

掌盟 SoEasy 服务平台

—H5 游戏 ZMSDK 开发者帮助文档

V2.4

版本号	完成日期	修改人	备注
2.0	2017-02-05	郭博	添加支持白鹭等渠道； 接口增加 getLoginTypes、 spLogin 两个接口
2.1	2017-05-09	张宋跑	主体接口没有更改，细化 接入说明
2.2	2017-05-19	张宋跑	不影响之前版本 增加返回用户校验
2.3	2017-07-25	张宋跑	上报游戏角色接口参数补 充 SDKJS 支持 https
2.4	2017-07-31	张宋跑	Cp 支付订单加入校验
2.5	2017-12-25	程礼化	添加打包类型选项

目录

掌盟 SoEasy 服务平台—H5 游戏 ZMSDK 开发者帮助文档.....	1
目录.....	3
1.1 功能概述.....	4
1.2 前期准备.....	4
1.3 平台信息配置.....	4
1.4 SDK 接入流程.....	5
1.5 SDK 接入.....	5
1.6 SDK 功能接口说明.....	7
1.7 注意事项.....	16

1.1 功能概述

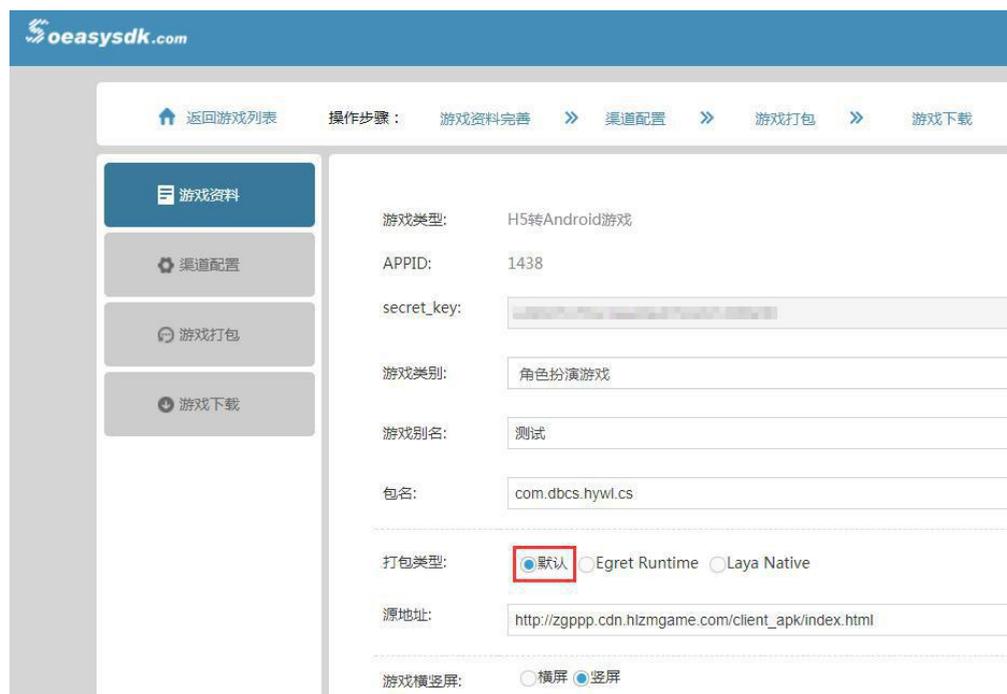
掌盟 SoEasy 服务平台的 H5 游戏 ZMSDK 旨在为 H5 游戏 CP 提供尽可能统一的渠道接入方式。“程序对接一次，支持所有渠道发布。”是我们的服务目标！

1.2 前期准备

在掌盟 SoEasy 服务平台（下文统称平台）注册账号、创建游戏配置并获取相关信息。
平台地址：<http://dev.soeasysdk.com/>

1.3 平台信息配置

在平台创建 H5 游戏，其中“打包类型”选择“默认”。（如果需要选用其它打包类型，请联系商务索取相应的对接文档）



The screenshot shows the configuration page for an H5 game on the SoEasy SDK platform. The page title is "oeasysdk.com". The breadcrumb navigation is: 返回游戏列表 > 操作步骤: 游戏资料完善 >> 渠道配置 >> 游戏打包 >> 游戏下载. The left sidebar contains: 游戏资料 (selected), 渠道配置, 游戏打包, 游戏下载. The main content area shows the following configuration fields:

