

目 录

公共SDK接入

接入必读

责任使命

流程图

客户端接入文档

服务端接入文档

公共使用指南

获取配置文件

常见问题处理

打包工具常见问题及处理

接入常见问题及处理

登录充值常见故障处理

打包工具、签名工具

打包工具使用文档

签名工具使用文档

打包工具默认签名cbzr.keystore信息

公共SDK海外版

谷歌/FaceBook后台操作流程

华为海外后台操作流程

海外安卓常见问题

华为国内后台操作流程

谷歌签名配置

TA后台关联AF后台数据操作

公共后台操作文档

SDK后台操作

使用后台配置商品列表功能

微信小游戏操作流程

游戏实名认证微服务

实名系统主体密钥更新（三个月一次）

v2

公共实名安卓SDK接入文档

新海豹SDK操作流程

文明用语检查服务

投放包体测试事项

 安卓APK母包测试事项

 安卓9133SDK测试事项

 微信小游戏测试事项

 安卓星云投放SDK测试事项

对内安卓文档

 1)SDK添加渠道对接文档(内部)

 2)SDK资源抽取提交到公共后台

 3)gradle快捷抽取资源

 4)打包工具配置相关

对内H5文档

对内苹果文档

乐游/飞跃内部文档

 接口接入说明文档

 前后端联调文档

 选择大区接口说明

 上报事件到公共后台

公共SDK接入

欢迎您接入公共SDK

公司内部游戏接入公共SDK需要到的材料：

1. 接入文档及Demo
2. 接入必读[\(此文\)](#)

上面材料公司内部游戏请找平台人员申请

发送参数申请邮件

[标题] 项目组 - 游戏名 - 申请接入公共SDK参数
[项目负责人] xxx
[运营对接人员] xxx qq/手机
[对接目的] 游戏自研联运发行/其它
[是否有公共后台帐号] 是/否

发给运营中心负责人 赵毅春 zhaoyichun@gzyouai.com 抄送 赵楚凡zhaochufan@gzyouai.com 赵艳zhaoyan@gzyouai.com

如果首次接入，收到邮件回复参数后需要运营人员到16楼技术中心学习后台和打包工具使用，以便更快成功接入，顺利发行。

发行平台找对应运营人员

- 上海游爱 企业QQ上找 黄伟肯
- 神手 联系对应运营人员
- 光卡 联系对应运营人员
- 公共SDK海外发行平台 企业QQ上找 李海波

遇问题请在左面常见问题中查找，无法解决按模板提问，提高问题解决效率

接入必读

欢迎您接入公共SDK

公司内部游戏接入公共SDK需要到的材料：

1. gameSimpleName:xxx
2. gameAppKey:xxxx
3. 接入文档及Demo[点击下载](#)
4. 接入必读([此文](#))

上面材料公司内部游戏请找平台人员申请

发送参数申请邮件

[标题]项目组-游戏名-申请接入公共SDK参数

[项目负责人] xxx

[运营对接人员] xxx qq/手机

[对接目的] 游戏自研联运发行/其它

[是否有公共后台帐号] 是/否

发给技术中心负责人 赵毅春zhaoyichun@gzyouai.com 抄送 赵楚凡zhaochufan@gzyouai.com

如果首次接入，收到邮件回复参数后需要运营人员到 运营中心 学习后台和打包工具使用，以便更快成功接入，顺利发行。

发行平台找对应运营人员

- 上海游爱 飞书 赵艳
- 神手 联系对应运营人员
- 光卡 联系对应运营人员
- 公共SDK海外发行平台乐游 飞书 李海波

遇问题请在左面常见问题中查找，无法解决按模板提问，提高问题解决效率

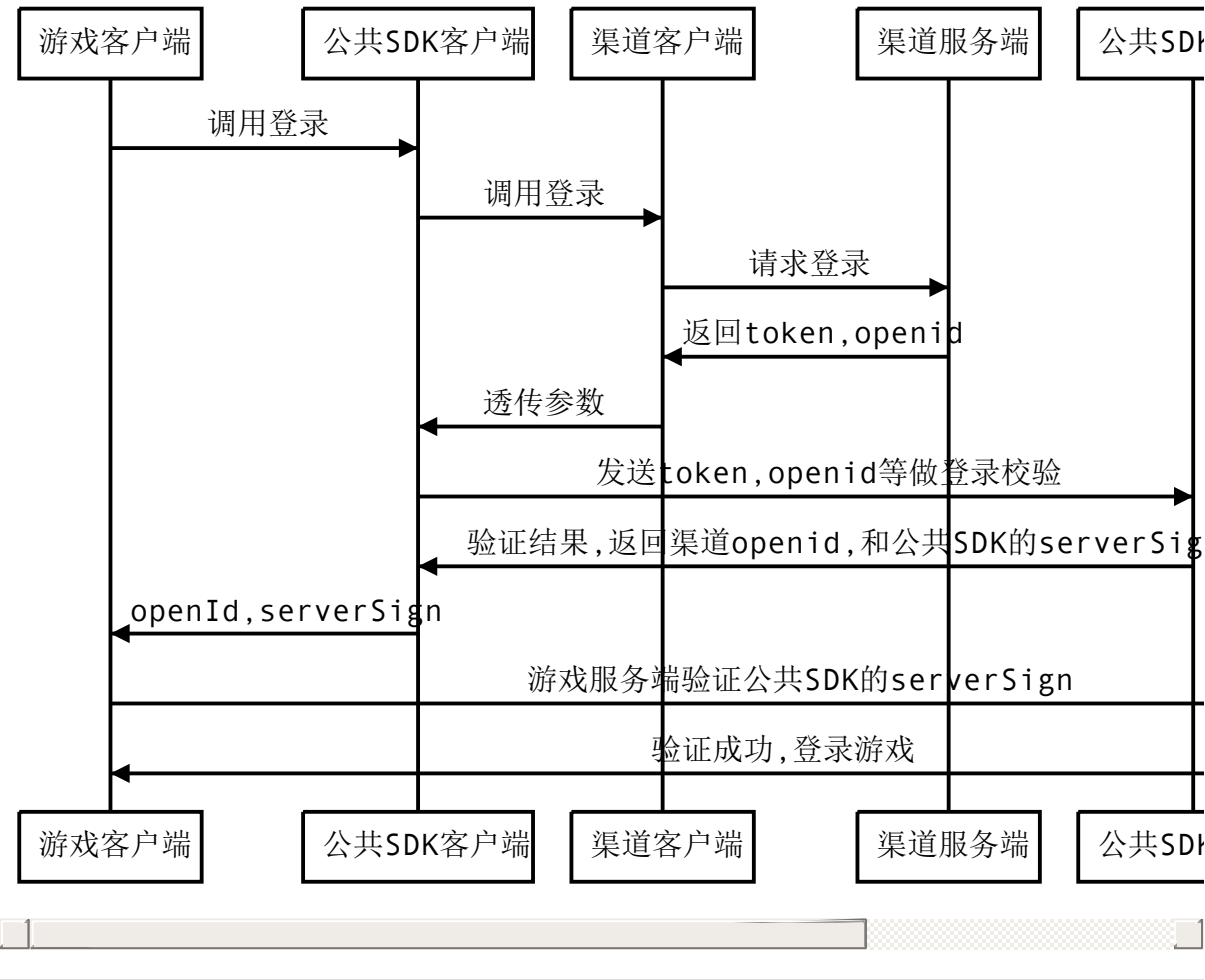
责任使命

责任心

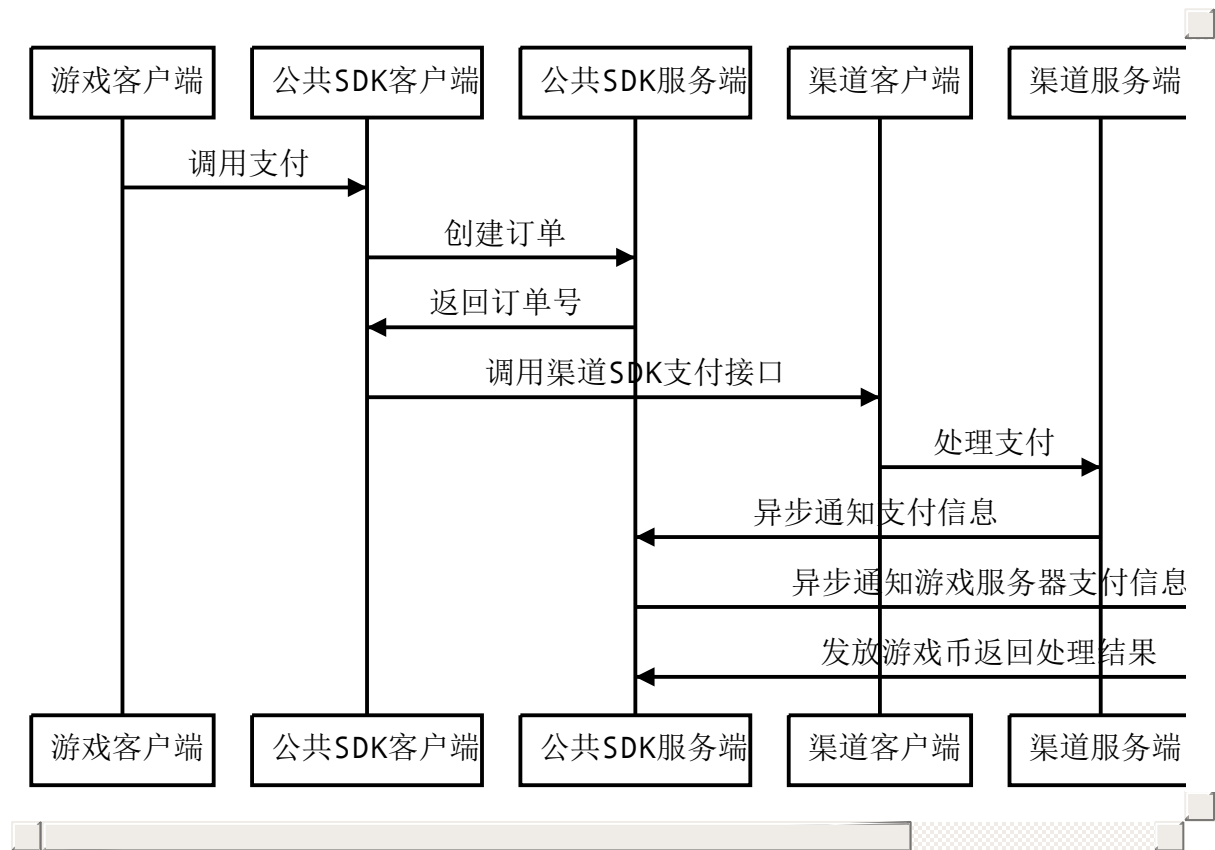
- 严格执行流程，适应快速迭代的节奏，提高业务素养
- 完成目标计划，可以自由做其它事情；
- 服务态度：
 - 一小时响应机制 —— 对于需求、问题，必须做到先回应；
 - 沟通必须到直接人，比如 与对方技术直接沟通；
 - 措辞必须平和 沟通，不能推诿；
 - 素养：
- 除了专业技术技能，也需要加强文档编辑积累，更多辅助技能；

