

小游戏sdk接入文档

更新时间2024-12-04

文档共分为两部分，第一部分为前端接入文档，第二部分为后端接入文档。

前端部分主要解决了以下几个问题:

- 1.用户对应盛也平台的数据回传问题,为后续的投放和买量数据分析做铺垫.
- 2.封装好了抖音小游戏的虚拟支付,以及获取用户openid,减少了游戏方开发的复杂性.

后端部分主要解决了以下几个问题:

- 1.建立签名验证体系,验证双方消息的真实性,避免恶意伪造请求.
- 2.使用虚拟支付时,盛也向游戏方的回传支付数据.
- 3.获取小游戏的access_token.由于双方系统中都有刷新access_token的需要,所以协调出一个access_token的获取方式。本文档内access_token由盛也刷新,并提供接口供游戏方获取最新的access_token和对应的过期时间.

目录

- 接入须知:
- 一、前端部分
 - 1、引入sdk文件
 - 2、初始化SDK
 - 3、用户登录
 - 4、进入游戏
 - 5、用户发起支付
 - 6、【聊天】检测屏蔽词
 - 7、流量主广告行为上报
 - 8、获取侧边栏icon开启状态信息
 - 9、拉起侧边栏引导
 - 10、拉起订阅按钮
- 二、后端部分
 - 1、签名说明
 - 2、获取accessToken参数
 - 3、盛也支付回调接口

- 4、给用户发送订阅消息

接入须知：

△注意：sdk接口返回字段类型为非统一类型【根据不同需求返回int、string等】，建议CP方根据自身引擎及语言对sdk返回值进行类型转换，避免出现类型对不上导致报错的情况

1、接入sdk时需盛也提供对接参数，每个游戏包都有一套独立参数，不可重复使用！

2、必接项：

(1) 内购类&混合变现类

[一、前端部分]：1、2、3、4、5

[二、后端部分]：3

(2) 纯广告类

[一、前端部分]：1、2、3

3、接入注册登陆接口syLogin需注意：

需使用syLogin回调返回的openid字段作为用户唯一身份标识使用，不可使用其他字段作为用户唯一身份标识，避免后续玩家注册登陆时出现异常情况。

4、接入后端接口3：盛也支付回调接口时需注意：

请研发服务端在收到用户支付回调请求时校验支付金额。如出现金额不一致的情况，请勿发货！！

此类问题主要针对存在用户对游戏抓包，修改订单金额进行异常操作。

一、前端部分

前期准备：将sysdk-conf.js， sysdk-wxapp.js 放入项目文件目录。配置好sysdk-conf.js内的参数

```
module.exports.APP_ID = 'xxxxxxxxxxxxxxxxxxxxxx'; // 小游戏的appid
module.exports.GAME_KEY = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'; // 盛也提供，用于识别游戏
```

对接流程如下：

1、引入sdk文件

使用相对路径，在小游戏中引入sysdk-wxaapp.js

```
import Sygame from './sysdk-wxapp';
```

2、初始化SDK

功能：上报用户的入口参数。请务必在最开始就调用初始化init函数，参考代码如下：

```
// 需要把tt.getLaunchOptionsSync得到入口参数放到初始化文件中
let data = tt.getLaunchOptionsSync();
Sygame.init(data)
```

3、用户登录

功能：用户注册、登陆。返回用户唯一身份标识字段：openid；参考代码如下：

```
// 请使用openid作为用户唯一凭证
Sygame.syLogin().then((res) => {
  console.log(res);
  // res.code      状态码，
  // res.message    状态说明
  // res.openid     用户openid（用户唯一凭证）
  // res.is_new_user 判断当前用户是否为新用户。bool类型，true表示当前用户是新用户，false表示当前用户为老用户
  // res.session_key 用户session_key
})
```

