

360 秒开游戏中心 H5 游戏接入文档

Version : 1.0.0

360H5 游戏中心开放平台

2017/3/13

目录

修订记录.....	1
项目描述:	2
1. JS 引入方法	2
2. 初始化配置（必接）	3
2.1 加载 JSsdk.....	3
2.2 初始化设置.....	3
2.3 初始化后执行的方法.....	4
3. 游戏登录与获取用户信息（必接）	4
3.1 游戏登录校验签名方法.....	5
3.2 获取用户信息.....	5
4. 支付配置（必接）	6
4.1 订单发起和调用支付.....	6
4.2 支付成功通知和签名校验（服务端接口）	8
附录 1: 游戏签名方法.....	9

修订记录

版本号	修改时间	修订人	修订内容
1.0.0	2017/3/13	杨中意	1. 增加登录相关接入内容； 2. 增加支付相关内容； 3. 增加退弹，悬浮窗相关内容（不需要游戏接入）；

项目描述:

此项目为 360H5 游戏中心提供的 JSsdk 用于游戏接入。游戏接入 sdk 后,联系对接商务。上传对应游戏链接和展示 icon 等信息后,即可入库 360H5 游戏中心。此 sdk 主要辅助产品上线后的使用 360 渠道的登录系统,支付系统和对应的辅助工具。下文会对具体的接入模块做详细介绍,需要注意的问题有:

1. 游戏上线前,商务同事会分配游戏接入 sdk 所用的 game_id, game_key, game_secret 等参数。此部分参数需要应用在支付验参等接口上, **请游戏接入前务必了解自己游戏的对应参数。其中 game_secret 请勿泄露。**
2. 用户打开游戏地址后, 360 平台会根据用户的身份状态,提示用户完成必要的登录流程,之后 iframe 引入游戏地址,同时会在游戏地址上面附上 360 用户信息参数和签名,请游戏获取参数,验证签名, **绑定用户对照关系后进入游戏(详见登录部分)**
3. 支付部分允许游戏的接入工程师填写不同的支付回调地址, **务必使用自己服务器的支付回调地址**,回调地址为正常的 http 80 服务为佳,接口应稳定可靠响应快,避免游戏上线后充值失败 **(详见支付部分)**
4. 用户调用支付后,游戏中心会通过回调地址调用游戏服务端用于道具发放。游戏方需要校验支付签名,并通过自己的订单号校验我方提供的请求校验用户 id,商品名称和对应金额。避免被恶意篡改。 **(详见支付部分)**
5. 游戏方收到我方支付回调后,校验完成后,需要根据支付信息发放对应道具。 **并根据方法向我方返回 ok**,避免我方重复调用。360 不排除会多次调用通知接口,请勿重复发送道具给用户。 **(详见支付部分)**

1. JS 引入方法

概况如图(示例为调用支付的参数示例,在实际环境中,应该是通过 cp 自身接口调用获取):

```
<script type="text/javascript" src="http://api.h5.u.360.cn/tool/config/sdk?v=1"></script>
<script type="text/javascript">
    window.mk.ready(function(){
        window.mk.buy({
            game_key: "1", // 应用在开发者平台配置的值
            plat_user_id: "plat_user_id_3333", // 平台颁发给CP的用户账号
            game_name: "游戏名字",
            amount: 634, // 商品价格, 单位 (元)
            product_name: "商品名称",
            product_desc: "商品描述",
            notify_url: "/pay-result.html", // 支付成功后, 服务器会将支付信息通知游戏服务器
            order_id: "order_id_123", // 订单编号
            sign: "sign_d3i3", // 签名, 防刷订单
            timestamp: parseInt(new Date().getTime() / 1000) // 时间戳
        });
    });
</script>
```

图一 引入 SDK

2. 初始化配置（必接）

2.1 加载 JSsdk

在游戏页面加载 SDK:

正式服务器: `//api.h5.u.360.cn/tool/config/sdk?v=1`; (注意: 不需要添加 http, 以便自适应 http 和 https 协议)

目前 360 仅提供 http 服务. 请在游戏页面加载后, 尽早的加载 SDK, 以得到完整的 SDK 功能。

示例:

```
<head>
    <script
type="text/javascript"src="//api.h5.u.360.cn/tool/config/sdk?v=1"></script>
</head>
```

2.2 初始化设置

```
<script>
    window.mk.ready(callback)
</script>
```

游戏打开后, 秒开游戏平台会自动与游戏内的 js-sdk 建立通信; 通信建立后, 会调用传入 `ready` 方法中的回调方法; 如果通信已建立, 那么执行 `window.mk.ready(callback)`; 会立即调用传入的 `callback`”

一个 demo 范例示意如下:

接入用户信息范例,本范例仅供参考

```
<script type="text/javascript">
window.mk.ready(
    function() {
        $.get("您的验证签名的接口"+location.search, function(data){
            alert("签名验证的结果: " + data.data.sign);
            alert("欢迎您: " + data.data.nickname);
            for (var sProp in data.data.userinfo) {
                // 显示用户信息
                console.log(sProp+'-'+data.data.userinfo[sProp]);
            }
            //todo 游戏自身的各种业务 ,如绑定角色,显示设置昵称等
        });
    }
);
</script>
```