流程图

公共SDK登录流程图



公共SDK充值流程图



客户端接入文档

点我获取 最新**SDK**接入文档和**demo**

服务端接入文档

公共SDK服务端接入文档

1. 登录验证

客户端发送请求参数给服务端, 服务端对参数进行 MD5 加密, 如果加密结果和serverSign 一致, 则验证通过, 否则不通过

参数	类型	描述
userType	int	用户类型 ID(渠道类型,兼容以 前帐号体系)
openId	string	平台渠道方返回用户ID
timestamp	string	时间戳,公共SDK发送到客户端的参数timestamp, 服务端可以判断 timestamp 是否 在有效的登录时间内
serverSign	string	验证密钥,公共 SDK 发送到 客户端的参数 serverSign,加密串:gameAppkey=%s&userType=%s&openId=%s×tamp=%s注意: (改为游戏本身的gameAppkey值)

验证方式:

serverSign=MD5(gameAppkey=%s&userType=%s&openId=%s×tamp=%s) 如:

md5("gameAppkey=xxxx&userType=1&openId=123×tamp=1111111111")

gameAppkey : 在公共SDK后台的 Md5加密的KEY

注意!CP必须要以userType+openId保证帐号唯一性

2. 充值验证(通知游戏方发货)

公共SDK服务器接收到渠道服务器充值回调, 对回调进行 MD5 验证, 如果验证通过, 则采用Post 方式马上回调给游戏充值服务器, 否则不会回调充值服务器, 然后再返回 信息给渠道服务器。回调参数如下:

参数	类型	描述
serverId	int	服务器 ID
playerId	int	角色 ID

orderId	string	渠道方订单 ID
payAmount	double	充值金额(如有商品 ID,此值为商品的金额, 单位元)
currency	string	充值货币(CNY:人民币)
goodsId	string	商品ID,参数对一些有商品列表 的充值有效
goodsName	string	商品名称
remotelp	string	客户端充值 IP
custom	string	透传参数 ,长度 512
serverSign	string	发给充值服务器验证的注意:(改为游戏本身的 gameAppkey 值)
uld	string	平台渠道帐号 ID
gameSimpleName	string	平台SDK后台游戏代号名
sdkSimpleName	string	平台SDK代号
queryId	string	平台SDK订单号
postTime	string	订单创建时间戳
sign	string	升级Md5加密的值

验证方式:

旧:serverSign=MD5(serverId=%s&playerId=%s&orderId=%s&gameAppKey=%s)

升

级:sign=Md5('serverId=%s&playerId=%s&orderId=%s&payAmount=%s&goodsId=%s&sdkSimpleName=%s&gameAppKey=%s')

%s:为相应的值

gameAppkey : 在公共SDK后台的Md5加密的KEY

验证通过后,返回公共 SDK 服务器数据,数据格式:{"code":0,"message":"无订 单"}。code:0 表示失败,1 表示成功,message:描述原因。

注意!CP必须要以orderId+sdkSimpleName保证订单唯一性,建议使用sign验证

3. 网页下单(渠道使用)-不常用

请求地址 : 运营获取

请求类型 : post

参数:

参数	类型	描述
----	----	----

data	string	标准json字符串,内容参见以下说明
------	--------	--------------------

data字段说明:

参数	类型	描述
serverId	int	服务器 ID
playerId	int	角色 ID
postAmount	double	充值金额(如有商品 ID,此值为商品的金额, 单位元)
productId	string	商品ID,参数对一些有商品列表 的充值有效
productName	string	商品名称
custom	string	透传参数 ,长度 512
userId	string	平台渠道帐号 ID

返回内容:

参数	类型	描述
message	string	请求报错时候的错误信息
code	int	请求结果: 1 成功, 其他为失败
data	json	json字典, 内容参见以下说明

data字段说明:

参数	类型	描述
postTime	int	下单timestamp
serverId	int	服务器 ID(重要)
playerId	int	角色 ID(重要)
postAmount	double	充值金额(如有商品 ID,此值为商品的金额, 单位元)(重要)
productId	string	商品ID,参数对一些有商品列表 的充值有效(重要)

custom	string	透传参数,长度 512(重要)
uld	string	平台渠道帐号 ID(重要)
gameSimpleName	string	平台SDK后台游戏代号名
sdkSimpleName	string	平台SDK代号
queryId	string	平台订单号(重要)
products	json	商品表,苹果用

请求结果参考::

```
{'message': '', 'code': 1, 'data': {'postTime': 1566040451, 'danjiConfig': '', 'uId': u'2587803', 'playerId': u'4351590982', 'sdkSimpleName': u'apple_shenshou', 'custom': u'1#0#', 'currency': 'CNY', 'other': '', 'products': {'u'com.gzss.rsrzrcj.328': {'u'amount': u'328', u'name': u'3280\u91d1\u5e01'}, u'com.gzss.rsrzrcj.648': {'u'amount': u'648', u'name': u'6480\u91d1\u5e01'}, u'com.gzss.rsrzrcj.68': {'u'amount': u'68', u'name': u'680\u91d1\u5e01'}, u'com.gzss.rsrzrcj.198': {'u'amount': u'198', u'name': u'1980\u91d1\u5e01'}, u'com.gzss.rsrzrcj.6': {'u'amount': u'6', u'name': u'60\u91d1\u5e01'}, u'com.gzss.rsrzrcj.25': {'u'amount': u'25', u'name': u'\u6708\u5361'}, u'com.gzss.rsrzrcj.30': {'u'amount': u'30', u'name': u'300\u91d1\u5e01'}}, 'remoteIp': '211.97.130.143', 'postAmount': 6.0, 'queryId': '2019081719141186011839B6', 'serverId': u'4150', 'gameSimpleName': u'1031', 'productId': u'1001'}}
```

4. 充值服务器IP地址

```
180.184.161.139
180.184.171.182
180.184.161.184
180.184.161.63
```

公共使用指南

1、熟悉公共后台使用(运营)

- 文档地址
<http://docs.tech.gzyouai.com/docs/publicsdk/publicsdk-1b9t4amlo5u48>

2、在公共后台出对应配置文件(运营)

- 首先使用公共给的账号登录，然后创建对应的游戏版本(具体使用查看1步骤文档)
- 添加对应渠道(参数由公共平台提供)
 - android/ios平台需要在对应游戏版本中添加公共sdk，填写对应参数，完成后下载对应配置文件(poolsdk.xml/pool_setting)给对接技术,技术在对接中使用
 - h5平台在对应游戏版本中添加公共h5sdk,填写对应参数，完成后复制生成的H5生成地址给对接技术(此地址用于后面技术对应完聚合h5 sdk后进行测试联调)

注：详细操作步骤文档：

<http://docs.tech.gzyouai.com/docs/publicsdk/publicsdk-1bvf6mcpaealh>

3 对接公共sdk(技术)

- 下载公共sdk对接资源文档
 - Android下载地址：
http://gitlab.9133.com:10080/public_service/public_sdk_android/tree/master
 - h5下载地址：
http://gitlab.9133.com:10080/public_service/public-js-doc/tree/master
- 找运营拿对应平台的配置文件(运营在步骤2中获取)
 - Android平台拿配置文件poolsdk.xml用于对接中替换(具体可查看对接文档)
 - h5平台拿生成的测试地址，用于接入测试联调
- 熟悉公共sdk对接流程图，下面为流程图地址：
<http://docs.tech.gzyouai.com/docs/publicsdk/publicsdk-1b5h0te539nbs>
- 开始对接聚合sdk(有问题接入群沟通)
- 联调测试各个接口功能(有问题接入群沟通)
- 各个接口调通后出个母包给公共测试，公共会测试调用接口情况进行测试
- 公共测试通过后，安卓生成母包给运营，h5直接告诉运营出对应渠道短链

4 打渠道包(运营)

- Android平台
 - 拿到技术对接了公共sdk的母包

- 公共后台对应游戏版本中添加要打的渠道，填写对应参数(参数由对应渠道提供)，具体可查看步骤1的使用文档
- 熟悉打包工具使用
文档地址：
<http://docs.tech.gzyouai.com/docs/publicsdk/publicsdk-1b7mc1ma229jo>
- 使用打包工具出对应渠道包
- 打出渠道进行测试
- 测试完成给包渠道
- h5平台
 - 公共后台对应游戏版本中添加要打的渠道，填写对应参数(参数由对应渠道提供)，具体可查看步骤1的使用文档
 - 获取填写后的H5生成地址给渠道

获取配置文件

注:生成的Android配置给技术接入使用

注:h5地址给技术接入联调使用

Android/IOS平台:

- 确认熟悉了后台使用(<http://docs.tech.gzyouai.com/docs/publicsdk/publicsdk-1b9t4amlo5u48>)
- 在后台创建对应的游戏版本(不明白的可查看后台使用文档)
- 在对应游戏版本添加公共sdk渠道参数，然后下载配置，具体如下：

步骤1

步骤2

步骤3

步骤4

步骤5 下载该配置文件发送给游戏接入人员

h5平台

- 后台的对应游戏版本中添加下游戏的h5地址

步骤1:

步骤2:

- 在对应版本中添加公共**h5sdk**渠道，填写参数保存，复制生成的**h5**地址给技术
步骤1:

步骤2:

步骤3:

常见问题处理

打包工具常见问题及处理

打包工具常见故障处理

提问模板

- 需要 ☐ 运营 按模板提供以下信息
 - 一般回调地址已经包含项目所有信息

紧急程度：[一般|紧急]
参数自检：[是|否]
提供log/sdk版本/母包
问题描述述：

打包报错请把打包日志**log**发给相关技术

文件的路径不要花里胡哨

闪屏 母包 图标等的文件夹路径名字 ☐ 不要使用 特殊符号 空格 中文标点符号！（打包工具做了初步校验）

如果打包后发现在**assets**中视频音频资源无法播放

打包时把 ☐ 不压缩 选项勾上

接入常见问题及处理

提问模板

- 需要 运营 按模板提供以下信息
 - 一般回调地址已经包含项目所有信息

紧急程度：[一般|紧急]
参数自检：[是|否]
提供游戏渠道
问题描述述：

常见问题处理

1.sign错误

是否更改**poolsdk.xml** 为正式参数，没有请找对应运营人员申请

1.魅族

调试登录时，提示“游戏不存在”或“游戏签名不正确”