4、进入游戏

用户在游戏内有创建角色，进入游戏，用户升级三种操作时，上报数据。注意：本功能中的区服ID（server_id）如果产生合服操作，合服之后应传角色母服，而不是当前服

```

let data = {
  role_id: 4, // string 游戏方处的用户id, 必填, 长度不得
超过32位
  role_name: "角色名称", // string 角色名称
  role_level: 80, // int 角色等级
  server_id: "1", // string 区服id, 必填 (注意: 如果产生合服
操作, 合服之后应传角色母服, 而不是当前服)
  server_name: "3333", // string 区服名称
  role_vip: 8, // int 角色vip等级
  role_power: 89828828, // int 战力、武力之类角色的核心数值
  report_type: "entergame", // string 上报类型 "entergame" , "cr
eaterole", "roleupgrade"
};
Sygame.syReportRoleInfo(data).then((res) => {
  console.log(res);
  // res.status 状态码
  // res.info 状态信息
  // res.can_pay 是否可以让用户看到支付入口, 1为是, 0为否。调取syRe
portRoleInfo的时候需要传入role_level
})

```

参数名称	说明	类型	长度	是否必传
role_id	角色ID	string	32	是
role_name	角色名称	string	100	是
role_level	角色等级	int	11	是
server_id	区服id, (注意: 如果产生合服操作, 合服之后应传角色母服, 而不是当前服)	string	32	是
server_name	区服名称	string	100	是
role_vip	角色vip等级	int	11	是
role_power	战力、武力之类角色的核心数值	int	11	是
report_type	上报类型 “entergame” 进入游戏, “createrole” 创建角色, “roleupgrade” 用户升级	string	100	是

注:
必填字段:

```
role_name, server_id, report_type , role_id, server_id, server_name, role_level, role_vip
```

5、用户发起支付

对接本功能前请先对接功能4(创角升级上报功能)。本功能需要游戏方后端提供接口。支付流程如下：

(1) 用户发起支付，游戏方前端通过自己的后端接口，生成一笔订单信息，此笔订单内包含用户的身份以及购买道具等信息。订单id要求长度为10-32，是订单的唯一标识，即使清档后也不能重复。建议使用appid+无序字符串生成，以避免和其他游戏的订单重复，若长度超出，可使用md5加密。

(2) 游戏方后端将此笔订单对应的订单id传给前端。

(3) 前端拿到订单id后，把订单id作为下段代码内payData的order_id字段，调用syPay函数。

(4) 用户完成支付，盛也收到到账通知。

(5) 盛也方通过[后端接口3](#)异步的请求游戏方的回调地址，回传步骤3的order_id字段，通知用户充值成功。

(6) 游戏方在收到充值成功的信息后，尽快给用户发货。

参考代码如下：

```
let payData = {
  product_name: "商品名", // 商品名称 可选
  product_id: "商品ID", // 商品ID 可选
  order_id: "gorder1233", // string 订单号 必填，游戏方生成的订单id，长度为10-32的字符串，是订单的唯一标识，即使清档后也不能重复。建议使用appid+无序字符串生成，以避免和其他游戏的订单重复，若长度超出，可使用md5加密。
  product_price: 0.01, // 必填，商品价格，以元为单位
  extends_param1: "额外参数1", // 选填，服务器将此参数原封不动回传至CP服务器
  extends_param2: "额外参数2", // 选填，服务器将此参数原封不动回传至CP服务器
  role_id: "角色ID", // 必填，角色ID，与创角升级上报的role_id保持一致
  role_name: "角色名称", // 选填，角色名称
  role_level: "角色等级", // 选填，角色等级
  server_id: "服务器ID", // 必填，服务器ID（注意：如果产生合服操作，合服之后应传角色母服，而不是当前服）
  server_name: "服务器名称", // 选填，服务器名称
  role_vip: "VIP等级" // 选填，VIP等级
}
```

注意：role_id、server_id需与上方角色信息上报接口中参数信息保持一致

//可参考下列代码段

```
Sygame.syPay(payData).then((res) => {
```

```

    console.log(res);
    console.log(res.info);        // 提示信息
    console.log(res.payId);       // 订单ID(不是orderid)
    console.log(res.status);      // 接口状态信息
}