- 游戏类型: H5转Android游戏
- APPID: 1438
- secret_key: [Redacted]
- 游戏类别: 角色扮演游戏
- 游戏别名: 测试
- 包名: com.dbcs.hywl.cs
- 打包类型: 默认 Egret Runtime Laya Native
- 源地址: http://zgppp.cdn.hizmgame.com/client_apk/index.html
- 游戏横竖屏: 横屏 竖屏

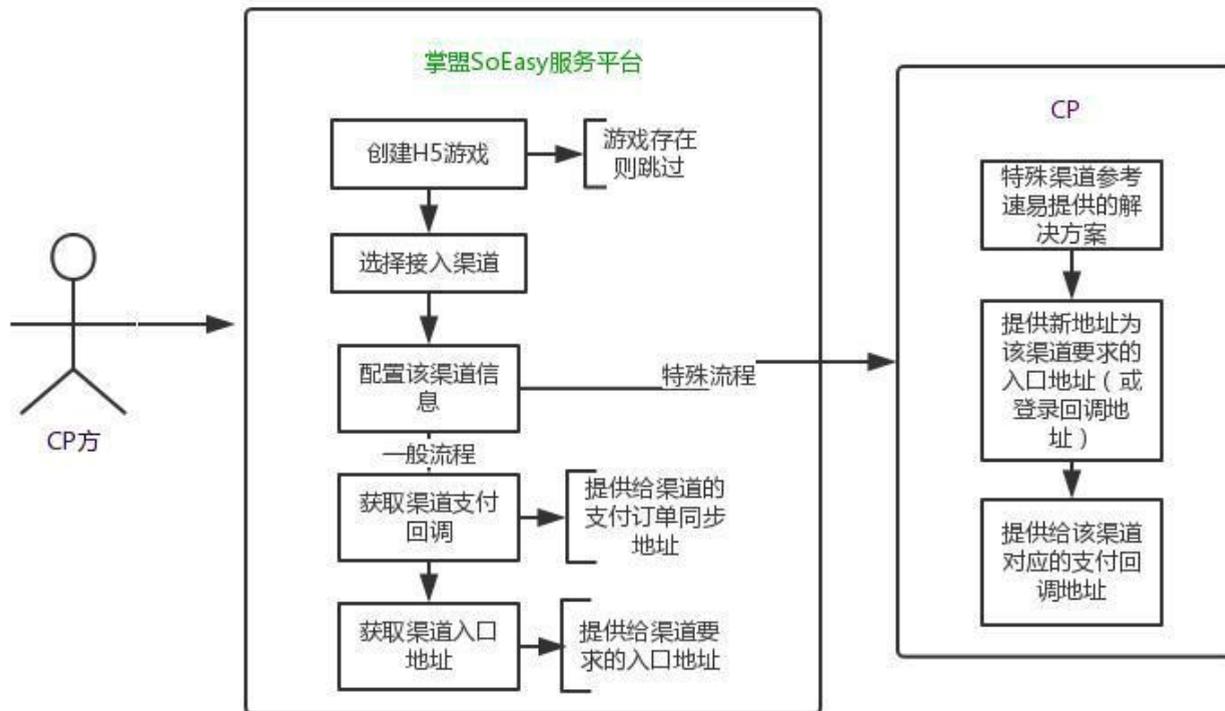
渠道配置时速易平台生成两个链接，分别是“渠道支付回调”和“渠道入口地址”，一般提供给合作渠道这两个链接，可获取配置参数。

渠道支付回调：指与合作渠道要求提供的支付成功通知地址

渠道入口地址：有些渠道也叫登录回调地址，指与合作渠道要求提供的游戏入口地址（个别渠道游戏链接不支持带参数，需要 CP 参考速易提供的解决方案，提供新链接给渠道;如 CP 自己实现游戏链接必须带参数 `zmcon={速易按渠道生成字符串}`)