SDK 服务端会对参数作必要的校验，校验不通过则有此提示

- 1、联系运营检查该游戏是否打开了联调状态
- 2、联系运营检查是否创建了该游戏
- 3、上架后的游戏需要从游戏中心下载 **APK** 包进行安装

2.搜狗

参数**QQ_APPID**要自行到**QQ**开放平台申请手**Q ID**

签名必须跟应用宝**YSDK**一致

3.华为

需要母包删除 **libs** 中的 **YouaiSDK.jar** 再打包

4.三星

参数 **waresid** 为商品ID，需去三星后台设置为应用传入价格

1. 出现“The signature of this application is not registered the server”这样的错误
 - a) 保证用对了三星的appid,appsecret，这两个参数是必须向三星申请的，申请渠道：Jisun Seo(jisun.seo@samsung.com),Chris Lee (kt85.lee@samsung.com)
 - b) 三星的appid必须绑定cp自己的包名，在申请上面两个参数的时候，要把自己的要发布的包名和签名发给你三星，让他们绑定，再绑定之前或是你调试的时候是不能用的
 - c)必须用发布的包，不能用直接在eclipse或android studio中run的包
2. 进不去收银台，出现“您已取消授权”这样的提示
 - a) 确保您用的是三星手机，因为这版支付sdk是定制在三星手机的，必须用三星手机调试或测试及使用。
 - b)三星账号过期的时候要重新登录
 - c)在账号登录界面你点击了取消或返回按钮也会有这样的提示PS: 从V2.0.0版本以后提示已经更新为“请确认已使用三星手机，并已登录三星账号”
3. 进不去收银台，出现“账户管理器中已存在三星账户”这样的提示
 - a)检查你的清单文件中是否已经注册了账户权限
4. 手机支持Samsung Pay但是在收银台使用Samsung Pay支付时提示“手机不支持Samsung Pay”进入SamsungPay App设置密码添加一张可用的银行卡即可。

5.联想

waresid请到联想后台申请商品编码（使用开放价格模式）

6.UC

提交角色信息接口

PoolSdkHelper.submitRoleData接口中的角色创建时间和角色等级变化时间需传正确的值，否则不过审

poolRoleInfo.setRoleChangeTime();//角色等级变化时间

poolRoleInfo.setRoleCTime();//角色创建时间

7.酷派

platPublicKey平台公钥默认值

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDLx50R+CpH6VD/Efav0TpH0ydfAQVjvm/h
/SGG/RbM/03ZDfk8JrgezyFwqX06pjKevHZz66Umoq1/3Z9oqhB1JezsM03q2woKbReGSKAP
QSkUJg2FY6DdZ1v1gMuCRtF6W0HV7n2TbUvRhZugMpx8PJEV+r6Qf1L41Gt8w6FN8wIDAQAB
```

如渠道有提供以提供为准，没有填默认值

8.应用宝YSDK

请保留好在应用宝后台申请时使用的签名文件，打包时需使用一致的

9.OPPO

publicKey应用公钥

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCMreYIkPwVovKR8rLHWlFVw7YDfm9uQ0JK
L89Smt6ypXGVdrAKKl0wNYc3/jecAoPi2ylChfa2iRu5gunJyNmpWZz1CNRIau55fxGW0XEu
553Iipr0Zcaw50uYGlF60ga8QT6qToP0/dpiL/ZbmNU09kUhosIjEu22uFgR+5cYyQIDAQAB
```

如渠道有提供以提供为准，没有填默认值

10.HTC

joloPrivateKey私钥

- “HTC秘钥生成工具与文档”

<https://pan.baidu.com/s/1dE7FSm1>

强烈建议找渠道技术帮忙生成

11.卓易

1.注册账号时输入的账号密码文字颜色和输入框底色相同

把的theme改成android:theme="@android:style/Theme.Black.NoTitleBar"

或者把Activity的theme改成android:theme="@android:style/Theme.Black.NoTitleBar"

2.集成到游戏后，发现卓易悬浮图标点击没反应

如果是Unity3D的游戏的话，由于Unity3D和Android之间的事件冲突，导致Android事件得不到响应。

解决办法是，需要将游戏的主Activity（游戏入口）默认继承的NativeActivity或UnityPlayerNativeActivity改为继承Activity，把UnityPlayerNativeActivity里面的方法（如下图所示）拷贝到主Activity。

现在最新Unity3D版本游戏入口也是默认继承Activity。



12.打包

参数提前确定准备好，尤其是icon，闪图那些

填写某渠道参数时，请牢记到对应渠道后台填写充值回调地址

13.多玩YY

支付时显示支付未成功，**appid**不存在

因为签名不正确，在游戏提交后联运管理台后，会重新打包签名。（如不了解请与渠道联系）

javax.net.ssl.SSLHandshakeException: Chain validation failed 报错时

检查下本地手机时间是否正常，是否有代理或者安装其他证书

14.360

支付时显示**Mobile pay signature invalid**

后台填写的QHOPENSdk_PRIVATEKEY参数错误导致，注意此处private_key，不是app_secret，其生成算法为app_secret + “#” + app_key的加密MD5值，结果为32位全小写（<http://pmd5.com/>在线计算MD5）



如图，替换自己的app_secret和app_key即可

登录充值常见故障处理

SDK充值常见故障处理

获取问题反馈地址

每一个渠道包，都带有一个独立的问题反馈地址，**运营**在填写完**sdk**参数之后，自动生成，运营可以单独保存地址到本地。当需要技术协助的时候，右键复制链接地址把地址填写到提问信息里面，取代以前的充值回调地址。

提问模板

- 需要 **运营** 按模板提供以下信息

紧急程度：[一般|紧急]
参数自检：[是|否]
上线：[是|否]
问题类型：[充值|登录|接入]
服务端参数：充值问题时提供
问题反馈地址：
问题描述述：

问题反馈例子

- 如

紧急程度：紧急

参数自检： 是

上线： 否

问题类型： 充值

服务端参数：

问题反馈地址： 获取位置请看上图

问题描述述： 充值不到账，客户端问题，请把客户端日志放在这里

sdk登录失败，在后台没有登录日志

1. 检查服务端参数是否填写
2. 点击更新sdk配置
3. 按照模板反馈给平台技术

游戏登录失败，游戏登录校验不通过

1. 检查字段大小写是否正确
2. 检查签名串是否正确，字段顺序是否一致
3. 检查是否编码导致签名不一样
4. 检查用户类型是否一致
5. 检查加密的密钥是否使用正确和是否提供正确（后台是否配置正确）
6. 算法和文档是否一致 [服务端对接文档](#)

充值成功，后台没有订单(调试阶段，游戏没上线)

1. 检查服务端参数是否填写
2. 检查渠道回调地址是否填写
3. 按照模板反馈给平台技术

批量玩家充值失败，后台没有订单(上线阶段)

1. 和渠道确认参数有没有修改
2. 反馈信息标记已上线
3. 按照模板反馈给平台技术

后台有订单号，但游戏充值不到账

1. 公共平台作为渠道和游戏中间的连接体，只负责接收渠道的充值成功信息，并将接收到充值信息通知游戏。
2. 充值金额，订单号以渠道实际通知为准。
3. 在这过程中可能会出现，渠道通知金额和游戏创建订单时输入的金额不符、一笔游戏订单多次重复充值等情况。
4. 部分游戏可能可以正常处理，但校验严格的游戏可能会不到账。这种情况需要 [运营](#) 通知 [游戏](#)，游戏手动补单到账。
5. 后台不提供修改订单，补发功能

打包工具、签名工具

打包工具使用文档

首先 [下载打包工具](#) [点击下载](#)

1. 登陆需要打包对应的平台

2. 选择需要打包的游戏项目和**SDK**版本

3. 注意各种素材文件的路径不要有空格和特殊符号，打包报错请发送打包日志给相关技术

4. 小工具里面的工具 有需要的 **请自行愉快使用**

5. 查看页面可以查看APK**信息（打包后可以查看到底你填的参数有没有错误）**

6. 批量打包要用 **打出渠道包当母包**，只支持修改渠道号，渠道标识**1**，渠道标识**2**，表格模板在帮助页面有，**批量打包的签名在打包配置页面**配置，如无配置则使用全局默认签名

7. 其他相关帮助说明请查看帮助页面

8. 打包额外配置说明 **（大部分情况默认值就好，有需要技术会与你告知）**

- ~~1. 整合（5.0版本开始打包工具添加res资源整合方案，母包跟渠道资源冲突时，优先使用母包资源）~~
- ~~2. 不优化（打包工具默认使用zipalign对回编译后apk进行优化，打钩后则不优化）~~
- ~~3. 不改Unkonwn文件夹（打包工具默认回编译后再复制unknown文件进apk）~~
4. 压缩（打包工具默认将assets文件夹不压缩处理，避免打包后游戏CG、音频无法播放）
如果觉得打包后包体太大，请勾选，但是又有例外的文件不压缩，勾选压缩选项，上面yml配置选择例外文件，格式为下图

注意 target>=30的母包 需要加上

- arsc

5. 忽略assets（当母包assets文件数量过多、文件名过长，造成回编译时超出CMD命令长度而失败，请打钩）
6. 显示Log（打包后默认公共层SDK的Log输出关闭）
7. 母包public（默认资源整合时，会删除母包public.xml以提高合并效率，如果母包资源ID写死的请保留）
- ~~8. 渠道public（默认资源整合时，会删除渠道SDK的public.xml以提高合并效率，但是如果渠道是用固定包名获取资源ID的方式请保留）~~
- ~~9. 旧母包（因历史原因遗留下聚合的打包方式，使用于同时2个平台接入，后面不在维护，所以保留聚合打包方式入口）~~
10. 兼容模式（6.3版本后默认aapt2打包方式，勾选使用aapt1进行打包）
11. V2签名（上谷歌应用商店才勾选，国内不用勾选，打包工具默认用V1 jarsigner签名方式，勾选后使用apksigner V2签名方式 ！注意！V2方式无法使用投放计划服务端注释apk包修改渠道号的功能！）

9. 补丁包的使用方法

如果需要在打包时添加或替换assets文件可使用该功能

10. pad包的使用方法(国内不用看)（Play Asset Delivery）

1. 因为谷歌后续使用**aab+pad**包的上传应用模式 **aab包 包含pad资源** 的格式如下

其中

base 为母包资源

META-INF 为签名信息

BUNDLE-METADATA 为编译工具的信息

BundleConfig.pb 为aab包的编译信息

其他的文件夹 都是 **pad** 资源包

2. 生成打包工具可以使用的**pad**资源包

2.1 请确保游戏提供的**asset**资源包格式如下

压缩包的根目录文件夹为 **assets**，资源都在文件夹内部
压缩包的根目录文件夹为 **assets**，资源都在文件夹内部
压缩包的根目录文件夹为 **assets**，资源都在文件夹内部

2.2 打开打包工具工具页面 生成PAD资源包功能页

PAD资源文件名 填写游戏定义的PAD资源文件名，对应游戏pad工程 build.gradle 文件 **packName**

分发模式 选择游戏PAD资源的分发模式，对应游戏pad工程 build.gradle 文件 **deliveryType**

运营如不清楚上面两项请问游戏开发，下图为对应游戏pad工程参考

附 分发模式 解释

install-time 资源包在用户安装应用时分发。这些资源包以拆分 APK（APK 集的一部分）的形式提供。它们也称为“预先”资源包；您可以在应用启动时立即使用这些资源包。这些资源包会增加 Google Play 商店上列出的应用大小。用户无法修改或删除这些资源包。

fast-follow 资源包会在用户安装应用后立即自动下载；用户无需打开应用即可开始 **fast-follow** 下载。下载过程中，用户仍然可以进入应用。这些资源包会增加 Google Play 商店上列