2.3 初始化后执行的方法

通信建立后,可执行的方法:

1. `window.mk.bindAccount();`

//用户绑定登录信息; 如果游戏前端知道目前登录的试玩账号,可调用该方法让玩家绑定登录信息

2. `window.mk.buy(pay_data, callback);`

//游戏调用 sdk 的支付功能; pay_data 参数 4.1 章节参数集,

3. `window.mk.showAlertBox(title, content);`

//游戏可以调用该方法显示一些提示信息; 如果不需要显示 title, 那么 title 可以为 false 或空字符串;

3. 游戏登录与获取用户信息 (必接)

此部分接口为获取从 360H5 游戏中心登录的当前用户, 包含获取用户信息。

1. H5 游戏中心 iframe 打开游戏, 游戏按照步骤 2 初始化 SDK。Sdk 主动调用登录 (sdk 自有逻辑, 此处不需要开发者自己调用)。

2. sdk 登陆成功后，会将参数拼接在打开的 url 中。
3. 游戏需要校验签名后，根据 URI 中的对应内容做参数获取，用于账号绑定。

3.1 游戏登录校验签名方法

Sdk 登录成功后，会将用户的部分信息拼接在 iframe 中，避免有人恶意修改，游戏需要先校验签名再根据获取的用户信息做账号映射。

- 签名机制与方法：

根据下列验证表，和附录 1 的签名方法，进行签名计算和校验

- 签名参数集（游戏自身参数不参与签名计算）

参数	描述	是否必须
t	当前时间戳，游戏方必须验证时间戳，暂定有效期为当前时间前后 5 分钟	是
nonce	随机数	是
plat_user_id	平台用户 ID	是
nickname	用户昵称	是
avatar	头像	是
is_tourist	是否为游客	是
sign	签名，游戏方必须按照验证规则来验证	是

3.2 获取用户信息

校验参数完成之后，根据返回的 URI 中显示的参数，需要游戏方用户信息做账号映射，具体参数如下

参数	描述	是否必须应用
plat_user_id	平台用户 ID：唯一 id 需要游戏方将此 id 与自己的游戏 id 做账号映射	网游必须
nickname	用户昵称：此部分为平台昵称，游戏方可选取是否需要	否
avatar	头像：此部分为平台昵称，游戏方可选取是否需要	否

- 重点问题备注：

1. plat_user_id 是最长可达 32 的字符串，是字符串。
2. 平台接入 360 登录，微博登录，游客登录等其他登录方式，均由游戏中心进行账号校验和账号转换，游戏仅通过 plat_user_id 作为唯一用户 id 进行关系映射即可。

4. 支付配置（必接）

此部分为带支付游戏必接接口，如果无支付游戏可省略此部分内容。H5 游戏中心目前提供微信和支付宝两种支付方式供游戏自主调用。

● 支付步骤

1. 用户点击游戏的收银台和付费界面，游戏向自身服务器生成购买订单, 返回必要参数和签名，之后页面调用 `window.mk.buy` 方法，呼起 360 支付页面。
2. 360 自有逻辑展示收银台，引导用户完成付费。
3. 360 服务器调用游戏提供的 `notify_url`，通知游戏付费成功，游戏需要返回 `ok` 字样给 360, 标识成功响应, 然后根据自身逻辑实时或者离线给用户发放对应商品。

4.1 订单发起和调用支付

应用方需要生成自己的订单号，应用订单号不能重复提交，并且一个应用订单不管是否支付成功，都只能(成功)支付一次，由于成功依赖支付服务的回调可能存在延时, 建议应用方尽量每次都重复生成订单号, 避免重复支付。游戏服务器生成订单号后，调用 h5 游戏中心的支付接口购买游戏内道具。

● pay 接口(JS 接口)

`window.mk.buy(pay_data,callback)`

参数说明：

`pay_data` : 支付信息, 将数组转换为 json 格式传递

支付信息 `pay_data` 间下面参数说明(游戏方服务端生成)

Callback, 回调函数, 其中参数如下 `callback(data)` :

字段	描述	备注
<code>data.errno</code>	0,成功, 其他,失败,一般为 360 秒开平台错误, 支付由于多次跳转,如果支付渠道发生错误或半途跳转被终止,可能无法返回本参数,时候成功请以后端通知为准(只有成功才会后端通知)	必须
<code>data.errmsg</code>	提示信息,如果成功返回 ""成功""字样	必须
<code>data.order_id</code>	应用方的订单 id	必须

● 参数集

游戏将下面的字段作为数组,json_encode 为字符串,通过参数传递

字段	描述	备注
game_key	应用名称	必须
game_name	游戏名称	必须
plat_user_id	平台颁发给 CP 的用户账号	必须
amount	金额单元为元	必须
product_name	游戏名称	必须
product_desc	商品描述描述	可选
notify_url	回调通知 cp 服务端后端回调地址	必须
order_id	cp 服务端订单 id,不允许重复使用	必须
sign	按照签名算法计算出来的签名	必须
timestamp	时间戳,超过 30 分钟的订单无法支付	必须