注：首次接入选择"杭州掌盟"渠道,无需填写参数，方便快捷，支持到位。

1.4 SDK 接入流程



1.5 SDK 接入

请按照如下步骤接入:

```
<script type="text/javascript">
```

```
    //sdk 加载完毕回调
```

```
    window.zmInitSucc = function() {
```

```
        console.log("cp start work");
```

```
        //TODO 必须在这里或者该方法调用之后进行 sdk 调用
```

```
        ZmSdk.getInstance().init(function(data){
```

```
            //初始化成功之后调用其他 sdk 能力 如 获取用户信息、支付、角色上报、设置分//享信息、分享等...
```

```
//示例:
if(data.retcode === "0"){
    var userinfo = ZmSdk.getInstance().getUserInfo();
    .....
}else if(data.retcode === "1"){
    //初始化失败处理
}
});
}
</script>

<script type="text/javascript"
src="http://soeasysdk.com/soeasysr/zm_engine_v 2.js"></script>
或者
<script type="text/javascript"
src="https://cn.soeasysdk.com/soeasysr/zm_engi ne_v2.js"></script>
```

注意:



`zmInitSucc` 方法的声明需要放在 `zm_engine_v2.js` 文件引入之前并保证是全局访问，SDK 在 `zm_engine_v2.js` 文件引入并初始化成功之后会回调 `zmInitSucc` 方法通知游戏调用 `sdk` 的其他功能接口。

全部参数动态传到最终游戏页 JAVA 代码:

```
Iterator iterator = request.getParameterMap().entrySet().iterator();
StringBuffer param = new StringBuffer(); param.append("换成游戏最终链接");
```

```
int i = 0;
while (iterator.hasNext())
{
    i++;
    Entry entry = (Entry) iterator.next();
    if(i == 1)
        param.append("?").append(entry.getKey()).append("=");
    else
        param.append("&").append(entry.getKey()).append("=");
    if (entry.getValue() instanceof String[])
        { param.append(((String[]) entry.getValue())[0]);
        } else
        { param.append(entry.getValue())
        ;
        }
}
response.sendRedirect(param);
```

- 1.6 节中涉及的功能接口都必须在 `ZmSdk.getInstance().init(Fn)` 初始化成功之后调用。
- 1.6 节中非 **全版本** 的功能接口在调用时必须使用 `ZmSdk.getInstance().isSupportMethod(methodStr)` 方法做判断。

1.6 SDK 功能接口说明

- **检测方法是否被支持** **全版本**

```
var issupport = ZmSdk.getInstance().isSupportMethod(methodStr);
```

`methodStr`:检测的方法名

`return`: 返回值 `false` 不支持 `true` 支持

说明: 该方法用于检测渠道是否支持某个功能接

口 **例如**: 在调用 `share` 接口时

```
if(ZmSdk.getInstance().isSupportMethod("share")){ Z
    mSdk.getInstance().share(shareData,Fn);
}
```

➤ 设置 iframe 深度 需要在 init 之前调用 全版本

```
ZmSdk.getInstance().setFrameDeep(mFrameDeep);
```

mFrameDeep:游戏 frame 深度 游戏本身没有嵌套可以不调用 如果有嵌套填入嵌入层数

说明: 该方法用于设置游戏嵌入 iframe 的深度, 像腾讯玩吧渠道不允许游戏嵌入 iframe, 默认情况下不用实现这个方法

➤ 游戏初始化 全版本

```
ZmSdk.getInstance().init(Fn);
```

Fn(data){...}:成功 data -> {"retcode": "0 成功 1 失败}

说明: 该方法用于游戏初始化完成渠道用户认证等操作, 需要在进入游戏时调用。

➤ 上报游戏角色或者服务状态 全版本 (此上报服务有关统计数据 (进入游戏用户))

```
ZmSdk.getInstance().reportRoleStatus(roleInfoJSON);
```

roleInfoJSON:{"datatype": "必填 1.选择服务器 2.创建角色 3.进入游戏 4.等级提升 5.退出游戏", "serverid": "服务器 id", "servername": "服务器名称", "roleid": "角色 id", "rolename": "游戏角色昵称", "rolelevel": "角色等级", "fightvalue": "战力", (以上字段为必填字段) "moneynum": "游戏币", "partyname": "工会", "rolecreatetime": "角色创建时间", "rolelevelmtime": "角色升级时间", "gender": "角色性别, 可传 '男'、'女' ", "professionid": "职业 id", "profession": "职业名称", "vip": "vip 等级", "partyid": "所在帮派 id", "partyname": "所在帮派名称", "partyroleid": "帮派称号 id", "partyrolename": "帮派称号名称", "friendlist": [{"roleid": "关系角色 id", "intimacy": "亲密度"}, ...]} **JSON 对象**