出的应用大小。

on-demand 资源包会在应用运行时下载。

2.3 如果游戏提供的母包是包含资源可以直接运行的母包（非小的**apk**跟**pad**压缩包）

- 先用以下小工具把母包拆分成 不含资源的 **apk** 和 **pad**资源压缩包

- 再按照下面第三点 进行打包

3. 打包工具目前只支持 **apk母包+pad资源包** 打包生成 **aab(内置pad资源包)** 的方式

1. 将上一步中生成的**PAD**资源包保存（可支持不同的包名打包）
2. 与打包**APK**包的步骤一样，只需要把 **aab** 选项勾上，**pad**资源选择（支持多个文件选择）正常打包即可

签名工具使用文档

签名工具的使用方法

签名工具——[点我下载](#)

1. APK签名包含三种签名方式：

- **java签名**：使用JAVA的签名方式对apk进行签名。
- **androidV1签名**：使用Google Android自带的V1签名方式进行签名。
- **androidV2签名**：使用Google Android自带的V2签名方式进行签名。

2. 特殊情况说明：

v2签名是Android新发布的签名方式，单独签名会导致低版本手机运行失败，如果想使用V2的新属性，可以使用组合签名方式进行签名(v1、v2同时勾选)（只支持8.0以上机型）

3. 如果不清楚需求请跟渠道沟通

打包工具的默认签名文件

别名密码和签名文件密码都为 **mango123**

打包工具默认签名cbzr.keystore信息

最新打包工具默认签名不自动配置了

需要手动配置之前的默认签名的话 选择工具目录下文件夹 **keystore/cbzr.keystore** 文件

KeyAlias = cbzr.keystore

KeyStoreKey = mango123

KeyAliasKey = mango123

签名其他详细信息

密钥库类型: **jks**

密钥库提供方: **SUN**

您的密钥库包含 **1** 个条目

别名: **cbzr.keystore**

创建日期: **2015-3-31**

条目类型: **PrivateKeyEntry**

证书链长度: 1

证书[1]:

所有者: CN=www.gzyouai.com, OU=www.gzyouai.com, O=mango, L=China, ST=GD, C=86

发布者: CN=www.gzyouai.com, OU=www.gzyouai.com, O=mango, L=China, ST=GD, C=86

序列号: 551a0e4a

有效期为 Tue Mar 31 11:02:34 CST 2015 至 Thu Oct 05 11:02:34 CST 2124

证书指纹:

MD5: 4E:98:D7:27:5D:47:95:9B:EE:A1:1B:FE:20:D8:E7:B2

SHA1: 11:89:59:8F:CD:5B:6A:6F:D2:EF:41:BE:17:FB:70:51:AB:BD:5F:DB

SHA256:

B1:B1:60:36:2D:57:D9:67:06:73:5E:40:BD:74:F2:1A:DC:01:AF:DD:0D:89:94:3A:7D:7E:88:ED:95:BA:

A9:3E

签名算法名称: SHA1withRSA

主体公共密钥算法: 1024 位 RSA 密钥

版本: 3

公共SDK海外版

谷歌/FaceBook后台操作流程

1. Google登录

- 打开谷歌后台地址
<https://console.developers.google.com/projectselector2/apis/credentials?hl=zh-cn&supportedpurview=project>
- 在凭据中创建项目
创建项目填写对应的应用信息获取客户端ID（填写到公共后台对应的server_client_id）
根据下图步骤操作：

签名证书密钥找出对应渠道包的人拿 签名证书密钥找出对应渠道包的人拿

2. FaceBook登录

- 登录FaceBook后台地址创建应用

<https://www.facebook.com/login/?next=https%3A%2F%2Fdevelopers.facebook.com%2F>

根据下图步骤操作

将下图红框内的应用编号填写到公共后台乐游渠道的
facebook_app_id上 将下图红框内的应用编号填写到公共后台乐
游渠道的**facebook_app_id**上

- 添加登录功能(填写对应信息)
请根据下图步骤操作

填写对应包名和

根据要求填写密钥（可以找技术获取）

3. Google支付

- 打开谷歌后台地址
<https://play.google.com/apps/publish>
- 创建应用
请根据下图操作

- 获取公共密钥

将公共密钥填写到对应公共后台的**publicKey**上 将公共密钥填写到对应公共后台的**publicKey**上

- 填写应用信息（红框栏目上的信息必填）

- 应用版本栏目

需要完整填写商品详情，内容分级，定价和发布范围才能发不应用版本中的Alpha版本
测试阶段先创建一个Alpha版本（上传对应包和填写对应信息）

- 创建商品（在应用版本栏目的**Alpha**发布审核通过后进行操作）

- 添加测试人员（在发布Alpha版本成功后操作）

华为海外后台操作流程

获取相关参数信息

登录华为 **AppGallery Connect** 后台

<https://developer.huawei.com/consumer/cn/service/josp/agc/index.html>

点击我的应用进入

选择自己应用后进入

App ID 和 **App secret**在此处获取

redirect_uri在此处设置，需要跟公共后台服务端配置一致，填公共的服务器域名（不清

楚可以询问相关技术)

例如公共平台填写的游戏登录地址为

<http://login.public.sdk.gzyouai.com/logincheck/check>

PubKey 在此处获取

配置内购商品

- 1、添加商品，也可以导入，配置的商品ID最后跟在公共平台对应填写的要一致
- 2、把商品状态设置为激活状态才生效

海外安卓常见问题

google play:

- 1、alpha版发布成功后,点击跳转没有找到对应应用
A:查看是否选择了对应国家发布
- 2、安装页面提示您的所有设备与此应用不兼容
A:查看最低兼容版本是否超过您的设备
- 3、上传包时提示签名不对时
A:检查是否与第一次上传包的签名一致，要改签名最好重新创建一个游戏
- 4、提示您所在的地区无法下载时
A:将设置登录账号的国家设置成可下载
- 5.安装包与内容提供者冲突
A:核对Provider是否是包名下的路径
- 6.我们一方发生错误，请再试一次
A:发布版本后才能购买
- 7、您无法购买所要求的商品
A:需要谷歌上的下载包才能购买
- 8:没收到登录回调：
A:看游戏是否接入了onActivityResult接口

谷歌登录:

<https://console.developers.google.com/projectselector2/apis/credentials?hl=zh-cn&supportedpurview=project>

需要在凭证后台创建两个类型应用

- 1、web类型应用获取客户端ID(对应server_client_id)
- 2、Android类型应用填写正确的软件包名称和SHA-1 签名证书指纹

ApiException:10（签名问题）

- 1、当使用google play签名计划时上传到Google play时分发包会使用谷歌签名，Google登录凭证后台需要填写google play的签名SHA-1值

Facebook登录问题:

- 1、首先要确保登录的账号没有异常（账号审核或者封号等）
- 2、当登录出现散列密钥问题时，请核对下Facebook后台填写的散列密钥和当前安装包是否一致

上架**App**设备兼容问题:

- 1、查看支持的最低**Android**版本，Google play要求的26是指target
- 2、so库兼容的cpu类型
- 3、查看**App**所需要的功能该设置是否有
- 4、设置了需要硬件的功能(NFC等)
- 5、不能含有支付宝

<https://www.quicksdk.com/doc-782.html?cid=35>

华为国内后台操作流程

获取相关参数信息

登录华为 **AppGallery Connect** 后台

<https://developer.huawei.com/consumer/cn/service/josp/agc/index.html>

充值公钥的获取

APPID的获取

游戏服务 私钥的获取

CPID的获取

SHA256证书指纹 的设置

如果使用公共**SDK**打包工具的默认签名的话则设置下面，其他的自己获取自己签名文件的值

```
B1:B1:60:36:2D:57:D9:67:06:73:5E:40:BD:74:F2:1A:DC:01:AF:DD:0D:89:94:3A:7D:7E:88:ED:95:BA:A9:3E
```

agconnect-services.json文件的下载（发送给技术，会生成一份加密后的文件给回运营替换打包）

替换文件方式 **huawei_5.0.3.302\ForRes\values** 下同名文件

初始化、登录、充值等其他报错参考官方文档

<https://developer.huawei.com/consumer/cn/doc/development/quickApp-Guides/quickgame-faq-0000001112830270>

谷歌签名配置

谷歌签名配置

1. 创建应用的时候选择从**Java**密钥库导出并上传密钥

2. 打开打包工具小工具使用导出加密密钥

打包工具默认签名文件在工具目录下
/keystore/cbzs.keystore

```
KeyStore : cbzs.keystore  
KeyAlias : cbzs.keystore  
KeyStoreKey : mango123  
KeyAliasKey : mango123
```

3. 在谷歌后台上传生成的加密密钥**zip**包

TA后台关联AF后台数据操作

1、TA后台的操作

1. 数据管理
2. 数据集成
3. 第三方数据集成
4. AF配置-打开集成开关
5. 复制数据源地址（如果未配置数据上报地址点击跳转后公网地址填 <https://www.yassdt.com>）

2、AF后台的操作

1. API数据接口
2. Push API
3. 添加回传
4. 终端URL填写上一步复制的
5. 选项都勾上-字段和事件类型都勾上
6. 保存后可以点击发送测试-TA后台会显示有数据（显示异常是因为测试数据不对，需要打包测试）

公共后台操作文档

SDK后台操作

1、获取帐号后登陆首页：(光
卡：<http://bg.public.sdk.guangkatf.com:33080/> 游
爱：<http://public.sdk.gzyouai.com:33080/index>)

2、点击增加游戏项目

3、添加完刷新后可以看到创建的游戏

4、在创建的游戏上点击查看项目版本进入下图点击增加游戏版本

5、添加完后刷新点击查看接入**SDK**

*点击增加接入 进入下面渠道参数填写操作 *。

6、渠道**sdk**添加模板说明

- 商品列表添加说明

7、修改游戏充值回调地址

- 进入sdk页面点击更新配置

填入sdk要求客户端、服务端参数，增加完后刷新才可以看到，添加完参数点击右上更新sdk配置。
添加操作完成。

使用后台配置商品列表功能

因为部分渠道（例如苹果、谷歌、华为海外等）支付需要传入在渠道后台配置的商品**ID**来调起支付（不支持直接传入商品价格方式调起支付）避免游戏需要出多个母包或者配置来调整商品的**ID**

使用方法

游戏在支付接口中，请在商品信息中传入正确的 金额 和 商品名

公共层会根据这2个来获取到后台配置的唯一商品**ID**

微信小游戏操作流程

微信小游戏接入操作流程

1.去微信小游戏后台开通虚拟支付功能

2.前往米大师配置相关参数

3.比例选**1:10**的

4.获取沙箱参数还有正式参数 分别在公共后台跟**9133**后台配置

5.如果游戏调起支付报错请参考微信文档（一般是档位不匹配）

<https://developers.weixin.qq.com/minigame/dev/api/midas-payment/wx.requestMidasPayment.html>

6.微信项目需要添加的合法域名列表

```
https://static.public.sdk.gzyouai.com  
https://sdk.i9133.com  
https://public.sdk.gzyouai.com  
https://statistic.public.sdk.gzyouai.com
```


游戏实名认证微服务

游戏实名认证服务

主要内容:

- 游戏请求接口，Redis内存响应大部分请求，不存在数据再请求mysql.写入时先写入mysql,成功后再写入Redis.
- 读写即时更新在Redis,每次接收到请求，根据lastUpdateTime更新重置日在线累计和月充值累计，Redis再根据情况同步到Mysql.
- 实名验证接口使用腾讯云第三方接口。