示例:

PHP 完成数据签名组装, 输出为 js 的 json 对象:

```
<?php
```

```

    $pay_data = array(
        'game_key'=>'您的 game_key',
        'game_name'=>'挑战 100 亿',
        'plat_user_id'=>'f0a3fb5cd19595dd98424300452629c5',
        'amount'=>'100',

        'product_name'=>'游戏金币',
        'notify_url'=>'您的回调地址', //

        'order_id' =>'你的订单生成器生成的 id',
        'timestamp' => time(),

    );
    $game_secret = 'xxxxx';
    $sign = Helper_Sign::genSignature($pay_data,$game_secret);
    $pay_data['sign'] = $sign;

```

```
?>
```

```
//js
```

前台调用:

```

var pay_data = <?php echo json_encode($pay_data);?>;
<script language="javascript">
    window.mk.buy(pay_data);//360sdk 会自行打开收银台页面,完成支付
</script>

```

● sign 签名机制与方法:

根据上述参数, 按照[文档附录 1](#) 中的签名机制用 game_secret 进行签名。

● 错误信息:

code	说明

● 重点问题备注：

1. 游戏方需要生成对应的签名，已经对应的订单信息，调用支付方法。sdk 服务端会校验对应参数和签名的合法性。
2. 道具名称 product_name 会应用于用户收银台的展示，需要游戏方根据商品信息正确传入。
3. notify_url 为游戏中心支付成功的回调地址，需要游戏方传入。

4.2 支付成功通知和签名校验（服务端接口）

应用客户端调用支付接口时，需指定支付结果的通知回调地址 notify_uri。支付完成后，360 服务器会把支付结果以 GET 方式通知到此地址(建议应用服务端接口同时支持 GET 和 POST)。应用接收验证参数后：

1. 验证签名并检查 360SDK 服务端支付结果通知接口中的 plat_user_id，（游戏方自己的 id）与下单时游戏方记录的是否一致性。
2. 如果一致，则正常发货，并向 360 服务器返回 ok
3. 如果出现不一致，则可能出现交易信息篡改，需将通知接收状态、商品/道具最终发货结果状态及未发货原因回应给 360 服务端。

支付结果通知参数集

字段	描述	备注
game_key	应用 key	必须
plat_user_id	平台用户 id	必须
order_id	应用方订单 id	必须
amount	订单金额,单位为元,只支持整数,最低 1 元	必须
plat_order_id	360 平台生成的订单 id,建议记录,便于对账	必须
sign	按照签名算法生成的签名	必须

应用接收到支付平台回调的请求，参见附录 1 的签名算法对参数进行签名，然后和平台传递的签名 sign 比较，从而校验平台请求的合法性。

● 通知参数样例：

```
array (
  'game_key' => '05695309c5edba8e179545e1851eedba',
```

```
'plat_user_id' => 'f0a3fb5cd19595dd98424300452629c5',  
'order_id' => '1489484697',  
'amount' => '100',  
'plat_order_id' => '1703146389979232530',  
'sign'=>'bc621125c293791688b9b66b504790a9';  
)
```

- 重点问题备注:

1. 应用服务端在接收到通知消息并校验后, 返回纯字符串 ok 字样给 360 服务端, 则表示成功响应, 如果不返回 ok, 360 服务端将在一天内重复通知多次, 请不要重复给用户发货。
2. 应用应做好接收到多次通知的准备, 防止多次加钱。对于重复的通知, 应用可能发现订单已经成功处理完毕, 无需继续处理, 也要返回应答。否则, 360 会认为未成功通知, 会继续发送通知。
3. 请注意, 支付成功后, 360 只会在服务器后端通知应用, 请应用自行完成页面和自身服务端的交互流程, 确定用户是否真实支付, 同时给予发送商品。

附录 1: 游戏签名方法

签名算法不区分前后端, 只要在需要签名的地方, 均采用如下的算法

- 签名方法

1. 上文提及需要签名的参数集中, 必选参数必须有值, 而且参数值必须不为空, 不为 0., 字符集为 utf-8。
2. 格式为 md5(签名字符串+game_secret), 签名字符串是采用 utf-8 编码并且是 urlencode 的原始数据, 与参数 sign 做比较。
3. 在请求参数列表中, 除去 sign 参数外, 其他需要使用到的参数皆是要签名的参数。
4. 对所有不为空的参数按照参数名字母升序排列(如 php 的 ksort 函数)

- 示例:

```
// $secret 为游戏的密钥 game_secret  
public static function genGameSign($data,$secret)  
{  
    ksort($data);  
    $base = '';  
    foreach($data as $name => $value) {
```

```
        $base .= ($name . $value);  
    }  
    return md5($base.$secret);  
}
```

- 平台提供一个简易验证签名的地址:

<http://api.h5.u.360.cn/tool/index/validateSign>