说明: 1、该方法用于对游戏人物数据上报渠道, **必选**。SDK 会根据渠道不同进行上报。2、渠道存在不上报不给上线情况, 如果个别参数没有传空值, 建议都传值。3、极少数渠道要求通过服务端拉取游戏数据, 需要 CP 自行实现接口。4、reportRoleStatus 接口并非速易平台要求上报, 是渠道需要上报。

➤ 上报游戏数据 全版本 (此上报服务有关统计数据 (点击分享次数和点击商城次数))

```
ZmSdk.getInstance().reportRoleDetail(roleInfoJSON);
```

roleInfoJSON:{"datatype": "必填 1.点击分享 2.点击商城", "serverid": "服务器 id", "servername": "服务器名称", "roleid": "角色 id", "rolename": "游戏角色昵称", "rolelevel": "角色等级", "fightvalue": "战力", (以上字段为必填字段)...]} **JSON 对象**

说明: 1、该方法用于对游戏数据进行上报, **可选**。SDK 会根据渠道不同进行统计。2、如果个别参数没有传空值, 建议都传值。3、reportRoleDetail 接口并非渠道要求上报, 是速易统计游戏数据需要。



支付 全版本

ZmSdk.getInstance().pay(payinfojson, onPayCallBackFn);

payinfojson: {"check": "md5(fee+feeid+secret_key)", "feeid": "必填 cp 方自定义的计费 id", "fee": "必填 金额 分", "feename": "必填 商品名称", "extradata": "透传参数 支付回调通知时会带给 cp", "serverid": "游戏分区号", "rolename": "角色 名称", "roleid": "角色 ID", "servername": "分区服名称" }

JSON 对象（里面的具体对象请采用 **String** 传值）**onPayCallBackFn(data):**支付回调

- data.retcode 0 成功 1 失败 2 取消 3 跳转到了支付界面或渠道不支持
- data.msg 支付描述

说明: 该方法用于道具付费。必选接口，md5 中“+”不参与加密。

check=md5(fee+feeid+secret_key) 加密必须在服务端完成。

关于 serverid、rolename、roleid、servername, rolelevel 是渠道要求，并非速易平台要求上报。（code 返回-5 表示 check 值加密不正确）

游戏方生成订单时根据 feeid 校验 fee，保证 feeid 和 fee 一一对应。

Check 值为必传，请保证 secret_key 私密性，以免泄露造成恶意刷单。

"secret_key": 获取步骤: 登录速易平台 → H5 游戏 → 创建游戏(已创建省略此步) → 配置 → 游戏资料 → 获取 secret_key 如图



"feeid" : 指速易计费模板中的 feield

获取步骤: 登录速易平台 → H5 游戏 → 创建游戏(已创建省略此步) → 配置 → 渠道配置 → 计费模板管理 → 获取"feield"如图



计费点(充值档): 填写信息对应 CP 游戏中的所有充值

➤ 获取渠道登录认证后的用户信息 全版本

```
var userinfo = ZmSdk.getInstance().getUserInfo();
```

return:成功返回

```
{"common":{"sdkindx":"","sdkname":"","channel":""},"userdata":{"uid":"","name":"","sex":"","area":"","nick":"","t":"","sign":""}} 失败 返回{}
```

说明: 该方法用于获取登录认证后的渠道用户信息。必选接口。

userdata 中 uid、t、sign 必传, 其他字段依渠道情况而定。

sdkindx: 用来区分渠道渠道对照表:

<http://www.soeasysdk.com/channel/list.html>

用户返回校验算法

sign=md5(secret_key=1&t=2&uid=3) 参数首字母升序排列 md5:32 小写

➤ 设置分享信息

ZmSdk.getInstance().setShareInfo(shareData, sharedCBFn);

shareData: {"title": "标题", "content": "简介", "imgurl": "图片地址 可选", "ext": "透传参数", }

sharedCBFn (ret){...}: 回调结果 ret -> {"success": "ok" 成功}

➤ 分享

ZmSdk.getInstance().share(shareData, sharedCBFn);

shareData: {"title": "标题", "content": "简介", "imgurl": "图片地址 可选", "ext": "透传参数", }

sharedCBFn (ret){...}: 回调结果 ret -> {"success": "ok" 成功}

说明: 透传参数只有个别渠道支持，会附带在分享出去的地址上。**setShareInfo** 和 **share** 两个方法必接。

1. **setShareInfo(data, callback)** 初始化分享数据(进入游戏就可以初始化)，使用 **isSupportMethod** 判断方法是否存在，存在，显示游戏里分享按钮