游戏接入

游戏异步调用**Get**查询，没有返回实名信息，在适当位置显示实名认证图标，用**Post**方式添加玩家实名认证。

1) 请求实名状态

```
https://cid.gzyouai.com/api/v1/default/cid
```

调用方式：**HTTP get**

接口描述：

请求参数：

GET参数：

字段名称	字段说明	类型	必填	备注
oid	渠道返回唯一账号id(接入公共的建议是"userType_渠道openid")	string	Y	-

请求返回结果：

```
//请求返回信息 ymd:生日 online: 在线时长（分钟 没有设置过默认不传） amount:累计充值(元 没有设置过默认不传)
{"code":200,"data":{"ymd":"20200101",online:"60",amount:"150"},"msg":"ok"}
```

返回结果：

字段名称	字段说明	类型	必填	备注
code	状态	string	Y	200成功,10002，账号不存在
data	内容	json	Y	成功为实名后年月日,不存在为空json对象
msg	提示信息	string	Y	

例子：

```
https://cid.gzyouai.com/api/v1/default/cid?oid=userType_4243289
```

2) 实名验证接口

<https://cid.gzyouai.com/api/v1/default/cid>

调用方式: **HTTP post**

接口描述:

请求参数:

GET参数:

字段名称	字段说明	类型	必填	备注
oid	渠道返回唯一账号id	string	Y	-
num	用户绑定身份证	string	Y	-
name	用户姓名	string	Y	-

请求返回结果:

```
//  
{ "code": 200, "data": "", "msg": "ok" }  
or  
{ "code": 10001, "msg": "已存在该实名认证", "data": "" }
```

返回结果:

字段名称	字段说明	类型	必填	备注
code	状态	string	Y	200成功,10001, 已存在该实名认证
data	内容	json	Y	为空json对象
msg	提示信息	string	Y	

例子:

```
curl -X POST https://cid.gzyouai.com/api/v1/default/cid -d 'oid=userType  
_4243289&num=330502880702447&name=多多'
```

3) 更新信息接口 (更新在线时长和充值金额)

<https://cid.gzyouai.com/api/v1/default/update>

调用方式: **HTTP post**

接口描述:

请求参数:

GET参数:

字段名称	字段说明	类型	必填	备注
oid	渠道返回唯一账号id	string	Y	-
type	online/amount	string	Y	-
value	值	string	Y	-

请求返回结果:

```
//请求返回信息 ymd:生日 online: 在线时长(分钟) amount:累计充值(元)
{"code":200,"data":{"ymd":"20200101",online:"60",amount:"150"},"msg":"ok"}
or
{"code":10001,"data":"","msg":"已存在该实名认证"}
```

返回结果:

字段名称	字段说明	类型	必填	备注
code	状态	string	Y	200成功,10001, 已存在该实名认证
data	内容	json	Y	为空json对象
msg	提示信息	string	Y	

例子:

```
curl -X POST https://cid.gzyouai.com/api/v1/default/update -d 'oid=userT
ype_4243289&type=online&value=10'
curl -X POST https://cid.gzyouai.com/api/v1/default/update -d 'oid=userT
ype_4243289&type=amount&value=10'

curl -X POST http://127.0.0.1:8000/api/v1/default/update -d 'oid=userTyp
```

```
e_4243289&type=amount&value=10'
```

其它接口待添加

总结

```
//请求实名接口
curl https://cid.gzyouai.com/api/v1/default/cid?oid=userType_4243289
//验证实名接口
curl -X POST https://cid.gzyouai.com/api/v1/default/cid -d 'oid=userType_4243289&num=330502880702447&name=多多'
//更新月充值和日时长
curl -X POST https://cid.gzyouai.com/api/v1/default/update -d 'oid=userType_4243289&type=online&value=10'
curl -X POST https://cid.gzyouai.com/api/v1/default/update -d 'oid=userType_4243289&type=amount&value=10'
```

接入发送申请邮件

上面地址为测试地址方便接入，对应游戏地址请找平台人员申请

```
[标题]项目组-游戏名-申请接入实名验证
[项目负责人] xxx
[运营对接人员] xxx qq/手机
[对接目的] 游戏自研联运发行/其它
[日均在线用户数] 400w
[对接公共SDK] 是/否
```

实名系统主体密钥更新（三个月一次）

因政府实名后台密钥每隔半年需要更新一次，所以麻烦游戏主体负责人或相关运营到期前发新密钥给运营中心。谢谢！

发送邮件到技术中心申请对接参数，发给运营中心负责人 赵毅春 zhaoyichun@gzyouai.com 抄送 赵楚凡 zhaochufan@gzyouai.com 莫洪章 mohongzhang@gzyouai.com

[标题] 项目组 - 运营主体 - 密钥更新

[项目负责人] xxx

[运营对接人员] xxx qq/手机

[内容] 旧密钥 和 新密钥

例：

主体公司名称：xxx

旧secretkey:xxxxxx(正式参数)

新secretkey:xxxxxx(正式参数)

v2

游戏实名认证服务

主要内容:

- 游戏请求接口，Redis内存响应大部分请求，不存在数据再请求mysql.写入时先写入mysql,成功后再写入Redis.
- 读写即时更新在Redis,每次接收到请求，根据lastUpdateTime更新重置日在线累计和月充值累计，Redis再根据情况同步到Mysql.
- 实名验证接口使用国家规定实名接口。

游戏接入

游戏异步调用**Get**查询，没有返回实名信息，在适当位置显示实名认证图标，用**Post**方式添加玩家实名认证。

1) 请求实名状态

<https://cid.gzyouai.com/api/v2/default/cid>

调用方式: **HTTP get**

接口描述:

请求参数:

GET参数:

字段名称	字段说明	类型	必填	备注
oid	渠道返回唯一账号id(接入公共的建议是"userType_渠道openid")	string	Y	
key	公共接口分配的游戏项目参数	string	Y	-

请求返回结果:

```
//请求返回信息 ymd:生日 online: 在线时长(分钟 没有设置过默认不传) amount:累计充值(元 没有设置过默认不传)
{"code":200,"data":{"ymd":"20200101",online:"60",amount:"150"},"msg":"ok"}
```

返回结果:

字段名称	字段说明	类型	必填	备注
code	状态	string	Y	200成功
data	内容	json	Y	成功为实名后年月日,不存在为空json对象,如果认证中YMD记录为“1”
msg	提示信息	string	Y	

例子:

[https://cid.gzyouai.com/api/v2/default/cid?](https://cid.gzyouai.com/api/v2/default/cid?oid=userType_4243289&key=cda56c8c89fbd38adc51bb42dc3eb8d)
[oid=userType_4243289&key=cda56c8c89fbd38adc51bb42dc3eb8d](https://cid.gzyouai.com/api/v2/default/cid?oid=userType_4243289&key=cda56c8c89fbd38adc51bb42dc3eb8d)

2) 实名验证接口

<https://cid.gzyouai.com/api/v2/default/cid>

调用方式: **HTTP post**

接口描述:

请求参数:

GET参数:

字段名称	字段说明	类型	必填	备注
oid	渠道返回唯一账号id	string	Y	-
num	用户绑定身份证	string	Y	-
name	用户姓名	string	Y	-
key	公共接口分配的游戏项目参数	string	Y	-

请求返回结果:

```
//
{"code":200,"data":"","msg":"ok"}
or
{"code":10001,"msg":"已存在该实名验证","data":""}
```

返回结果:

字段名称	字段说明	类型	必填	备注
code	状态	string	Y	200成功,10001, 已存在该实名验证
data	内容	json	Y	为空json对象
msg	提示信息	string	Y	长度最长32位, 不然会返回“参数错误”

例子:

```
curl -X POST https://cid.gzyouai.com/api/v2/default/cid -d 'oid=userType_4243289&num=330502880702447&name=多多&key=cda56c8c89fbdc38adc51bb42dc3eb8d'
```

3) 上报信息接口 (更新在线时长和充值金额)

<https://cid.gzyouai.com/api/v2/default/update>

调用方式: **HTTP post**

接口描述:

请求参数:

GET参数:

字段名称	字段说明	类型	必填	备注
oid	渠道返回唯一账号id	string	Y	-
type	online/amount/login/logout	string	Y	要求 login/logout 必须接入
value	值	string	N	online/amount时需要填新增金额或时长
key	公共接口分配的游戏项目参数	string	Y	-
di	设备号	string	Y	

请求返回结果:

```
{"code":200,"data":"","msg":"ok"}
or
{"code":400,"msg":"请求参数错误","data":""}
```

返回结果:

字段名称	字段说明	类型	必填	备注

code	状态	string	Y	200成功
data	内容	json	Y	为空json对象
msg	提示信息	string	Y	

例子:

```
curl -X POST https://cid.gzyouai.com/api/v2/default/update -d 'oid=userType_4243289&type=online&value=10&key=cda56c8c89fbd38adc51bb42dc3eb8d&di=xxx'
curl -X POST https://cid.gzyouai.com/api/v2/default/update -d 'oid=userType_4243289&type=amount&value=10&key=cda56c8c89fbd38adc51bb42dc3eb8d&di=xxx'

curl -X POST http://127.0.0.1:8000/api/v2/default/update -d 'oid=userType_4243289&type=amount&value=10&key=cda56c8c89fbd38adc51bb42dc3eb8d&di=xxx'
```

其它接口待添加

总结

```
//请求实名接口
curl https://cid.gzyouai.com/api/v2/default/cid?oid=userType_4243289&key=cda56c8c89fbd38adc51bb42dc3eb8d
//验证实名接口
curl -X POST https://cid.gzyouai.com/api/v2/default/cid -d 'oid=userType_4243289&num=330502880702447&name=多多&key=cda56c8c89fbd38adc51bb42dc3eb8d'
//更新月充值和日时长
curl -X POST https://cid.gzyouai.com/api/v2/default/update -d 'oid=userType_4243289&type=online&value=10&key=cda56c8c89fbd38adc51bb42dc3eb8d&di=xxx'
curl -X POST https://cid.gzyouai.com/api/v2/default/update -d 'oid=userType_4243289&type=amount&value=10&key=cda56c8c89fbd38adc51bb42dc3eb8d&di=xxx'

//上报登陆登出
curl -X POST https://cid.gzyouai.com/api/v2/default/update -d 'oid=userType_4243289&type=login&key=cda56c8c89fbd38adc51bb42dc3eb8d&di=xxx'
curl -X POST https://cid.gzyouai.com/api/v2/default/update -d 'oid=userType_4243289&type=logout&key=cda56c8c89fbd38adc51bb42dc3eb8d&di=xxx'
```

接入发送申请邮件

上面地址为测试地址方便接入，游爱内部游戏需要在政府实名后台获取到参数(正式参数)后，添加106.54.82.128和121.5.210.49到IP白名单，发送邮件到技术中心申请对接参数，发给运营中心负责人赵毅春 zhaoyichun@gzyouai.com 抄送 赵楚凡 zhaochufan@gzyouai.com 莫洪章 mohongzhang@gzyouai.com 外部游戏请找平台对应人员。

```
[标题]项目组-游戏名-申请实名接口参数
[项目负责人] xxx
[运营对接人员] xxx qq/手机
[对接目的] 游戏自研联运发行/其它
[日均在线用户数] 400w
[对接公共SDK] 是/否
[参数] 游戏简称/appId/bizId/secretKey
[是否已添加白名单] 是/否
例：游戏简称: fytX
    appId:test-appId(正式参数)
    bizId:test-bizId(正式参数)
    secretkey:xxxxxx(正式参数)
    主体公司名称: xxx
```