```

参数名称	说明	类型	长度	是否必传
product_name	商品名	string	100	否
product_id	商品ID	string	100	否
order_id	订单号 必填，游戏方生成的订单id，长度为10-32的字符串，是订单的唯一标识，即使清档后也不能重复。建议使用appid+无序字符串生成，以避免和其他游戏的订单重复，若长度超出，可使用md5加密。	string	32	是
product_price	商品价格,以元为单位	int	10	是
extends_param1	额外参数1、服务器将此参数原封不动回传至CP服务器	string	100	否
extends_param2	额外参数2、服务器将此参数原封不动回传至CP服务器	string	100	否
role_id	角色ID，与创角升级上报接口的role_id保持一致	string	32	是
role_name	角色名称	string	100	否
role_level	角色等级	int	11	否
server_id	服务器ID（注意：如果产生合服操作，合服之后应传角色母服，而不是当前服）	string	32	是
server_name	服务器名称	string	100	否
role_vip	VIP等级	int	11	否

注：

1) 购买游戏币的时候，buyQuantity 不可任意填写。需满足 $\text{buyQuantity} \times \text{游戏币单价} = \text{限定的价格等级}$ 。如：游戏币单价为 0.1 元，一次购买最少数量是 10。有效价格等级请点击[抖音文档](#)。

6、【聊天】检测屏蔽词

功能说明：

- 1、该接口判断用户当前聊天发送内容是否是是需要屏蔽；
- 2、若需要屏蔽，则该内容仅允许用户本身可以看到，其他用户不允许看到；
- 3、若不需要屏蔽，则内容正常发送，所有人可见。

【注意无论内容是否需要屏蔽，都可成功发送，只是可见人不同】

参考代码如下：

```
//可参考下列代码段
let word = '测试'; //待检测内容
Sygame.syCheckShieldWords(word).then((res) => {
    console.log('屏蔽词检测结果', res)
    res.status:1001 // 检测成功
    res.isShieldWords:true // 是否需要屏蔽, true是; false否【查看功能说明2, 3】
})
```

7、流量主广告行为上报

参考代码如下：

```
//每次用户看完广告，游戏内发放奖励完毕之后，调用该接口进行数据上报
let testParams = {
    "position": 1, // 必填, int, 广告位, 1为激励, 2为banner
    "task": "幸运大转盘" // 必填, string, 展示场景名称, 若无展示场景请传空字符串
};
Sygame.syUploadCasualAdInfo(testParams);
```

8、获取侧边栏icon开启状态信息

参考代码如下：

```
//获取侧边栏icon开启状态信息
```

```
Sygame.syGetSidebarIconOpenStatus().then((res) => {  
    console.log(res);  
    //iconShowStatus    //icon展示状态:1展示、0不展示  
    //allowGetReward    //允许获得奖励: 1允许、0不允许  
})
```

9、拉起侧边栏引导

参考代码如下:

```
//注意, 调用该方法时需调用组件tt.onTouchStart监听开始触摸事件, 并且在拉起  
侧边栏之后调用组件tt.offTouchStart取消监听开始触摸事件;  
Sygame.sySidebarGuide();
```

10、拉起订阅按钮

注意: 该接口仅为拉起订阅按钮, 发送订阅消息给用户还需接入[后端接口4](#)

说明: 主要为拉起订阅按钮, 顺带统计用户点击允许订阅按钮、取消订阅按钮

参考代码如下:

```
let templateIds = ['xxxxx']; //模版ID 必填, 最多一次请求三个模版ID, 例  
如: ['xxxxxx01', 'xxxxxx02', 'xxxxxx03']  
//可参考下列代码段  
Sygame.syGetSubscribe(templateIds).then((res) => {  
    console.log(res);  
    //res.status //状态码; 1001, 成功; 4001, 失败; 1  
})  
  
//注意, 调用订阅方法时需调用组件tt.onTouchEnd监听触摸结束事件, 并且在拉起订  
阅消息之后调用组件tt.offTouchEnd取消监听触摸结束事件;
```