2. **share(data, callback)** 渠道的分享提示，使用 **isSupportMethod** 判断方法是否存在，存在调用此方法显示渠道的分享提示，不存在，显示自己的分享提示

1 和 2 方法，存在任意方法表示该渠道支持分享

1 和 2 方法，分享成功会回调返回 "success": "ok", 判断给用户发放奖励

➤ 显示关注二维码

ZmSdk.getInstance().showQRCode(showData, showCBFn);

说明: 该方法用于渠道内弹框显示关注二维码，部分渠道支持，非必选。判断获取关注状态

1 获取 user info 里面的 isSubscribe 字段, true 代表已经关注, false 表示没有关注, 判断给用户发放一次关注奖励。(非实时, 需要重新登录)

2 isSubscribe(data,callback) 返回关注信息 isSubscribe="true"已经关注
isSubscribe="false"没有关注(实时获取, 只有个别渠道支持, 需要用 isSupportMethod 判断)



发送到桌面

```
ZmSdk.getInstance().sendToDesktop(sendData, Fn);  
Fn(ret){...}:绑定执行 ret -> {"success":"ok" 成功}
```

说明: 该方法用于游戏内发送快捷方式到手机桌面, 部分渠道支持, 非必选。(判断奖励部分渠道是直接返回保存桌面接过, 部分渠道会在桌面进去的地址附带参数自行进行判断, 具体渠道请咨询技术人员)



切换账号

```
ZmSdk.getInstance().spChangeAccount();
```

说明: 该方法用于退出登录、切换账号等操作



调用渠道弹框

```
ZmSdk.getInstance().openDialog(url);  
url:弹出框需要加载的 url
```

说明: 该方法弹框加载指定 url



绑定手机

```
ZmSdk.getInstance().bindPhone(Fn);  
Fn(ret){...}:绑定执行 ret -> {"retcode":"0 成功 1 失败 2 取消","phone":"成功时存在", "msg":"没成功存在"}
```

说明: 该方法用于渠道绑定手机, 部分渠道支持, 非必选。



cp 需要实现方法 开放游戏音效控制接口

```
window.pauseAudio();//提供这个方法实现暂停游戏内所有音效
```

```
window.resumeAudio();//提供这个方法实现恢复游戏内所有音效
```



关闭游戏音效

```
ZmSdk.getInstance().setPauseAudio(Fn);
```

说明：该方法用于游戏内关闭游戏音效



开启游戏音效

```
ZmSdk.getInstance().setResumeAudio(Fn);
```

说明：该方法用于游戏内开启游戏音



分享逻辑

```
if(setshareinfo){  
    执行 setshareinfo//这个方法最好在进入游戏的时候调用  
}  
if(setshareinfo || share){  
    显示分享按钮  
}
```

点击分享按钮

```
if(share){  
    调用我们渠道这边的 share，显示渠道分享提示  
}else{  
    自己处理，显示游戏自己的分享提示  
}
```

1.7 注意事项(非常重要)

- 节中 **非全版本** 的功能接口在调用时必须使用 `ZmSdk.getInstance().isSupportMethod(methodStr)` 方法做判断, 对接时建议实现全版本和非全版本 (加判断方法) 功能可以兼容所有渠道。
- 节中涉及的功能接口都必须在 `ZmSdk.getInstance().init(Fn)` 初始化成功之后调用。
- `setShareInfo(data,callback)`初始化分享数据, 使用 `isSupportMethod` 判断方法是否存在, 存在, 显示游戏里分享按钮
- `share(data,callback)`渠道的分享提示, 使用 `isSupportMethod` 判断方法是否存在, 存在调用此方法, 不存在, 显示自己的分享提示
- `setShareInfo` 和 `share` 方法, 分享成功会回调返回"success": "ok",判断给用户发放奖励
- `showQRCode(data,callback)`//使用 `isSupportMethod` 判断方法是否存在, 存在显示游戏内的关注按钮, 点击调用此方法;
- 关注的判断是根据初始化的时候, 获取 `userinfo` 里面的 `isSubscribe` 字段, `true` 代表已经关注, 隐藏按钮, `false` 表示没有关注, 显示按钮, 同事判断给用户发放一次关注奖励
- `isSubscribe(data.callback)` 返回关注信息 `isSubscribe="true"` 已经关注 `isSubscribe="false"`没有关注



sendToDesktop 保存桌面 成功会回调返回"success": "ok",