需要多套参数的提交多个游戏。服务端接入用文档接口

公共实名安卓SDK接入文档

公共实名安卓SDK接入文档

接入准备 [点我跳转下载SDK相关资源](#)

！！本SDK用途为政府实名认证上报工作！！

1. 找相关负责人获取参数

serverUrl: 服务器域名 默认 <https://cid.gzyouai.com>
gameSimpleName: 游戏在公共后台的项目版本代号 如 fytx_test
key: 公共接口分配的游戏项目参数

2. 拷贝jar包文件 **PublicFcmSDK.jar** 到你的项目工程

3. 在你的项目工程 **AndroidManifest.xml** 文件中添加

```
<activity
    android:name="com.publicfcm.sdk.PublicFcmSDKUserCertificationView"
    android:configChanges="keyboard|keyboardHidden|orientation|screenSize|screenLayout"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.FullScreen">
</activity>
```

接口介绍

1. init 初始化（必接）

Activity	上下文
serverUrl	服务器域名
gameSimpleName	游戏简称
key	分配的上传key

PublicFcmSDKListener	SDK回调
----------------------	-------

```
PublicFcmSDK.getInstance().init(Activity activity, String serverUrl, String gameSimpleName, String key, PublicFcmSDK.PublicFcmSDKListener publicFcmSDKListener);
```

例子

```
PublicFcmSDK.getInstance().init(this, "https://cid.gzyouai.com", "default", "cda56c8c89fbdc38adc51bb42dc3eb8d", new PublicFcmSDK.PublicFcmSDKListener() {
    @Override
    public void userInfo(int totalGameTime, int totalPayMoney, String birthday) {
        PoolSdkLog.logError("游戏收到回调: totalGameTime:" + totalGameTime + ",totalPayMoney:" + totalPayMoney+",birthday:"+birthday);
    }

    @Override
    public void error(String errMsg) {
        PoolSdkLog.logError("游戏收到回调: errMsg:" +errMsg);
    }

    @Override
    public void result(Object data) {
        PoolSdkLog.logError("游戏收到回调: result:" +data);
    }
});
```

2. getUserInfo 获取用户信息（选接）

Activity	上下文
openID	用户openID

```
PublicFcmSDK.getInstance().getUserInfo(Activity context,String openID);
```

例子

```
PublicFcmSDK.getInstance().getUserInfo(this, "550_123");
```

3. updateUserData 上传用户行为（必接）

Activity	上下文
openID	用户openID
type	登录为 PublicFcmSDK.FcmSDKUserType.LOGIN 下线为 PublicFcmSDK.FcmSDKUserType.LOGOUT

```
PublicFcmSDK.getInstance().updateUserData(Activity context, String openID, String type);
```

例子

```
PublicFcmSDK.getInstance().updateUserData(this, "550_123", PublicFcmSDK.FcmSDKUserType.LOGIN);
```

4. openUserCertificationView 打开实名认证界面（**4、5必接1个**）（游戏项目无自己实名认证界面可以用公共统一提供的）

Activity	上下文
openID	用户openID

```
PublicFcmSDK.getInstance().openUserCertificationView(Activity context,String openID);
```

例子

```
PublicFcmSDK.getInstance().openUserCertificationView(this, "550_123");
```

5. setUserCertification 实名认证（**4、5必接1个**）（游戏项目有自己的实名认证界面可以调用该接口进行实名信息上报）

Activity	上下文
openID	用户openID
name	姓名
idNumber	身份证号码

```
PublicFcmSDK.getInstance().setUserCertification(Activity context,String  
openID,String name,String idNumber);
```

例子

```
PublicFcmSDK.getInstance().setUserCertification(this,"550_123","尚福乐",  
"211322198509260317");
```

6. PublicFcmSDK.PublicFcmSDKListener 回调

userInfo	获取用户信息接口 getUserInfo 的回调
error	上报接口报错信息回调
result	上报用户行为和实名认证结果回调

```
public abstract interface PublicFcmSDKListener {  
    public abstract void userInfo(int totalGameTime, int totalPayMon  
ey, String birthday);  
    public abstract void error(String errMsg);  
    public abstract void result(Object data);  
}
```

新海豹SDK操作流程

极光一键登陆**SDK**参数的相关获取和配置

1. 打开极光**SDK**的后台

<https://www.jiguang.cn/dev2>

2. 创建新的应用

3. 选择极光认证

4. 填写包名等信息（包名跟极光**SDK**的**AppKey**还有签名绑定）

包名设置后就不能修改了，需要在极光后台把整个应用删除

签名如果用打包工具默认的签名填 **4e98d7275d47959beea11bfe20d8e7b2**

5. 获取极光**SDK**的**AppKey**和**Master Secret**

6. 设置一键登陆**SDK**的**RSA** 加密公钥

RSA 公钥

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDuei8/MhhBh8QpuwJoCufSNBfYdQZhW5wE
Nso+2vbHQSTprfhTWlNZNCpGeyYjOnq66JCY8QNBVcQRR+K/bPojaK5UG0VDSshmK2TrsKbV
z7wCAr18ug2xwRymDUP50Vg7gIeH69ez8oSXyn6L2rN51fkhi76sqF5SQ07TDREx9QIDAQAB
```

海豹后台一键登陆参数的设置

1. 填写上面步骤获取到的相关参数

RSA 私钥

```
MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwgGJdAgEAAoGBA056Lz8yGEGHxCm7AmgK59IO
F9h1BmFbnAQ2yj7a9sdbJ0mt+FNau1k0KkZ7JiM6erokJjxA0FVxBFH4r9s+iNorlQY5UNK
GaRHZ0uwptXPvAICvXy6DbHBHKYNSnk5WDuAh4fr17PyhJfKfovas3nV+SGLvqyoXlJA7tMN
ETH1AgMBAAECgYEA0t+KLch/0Fr0M65g111lvj0oLkh0xtw53mQQfHoko1bQbC3gmtwRPZzq
EP70X3JTmjkrU5vKtBhB0sINGBYmLDpYmVlQSIKtpnMfUAJZts3eyJQAV30g2Za3xJLYvR4M
68y40LiybTTth2c+M+rvRzJf55vfsdygVxV0ypyeDTUCQD7R3oM4A7Pun1o+u99dSauEkn6
j6n1TPiidyAQAWN72jw2UmlLp7pfs15wXJ8Xf8WS5+uW+1KfR7aUhtis+/iDAkEA8vUjHPsY
E/I8uAkc17HyPo9QN3XKnxbZ3LM40u5bop2j/7ZJPmaBWJ8NdecfWfU+go0Qo0jwVyLhmpGZ
gqxyJwJAUnMhmRfrxD1s6QVtj7zg9D0MCSsamWKOmqbmAFp0Si9RkoF4C1r9zEaatk0t8Af8
zDdNFj1g3uKpBhM5BYmoAQJBAMuHKdwcSMZk1hLBp4TVz/4yZlufdJYA9DKbfM302br3XixY
EzGQA2kvDMKpVC0y8iPBcp+m+Ze1Y9Lth/Dc718CQHZZu7KZn5yYo1pVw6JJv/WNHDNRjdj7
NpdptcNCvbDKFxyqJAtfm0HWUdLrkFDe+UVNe0PJmKa3UV0c0WnJrfY=
```

RSA公钥私钥的生成

<http://web.chacuo.net/netrsakeypair>

或

<http://www.metools.info/code/c80.html>

或

文明用语检查服务

游戏角色姓名及聊天检查服务游戏接入

游戏异步调用用Post方式查询

1) 请求实名状态

```
https://cid.gzyouai.com/api/v2/default/filter
```

调用方式：**HTTP POST**

接口描述：

请求参数：

POST参数：

字段名称	字段说明	类型	必填	备注
content	检测内容 长度限制30	string	Y	
key	公共接口分配的游戏项目参数	string	Y	-

请求返回结果：

```
//请求返回信息 code == 200 没包含不正确字符串, code = 10005 包含, data 为过滤后内容, 非法标点符号会被自动过滤
{"code":200,"msg":"ok","data":"原请求内容"}
{"code":10005,"msg":"共建更好交流环境","data":"ddfsf**dsfddsfd**鱼**"}
```

返回结果：

字段名称	字段说明	类型	必填	备注
code	状态	string	Y	200不含/10005 含不正确字符串
data	内容	json	Y	过滤后内容，非法标点符号会被自动过滤
msg	提示信息	string	Y	

例子：


```
curl -X POST https://cid.gzyouai.com/api/v2/default/filter -d 'content=ddfsf 网/* 管dsfd dsfd&key=cda56c8c89fbd38adc51bb42dc3eb8d'
```

投放包体测试事项

安卓APK母包测试事项

新接入游戏母包测试

运营在拿到游戏提供对接完成公共SDK之后的apk（以下称为母包）先进行一轮正常的流程测试

1. 登录界面是否正常打开，是否能过正常登录进入游戏

2. 公共**SDK**在母包会登录之后会有一个红色的悬浮标（仅测试显示）点击可查看到母包各个接口的调用情况

3. 接口测试

其中 `onPause`、`onStop`、`onResume`、`onRestart`、`onStart` 需要在游戏中 按一次home键 回到桌面再回来游戏才可以测试

`onDestroy` 是结束游戏调用（运营无须关注）

进入游戏 接口对应公共后台的角色信息接口

运营需要核对游戏传入的角色参数是否正确（有无角色名等等）和 公共后台是否能查到相关角色信息

如果确认游戏调用 进入游戏 但是公共后台无数据请反馈到公共SDK技术这边检查

4. 支付测试

测试支付是否正确调起（需先在公共后台配置 IP白名单 ）

点击进行沙箱支付游戏是否到账

另：如果游戏调用 创建角色 接口后没有重新 进入游戏 接口的话，是无法调起支付的
顺序不能反！！

安卓9133SDK测试事项

1. 打包9133带OAID版本SDK事项

2.0以上OAID SDK需要一个包名对应一个证书才可以调用

1. 证书的获取申请（联系运营 [赵艳](#)）

证书需要填写 `example_batch.csv` 后发送到 `msa@caict.ac.cn` 申请
注意每个包名对应一个签名，申请时需要将所需申请的全部包名填写到表格中。

2. 打包事项

把对应的包名的证书文件放到SDK assets资源文件夹

3. 核对文件名跟公共后台配置是否对应

注意文件名带后缀

如 com.youai.dayu.yzbxy.cert.pem

对应后台参数

4. 测试

以上操作跟参数都核对无误之后打包测试

在公共后台的 角色日志 表中的 其他 信息的 第五项 代表客户端的 oaid 获取，如果常见的手机测试都能获取到oaid代表以上操作都无误

5. 头条SDK测试

核对头条后台联调的appid跟公共后台的头条appid是否一致

联调测试 查看 激活、注册、充值事件是否上报成功 金额是否正确

微信小游戏测试事项

1. 微信小游戏星云投放测试

游戏可以在模拟器中的启动参数加上 `channel=XXX`

对应星云投放的3套不同的参数进行初始化上报

不同的channel值会对应公共后台日志的渠道标识1

安卓星云投放SDK测试事项

安卓星云投放SDK依赖OAID SDK

OAID相关配置方法见文档 -》[9133SDK](#)

打包之后游戏按流程走一遍

在星云后台查询相关数据是否有上报即可

1. 在线时长（心跳上报）
2. 登录
3. 注册
4. 创角
5. 进入游戏
6. 升级
7. 提交充值

对内安卓文档

1)SDK添加渠道对接文档(内部)

SDK添加渠道对接文档（内部接入使用）

一、创建项目：

1、在 **_Tool** 目录里的 **CreateProject(as)** 目录下使用 **CreateProject.bat** 创建添加渠道的模板工程，该工程为 **android studio** 工程，（需先在电脑上安装 **python3**）

二、对接添加渠道接口

1、注意。导入创建的模板工程，需要添加渠道对接接口代码写在模板工程下的 **PoolProxyChannel** 类。

onCreate中一般添加渠道的初始化功能，在渠道初始化有回调通知的时候需要在初始化接口调用之前设置

