify_url"=> 'http://gamedemo.com/pay_confirm.php'
);
$data = signTheData($data); //使用签名方法签名
?>
var pay_data = <?php echo json_encode($data);?>;
```

前台调用:

```
<script language="javascript">
    // 客户端通过hgame-sdk.js发起支付请求
    // hGame.pay(payData, payType, callback)
    // payData: 支付数据, 参见前面的定义
    // payType: 优先支付方式, 参见前面的定义
    // 支付成功回调函数
    hGame.pay(payData, payType, function (result)
    {
        //result结构如下
        result.code      // 0=成功, -1=取消, 其他<0=失败
        result.message   // 错误描述
        result.showMessage // 显示提示
        result.data      // 返回数据
    });
</script>
```

result返回字段说明

字段	说明
code	0: 表示成功, -1: 表示取消, 其他<0: 表示失败
message	错误描述
showMessage	用于显示用的错误信息, 如果没有showMessage, 将使用message来显示
data	返回数据, 具体格式: ?

2.2.2 用户完成付费

用户通过 平台 提供的支付功能, 在支付宝、银联、Apple、微信等支付平台完成付费。

2.2.3 服务端支付成功通知

功能说明:

- 用户支付成功后 (只有用户支付成功的情况下, 平台 才会通知游戏方), 平台 会向游戏方下单时指定的notidy\_url发送支付成功通知。
- 平台 在没有接收到游戏方返回的 成功标志 时 会 一直重复 发支付成功通知, 在交易完成的第 (0min,2min,10min,20min,1hour,2hour,6hour,15hour,24hour)时段发起请求, 直至收到成功标志或者超时。

参数说明:

- **game\_key** : 这里是 平台 提供的game\_key (required)
- **game\_orderno** : 游戏方生成的唯一订单号 (required)
- **orderno** : 平台 生成的唯一订单号 (required)
- **subject** : 游戏道具名称 (required)
- **description** : 游戏道具描述 (option)
- **total\_fee** : 商品价格 (required)
- **signature** : 签名 (required)

上述参数按照文档中的签名机制(参见第三章 签名机制 )和game\_secret进行签名

请求示例:

请求URL: http://gamedemo.com/pay\_confirm.php , 这个是上一步里面设置的回调地址

平台 发起的POST参数:

game\_key=demo-game-2&game\_order\_no=order\_1233445&orderno=5454asd8as445456465&subject=道具名称&description=道具描述&total\_fee=1.00&signature=

返回示例:

```
1 {
2   code:0, // 0: 表示游戏支付成功, 非0: 表示失败
3   message: "游戏支付成功"
4 }
```

Code	说明
0	游戏支付成功
-1	用户不存在
-2	订单重复
-3	订单不存在

2.3 分享

将游戏的基本信息分享到微信、朋友圈、微博等社交平台。

功能说明:

调用 平台 的接口分享游戏基本信息到社交媒体

参数说明:

- **message** : 需要分享的消息

示例:

```
1 <script language="javascript">
2   var message = '这里游戏可以自己定义分享的消息';
3
4   // 分享的图片和分享地址为游戏登记时的内容
5   hGame.share(message, function (result)
6   {
7       result.code      // 0: 表示成功, -1: 表示取消, 其他<0: 表示失败
8       result.message    // 错误描述
9       result.showMessage // 用于显示用的错误信息, 如果没有showMessage, 将使用message来显示
10      result.data       // 返回数据
11  });
12 </script>
```

result返回字段说明

字段	说明
code	0: 表示成功, -1: 表示取消, 其他<0: 表示失败
message	错误描述
showMessage	用于显示用的错误信息, 如果没有showMessage, 将使用message来显示
data	返回数据, 具体格式: ?

2.4 游戏数据上报

游戏上报相关数据到小伙伴

游戏可实现自动或由用户触发上报相关数据

功能说明:

- 游戏上报角色 创建 数据时使用基础数据( `baseData` )格式
- 游戏上报角色 登录 数据时使用基础数据( `baseData` )格式
- 游戏上报角色 升级 数据时使用基础数据( `baseData` )及等级数据( `extendData` )
- 游戏上报角色 进度 数据时使用基础数据( `baseData` )及进度数据( `extendData` ), 比如第几关
- 游戏上报角色 分数 数据时使用基础数据( `baseData` )及分数数据( `extendData` ), 比如多少分 (总体)

方法名 `hGame.gameReport(action, baseData, extendData, callback)`

参数说明:

- **action:** 上报数据动作类型
- **baseData:** 基础数据, 包含时间, 角色ID, 区,服
- **extendData:** 不同上报动作的数据
- **callback:** 回调函数 `function(data){ ... }`

action合法值:

说明	值
角色创建	createRole
角色登录	enterGame
角色升级	levelUpgrade
角色进度	processReport
角色分数	scoreReport

baseData数据格式说明:

```
1 var baseData = {
2   "game_key": '1234567890',    //游戏平台提供的game_key
3   "role": 'xhb',              //游戏角色的唯一ID
4   "area": '1区' ,             //游戏区标志
5   "group": '1服'              //游戏服务器标志
6 }
```

extendData数据格式说明:

```
1 //上报角色升级扩展数据
2 var extendData = {
3     "level": 12
4 }
5
6 //上报角色进度扩展数据
7 var extendData = {
8     "progress": 'D区第一关',
9 }
10
11 //上报角色分数扩展数据
12 var extendData = {
13     "score": '123456',
14 }
```

示例:

```
1 var baseData = {
2     "timestamp": 1412345678, //时间戳
3     "role": 'xhb',           //游戏角色的唯一ID
4     "server": '1-1'          //游戏区服标志
5 }
6
7 var extendData = {
8     "level": 12
9 }
10
11 hGame.gameReport('levelUpgrade', baseData, extendData, function(data){
12     //这里是回调函数
13     console.log(data);
14 });
```

## 3. 签名机制

签名使用SHA1方式

格式为 `sha1(签名字符串+game_secret)`，并使用参数signature传递。

### 3.1 签名方法

在请求参数列表中，除去 signature 参数外，其他需要使用到的参数皆是要签名的参数。

GET方法为query中所有参数；POST方法为Form中所有的参数

### 3.2 示例

#### 3.2.1 php示例

```
1 //签名方法
2 function signTheData($data, $secret)
3 {
4     ksort($data);
5     foreach ($data as $k => $v) {
6         $tmp[] = $k . '=' . $v;
7     }
8     $str = implode('&', $tmp) . $secret;
9     $data['signature'] = sha1($str);
10    return $data;
11 }
```

### 3.3 签名错误

保证系统一致性，由于签名验证产生的错误，做以下规范

Code	说明
-9011	签名验证错误
-9012	缺少参数: signature, timestamp, nonce, game_key
-9013	timestamp 超过有效期
-9014	nonce 重复
-9015	没有对应的 secret_key