二、后端部分

1、签名说明

签名用途

为了验证数据的合法性，避免恶意的伪造请求，双方后端在产生数据交互时，通过签名的方式验证url请求的真实性。

签名生成规则

- (1) 在参数列表中，除去sign字段，无特殊说明都是需参与签名生成的字段。为空的字段请设置为0或空字符串，**不要设置为null**（会导致签名错误）。
- (2) 将参与生成签名的参数的key按照字典升序排序（使用ksort函数），然后把key=value用&连接组成字符串，得到待签名字符串。
- (3) 在步骤2得到待签名的字符串后直接拼接appsecret 得到签名字符串。盛也提供，建议不要暴露到前端。
- (4) 对签名字符串进行md5加密后转成小写得到签名。

通用参数说明

参数名	类型	说明
channel	string	我方提供的小游戏key，可暴露在前端
appsecret	string	我方提供，用于支付相关操作，建议不要暴露到前端
openId	string	用户openid，唯一标识

2、获取accessToken参数

功能：小游戏需要使用accessToken时，通过此方法获取accessToken

接口地址：

```
// 注：GET请求拼接此链接是&开头，如下
// https://docater1.cn/index.php?g=Home&m=MiniGame&a=getAccessToken&channel=test&appid=testappid&sign=test
GET: https://docater1.cn/index.php?g=Home&m=MiniGame&a=getAccessToken
```

请求参数：

参数名	类型	说明	长度	是否必传
appid	string	小游戏appid	20	是
channel	string	我方提供的小游戏key	32	是
sign	string	签名（具体参考 签名说明 ）	32	是

返回参数：

参数名	类型	说明
status	string	状态码
info	string	状态说明
accesstoken	string	获取到的accesstoken
expiretime	string	获取到accesstoken的过期时间

3、盛也支付回调接口

功能：盛也通知游戏方用户充值成功。盛也收到用户支付成功请求后，会根据游戏方给出的异步通知地址，通过POST 请求的方式将支付结果异步通知到游戏方系统。注意：支付回调接口必须返回字符串'success'（小写，且不包含引号）以表示正确收到通知。

为避免出现用户刷bug的情况，请研发服务端在收到用户支付回调请求时校验支付金额。如出现金额不一致的情况，请勿发货。

异步通知参数：

参数名	类型	说明
orderid	string	订单号
openid	string	用户openid, 唯一表示
price	int	价格（单位：元），为避免出现用户刷bug的情况，请研发服务端在收到用户支付回调请求时校验支付金额。如出现金额不一致的情况，请勿发货。
item_id	int	类别id

other	string	额外参数1 用"&"符号拼接下单时的额外参数1(extends_param1)和参数2(extends_param2)
sign	string	签名 （具体参考 签名说明 ）

4、给用户发送订阅消息

功能: 在用户点击订阅按钮后，确定订阅该消息，则调用接口向用户发送订阅消息。

请求地址:

- POST: <https://docater1.cn/index.php?g=Wap&m=MiniGameTt&a=sendTplMsgToUser>

- 参数说明

参数名称	说明	是否必传	类型	长度
openId	用户的openId	是	string	32
appId	游戏appId	是	string	18
channel	游戏GAME_KEY	是	string	32
templateId	模版ID	是	string	34
data	模板内容，格式形如 {"版本号": "v1.0", "版本描述": "新版本发布了"}	是	Object	

- 请求示例

```
//请求body
{
  "openId": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "appId": "xxxxxxxxxxxxxxxxxxxx",
  "channel": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "templateId": "MSGxxxxxxxxxxxxxxxxxxxx",
  "data": {
    "版本号": "v1.0",
    "版本描述": "新版本发布了"
  }
}
```

```
}
```

- 返回参数

参数名	类型	说明
status	string	状态码
msg	string	提示信息

```
//正确的请求回调
{
    "status": 1001,
    "info": "error"
}

//参数错误
{
    "status": 4001,
    "info": "error",
    "msg": "error param"
}

//发送订阅失败
{
    "status": 4002,
    "info": "error",
    "msg": {
        "err_no": 1001,
        "err_tips": "该用户未订阅"
    }
}
```