```
PoolSdkHelper.hasInit = true(默认是false)
```

（若果渠道SDK无初始化回调，则设置为false即可，同时下面的“渠道的回调方法中将结果通知游戏”操作则不再需要），设置为true后在渠道的回调方法中将结果通知游戏，如下例：

```
if(isSuccess){
    callBackListener.poolSdkCallBack(PoolSDKCode.POOLSDK_INIT_SUCCESS, msg);
}else{
    callBackListener.poolSdkCallBack(PoolSDKCode.POOLSDK_INIT_FAIL, msg)
;
}
```

2、接口说明（重点）

获取渠道参数说明：

PoolSdkConfig.getConfigByKey("appId")

PoolSdkConfig.getConfigByKey("appKey")

测试时直接将对应的值填到assets/poolsdk.xml下，Key 根据渠道sdk而定，测试后填到公共SDK网页平台上，保持命名一致。

登录：

login(Activity paramActivity, String paramCustom)

参数名称 参数说明

paramActivity Activity

paramCustom 扩展参数（直接返回回游戏）

在**login**方法里面写调用渠道的**login**接口，调用渠道**login**接口返回成功的地方调用**loginCheck**进行与PoolSDK服务端登录验证，样例如下：

调用渠道**login**接口返回失败处就直接返回失败，例如：

充值接口：

pay(Activity paramActivity, final PoolPayInfo yaPayInfo)

参数名称 参数说明

paramActivity Activity

PoolPayInfo 充值参数实体类

在**pay**方法中处理要添加渠道的充值方法

在调用渠道充值接口前要先与PoolSDK服务端创建充值订单，在创建订单成功回调方法中处理添加渠道的充值相关，订单创建失败直接回调充值失败方法。代码样例如下：

注意：创建订单成功后的queryId必须传给服务器，一般传sdk自定义参数字段，没有要和服务器商议看放在哪个字段传。

设置横竖屏样例：

提交角色数据接口：（**submitRoleData**）

submitRoleData(Activity paramAct,PoolRoleData,prd)

参数名称 参数说明

Activity activity

PoolRoleData 提交角色数据实体类

代码样例如下：

打开渠道中心接口：

如果sdk有用户中心接口，在hasChannelCenter方法中设置返回true,并在openChannelCenter接口实现打开SDK用户中心，没有则返回false.

openChannelCenter(Activity paramActivity)

代码样例：

切换账号接口：

如果渠道SDK有切换账号接口，则在hasSwitchAccount设置返回true，并实现switchAccount方法；如果SDK没有切换账号接口，则直接返回false，无需实现switchAccount方法。

switchAccount(Activity paramActivity,String paramCustom)

在此方法中调用渠道的切换账号接口

显示退出框接口

如果sdk有退出页面，hasExitDialog返回true,在下面接口实现，没有则返回false.

showExitDialog(Activity paramActivity)

注意：在相应时候返回 确认退出和取消退出，对应的Code 在PoolSDKCode,要统一使用。

游戏退出清理数据接口：（逐步不使用，清理sdk数据统一在sdk退出接口或者是onDestroy()

exitGame(Activity paramActivity)

打开论坛接口：

openForm(Activity paramActivity)

生命周期接口：

根据渠道SDK要求接入相应的生命周期方法。

至此，渠道**SDK**的接入已经完成。

接入完毕提交前自检步骤：

1、登陆后收到数据尽量多的返回给游戏，只要是sdk有返回的是否已经全部传递，充值queryid传递正确，汇率正确。

2、监听接口返回给游戏：登陆失败，取消登陆，退出游戏，取消退出，可根据PoolSDKCode里面的字段检查，sdk有的是否返回监听。

附加注意点：

1、当渠道存在微信支付时需要添加微信对应的WXEntryActivity时，将接入Demo的apk使用apktool工具进行反编译，然后找到对应的WXEntryActivity.smali文件，将此文件内的所有包含包名的地方替换为Lcom/myexample/wegame，之后将此文件放到ForFiles文件夹下，最后在config.xml中添加配置（具体可以参考Tools目录下的微信模版）

2、在渠道需要修改参数配置时注意点：（参考ysdk渠道）

渠道参数文件要在assets目录中，渠道参数字段需要一行对应一个字段，后台填写字段要与配置文件字段一一对应。Config.xml配置如下：

```
<operate from="assets" name="ysdkconf.ini" type="modifyConfig" />
```

3、在需要修改打包后apk中manifest中四大组件标签的增加，删除，修改操作时需要到config.xml添加配置：（可参考Tools下 对应的manifest模板）

=====

1、如渠道要求继承他们自身的application类

只需要在 PoolSDKApplication 继承他们的即可

```
publicclass PoolSDKApplication extends TWApplication {
```

```
}
```

2、如在此类需要使用自定义参数要先调用

`PoolSdkConfig.readPoolSdkConfigData(this);`

```
@Override
public void onCreate() {
    super.onCreate();
    PoolSdkConfig.readPoolSdkConfigData(this);
    PoolSdkConfig.getConfigByKey("APPID");
}
```

3、`PoolUtils.isLandscape`在`application`类此方法判断横竖屏无效，请使用上面第二点的方式让游戏自己填写横竖屏

```
@Override
public void onCreate() {
    super.onCreate();
    PoolSdkConfig.readPoolSdkConfigData(this);
    PoolSdkConfig.getConfigByKey("APPID");
    if (PoolSdkConfig.getConfigByKey("landscape").equals("true")) {
        //横屏
    } else {
        //竖屏
    }
}
```

4、渠道SDK的生命周期等方法都写在类中

如渠道SDK有初始化回调操作（即未收到初始化成功不能调用其他的接口）

则在`onCreate`接口把 `PoolSdkHelper.hasInit = true`;设置为true

然后在渠道SDK接口回调中调用

//通知游戏渠道SDK初始化成功

```
instance.callBackListener.poolSdkCallBack(PoolSDKCode.POOLSDK_INIT_SUCCESS,
PoolSDKCode.$初始化成功$);
```

或者

//通知游戏渠道SDK初始化失败

```
instance.callBackListener.poolSdkCallBack(PoolSDKCode.POOLSDK_INIT_FAIL,
PoolSDKCode.$初始化失败$);
```

来通知到游戏客户端

5、在`PoolProxyChannel.java`类中，判断游戏横竖屏可以使用 `PoolUtils.isLandscape` 方法

6、通知游戏渠道SDK已经注销账号成功 `logoutListener.onLogoutSuccess();`

7、切换账号成功通过`loginCheck`服务端校验成功返回新的登录信息给游戏

8、渠道SDK有用户中心

[@Override](#)

```
public boolean hasChannelCenter(Activity paramActivity) {
// 这里返回给游戏渠道SDK是否有用户中心接口
return true;
}
```

```
@Override
public void openChannelCenter(Activity paramActivity) {
    //在此处调用渠道SDK打开用户中心
    XXX.openChannelCenter();
}
```

9、渠道SDK有自己的退出游戏面板

```
@Override
public boolean hasExitDialog() {
    // 这里返回给游戏渠道SDK是否有退出游戏面板
    return true;
}
```

此处红色接口为某渠道SDK的退出接口例子

```
@Override
public void showExitDialog(Activity paramActivity) {
    XXX.showExitDialog(paramActivity, new XXXCallback() {
        @Override
        public void onSuccess() {
            // 回调游戏退出成功
            exitDialogListener.onDialogResult(PoolSDKCode.EXIT_SUCCESS,
            PoolSDKCode.$退出成功$);
        }

        @Override
        public void onCancel() {
            // 继续游戏
            exitDialogListener.onDialogResult(PoolSDKCode.EXIT_CANCEL, P
            oolSDKCode.$取消退出$);
        }
    });
}
```

10、渠道SDK有注销账号接口（注销的回调接口需调用第6点方法 `logoutListener.onLogoutSuccess()`; 通知游戏）[@Override](#)

```
public boolean hasLogout() {
```



```
return true;
}
```

```
@Override
public void logout(Activity paramActivity) {
    XXX.logout();
}
```

11、渠道SDK有自定义的闪屏，需要我们继承他们的闪屏入口，跳转到游戏Activity可以使用

```
Intent intent = new Intent();
PoolSdkConfig.readPoolSdkConfigData(this);
ComponentName component = new ComponentName(this.getPackageName(),
PoolSdkConfig.getConfigByKey("MainActivity"));
intent.setComponent(component);
startActivity(intent);
this.finish();
MainActivity=""配置游戏的 launcher activity 路径
```

SDK的配置文件需要添加

其中 ReplaceManifestTagB 为后台配置的参数 填入游戏的 launcher activity 路径

12、提交角色参数接口

```
@Override
public void submitRoleData(Activity paramActivity, PoolRoleInfo poolRoleInfo) {
    // String type = poolRoleInfo.getCallType();
    // if (type.equals(PoolRoleInfo.Type_CreateRole)) {
    // 写入渠道SDK提交创建角色信息
    // } else if (type.equals(PoolRoleInfo.Type_EnterGame)) {
    // 写入渠道SDK提交进入游戏信息
    // } else if (type.equals(PoolRoleInfo.Type_RoleUpgrade)) {
    // 写入渠道SDK提交角色升级信息
    // }
}
```

公共SDK只有 创建角色、进入游戏、角色升级这3种状态（有些渠道需要提交注册时的数据，我们可以使用创建角色这个事件来代替注册事件）

根据判断接入渠道SDK的提交角色参数接口

13、登录接口接入

```
@Override
public void login(Activity paramActivity, String paramCustom) {
    this.loginCustomString = paramCustom;
    XXX.login(paramActivity, new DouyuCallback() {
```

```

    @Override
    public void onSuccess(DouyuSdkParams data) {
        //渠道返回登录成功的账号信息，需要调用loginCheck接口来进行公共平台服
        务端进行校验
        loginCheck(data);
    }

    @Override
    public void onError(String arg0, String arg1) {
        //通知游戏登录失败回调
        loginListener.onLoginFailed(PoolSDKCode.$登陆失败$ + arg0 + a
        rg1);
    }
});
}

private void loginCheck(DouyuSdkParams data) {
    PoolLoginInfo info = createLoginInfo();
    info.setToken("" + data.get("sid", ""));
    info.setTimestamp(System.currentTimeMillis() + "");
    info.setOpenId("");
    info.setCustom(this.loginCustomString);
    info.setUserName("");
    info.setOther(DouyuGameSdk.getSDKVersion());
    PoolLoginChecker loginCheck = new PoolLoginChecker(info, this.loginL
    istener);
    loginCheck.startCheck();

    //下面的为重写loginChecker，部分渠道SDK需要服务端校验的二次信息可以在此获得
    // PoolLoginChecker loginChecker = new PoolLoginChecker(info, new
    // PoolLoginListener() {
    //
    //
    // @Override
    // public void onLoginSuccess(PoolLoginInfo loginInfo) {
    // loginListener.onLoginSuccess(loginInfo);
    // }
    //
    //
    // @Override
    // public void onLoginFailed(String msg) {
    // loginListener.onLoginFailed(msg);
    // }
    // });
    // loginChecker.startCheck();
}

```

14、支付接口

```

@Override
public void pay(final Activity paramActivity, final PoolPayInfo poolPayInfo) {
    PoolPayCreateOrder createOrder = new PoolPayCreateOrder(poolPayInfo,
        this.sdkUserId, createPayCreateOrderUrl());
    createOrder.createOrder(paramActivity, new PoolPayOrderListener() {
        @Override
        public void onCreateOrderSuccess(PoolPayOrderInfo poolOrderInfo) {
            //在此处调用渠道SDK的支付接口，订单号透传参数需要传入 poolOrderInfo.
            getQueryId()
        }

        @Override
        public void onCreateOrderFailed(String errorMsg) {
            if (payListener != null)
                payListener.onPayFailed(poolPayInfo.getCustom(), errorMsg);
        }
    });
}

```

`createPayUrl()`返回的为后台对应生成的回调地址（有些SDK的支付回调地址是在客户端上传的，这个时候就使用这个方法进行获取支付回调地址然后调用支付接口的时候传进去；有些SDK的支付回调地址是在渠道后台进行设置的，这种情况下就由运营进行配置，我们不需处理）

2)SDK资源抽取提交到公共后台

一、说明：依照《**SDK添加渠道对接文档(内部).docx**》文档，把渠道**SDK**接入到我们的**poolsdk**中间件之后，需要将相应的资源按一定的规范提交到公共**SDK**后台供打包工具下载。

二、资源组织规范：

如下图所示：

ForAssets文件夹：存放的是安卓工程中**assets**文件夹中的资源；

ForFiles文件夹：存放的是微信支付回调页**WXEntryActivity.smali**和**WXPayEntryActivity.smali**（如果渠道**SDK**存在微信支付的情况下，并且有这两个文件的要求下，就把它的**smali**文件保存在此次。如果没有，可以留空）；

ForLibs文件夹：存放的是**so**库；

ForRes文件夹：存放的是安卓工程中的**res**资源；

unknown文件夹：存放的是**apktool**反编译出来的**unknown**文件夹中的文件；

classes.dex文件：**apk**经过特殊处理后得到的**dex**文件（可通过在**_Tool**文件夹里**ExtractCode**文件夹中的**ExtractCode.bat**脚本进行抽取）；

config.xml文件：打包工具的配置文件；

ForManifest.xml文件：安卓工程中**manifest**中的文件，需去掉启动**activity**；

poolsdkfile.zip文件：该文件为以上文件的压缩，最终提交到公共**SDK**后台的也是此文件；

三、资源在仓库中的存储和命名规范：

如下图所示：

渠道SDK的简称：一般为渠道**SDK**名字的拼音全拼，要求与公共后台填写的简称一样（主要是方便以后的查找）

SDK版本号：要求命名要与渠道给过来的**SDK**文件标识的版本号一致（也是为了方便以后的查找）

foolsdkfile：该文件夹存放的就是资源组织规范中展示的内容

四、注意点：

- 1) 提取assets中的资源时，除了在安卓工程中的assets文件夹中的资源，也要留意有些jar包中也夹带着assets资源，这些资源也是需要复制到ForAssets文件夹中的。这里建议，使用apktool直接反编译apk，然后得出的assets的资源即为所有的assets资源。
- 2) ForFiles中关于微信的两个文件的samli资源的，请参考《SDK添加渠道对接文档(内部).docx》文档中的附加注意点的第一点。
- 3) ForRes中的资源，由于这资源是用来与游戏母包的资源进行合并再打包的，所以很可能存在资源同名的冲突，所以我们要去掉一些通用资源。例如:string.xml中的app_name，drawable中的ic_launcher.png，layout中的activity_main.xml等。否则打包工具因为资源冲突而打包失败。
- 4) ForManifest中复制的是渠道SDK里面必须的权限和四大组件以及各种meta-data的配置，尽量不要复制测试用例相关的资源。
- 5) config.xml打包工具配置文件，有些特殊功能的配置，可以参考《SDK添加渠道对接文档(内部).docx》文档中的附加注意点的第二点和第三点。

五、gradle自动抽取资源文件：

前四点是资源抽取的说明以及方法，这里提供一种快捷的抽取资源的方法，该方法基于gradle自动化构建系统。具体使用方法，请查看文档《gradle快捷抽取资源》。

3)gradle快捷抽取资源

一、环境：1）项目工程需使用**android studio**进行开发

2）需要在**window**系统开发（若使用苹果等系统的话，请自行修改相应脚本）

3）电脑需要安装**python3**

4）目前使用的**gradle**版本为**gradle-4.1-all**，安卓**gradle**插件版本为**com.android.tools.build:gradle:3.0.0**

二、配置：以下基于工程为**as**工程为前提（如果当前工程为**eclipse**项目，则跳到第四点转**eclipse**工程为**as**工程，然后再回到这里来看）

1）将工程切到**Android**，如果工程的**module**名称不是**app**，则先把**module**名称改为**app**。

2）查看该文件是否存在，如果不存在，则创建一个该文件，或者从已存在该文件的工程中复制一份放到这里，如下图为该文件里的配置：

1处为默认值，勿修改；

2处为**SDK**的简称；

3处为**SDK**的版本号；

4处为抽取资源的方式，值为1时为普通获取资源方式（就是直接从工程的**res**中复制资源到**ForRes**中，通常都是使用这种方式抽取资源。但是这种方式有个缺点，如果**SDK**方直接使用**R.id.xx**、

R.drawable.xxx等**R**文件的方式获取资源的时候，使用我们打包工具打包会报错，原理自行摸索了解），值为2时是特别的方式抽取资源（就是通过**apktool**反编译接好**SDK**的**apk**，然后把反编译出的资源复制到**ForRes**中，理论上该方式可以适合所有的场景，但是它也有缺点，就是它很容易跟母包的资源**id**发生冲突，遇到资源**id**冲突的情况下我们可以手动去掉相应的资源**id**然后把文件发给游戏方进行打包。建议：遇到**SDK**方直接使用**R.id.xx**、**R.drawable.xxx**等**R**文件的方式获取资源的时候或者**SDK**方提供的是**aar**包或者**combile**的库含有大量资源文件的时候如**v7**包时，可以选用该方式）；

5处创建新工程时用到，跟当前工程无关，但你需要生成一个新项目的时候才会使用到它。它后面的值，即是要创建的新工程的**sdk**简称。

3）检查**module**为**app**的**build.gradle**文件，

查看是否存在这些配置，如果不存在，则把以下代码复制到这个build.gradle文件中。

```
def rootPath = project.rootDir.getAbsolutePath()
def utilPath = rootPath.substring(0, rootPath.lastIndexOf('\\')) + "/PoolSdk_AsParent/app/util.gradle"
apply from:utilPath
```

4) 检查project的build.gradle文件，查看里面的配置，看是否有下图圈到的，没有则添加。

三、使用该自动化系统：

使用方法：在android studio中的下输入相应命令。

1) 一键抽取资源：gradlew OBC

（说明：根据第二步中的配置，SDK简称以及SDK版本号，直接在_SDK文件夹中生成资源文件）

2) dex文件修改：gradlew changeDex

（说明：如果项目中只是代码逻辑修改了或者jar包替换，资源文件不变，使用该方法可以快速修改之前以及生成的dex）

3) 一步打开抽取出来的资源所在的文件夹：gradlew OSDK

（说明：该方式主要是为了快速打开到资源位置，而不需要一个一个文件夹的点开，无需人工一个个查找）

4) 快捷生成新工程：gradlew CNP

（说明：在gradle.properties文件中配置的

，这里就可以用得上了，根据这个值我们可以生成一个名为PoolSdk_huoerlanTT的新工程。省去了到文件中找出脚本再生成工程的时间。）

5) 更多的快捷方法请自行开发

四、转eclipse工程为as工程：

1) 说明：由于历史原因，我们的工程既存在eclipse工程也存在as工程。由于我们的gradle抽取资源的

系统是基于as的，所以推荐以后以as为开发工具。

2) 步骤:

如上图所示，我们选到菜单栏的File-->New-->Import Project，然后选择eclipse工程，接着点OK，接着一直点next，最后点完成即可。

五、善后工作:

执行完一键抽取资源（gradlew OBC）之后，一般来说我们可以直接打压缩包将资源提交到公共后台了。但是有些文件（如：ForAssets中的某些文件参考YSDK的ysdkconf.ini，或者Formanifest.xml）是需要打包工具进行打包的时候要替换某些值的，例如Formanifest.xml中的某个activity的方向设置，某个provider的主机名，又例如ysdkconf.ini文件中的QQ_APP_ID的值。由于这些都是动态的需求，所以目前该自动化系统无法处理这些。这种情况下我们就需要根据具体情况，或修改config.xml文件，或修改Formanifest.xml中的内容。具体可以参考文档

《打包工具配置相关》

六、modeType的选择以及问题处理:

在第二步（配置）中已经初步提到了两种方式的特点。

1)值为1的时候，资源的抽取是一种正常的方式，所以如果打包的时候出现资源冲突，我们可以根据打包工具的日志迅速查找到是什么资源冲突了，我们一般直接把SDK方的相应资源删掉，然后发给游戏替换即可。

2)值为2的时候，我们的资源抽取方式是使用了反编译后的资源，所以这里我们的资源会多次两个文件，一个是public.xml文件，另外一个ids.xml文件。但使用这种方式进行资源抽取发生跟游戏的资源冲突的时候，我们一般可以把public.xml文件以及ids.xml文件中的冲突资源删掉，然后把这两个文件发给游戏方替换即可。

七、额外要求:

1) 要求所有的资源要放到module为app的res文件夹中。

2) 要求YSDK中使用到的微信相关文件，如WXEntryActivity.java需要放到module为app的src下的任一个包中

八、自动抽取资源系统的原理：

该系统依赖于**gradle**自动化构建系统，辅以**python**脚本以及**window**的批语言。**gradle**插件的源码在**util.gradle**中可以查看。

4)打包工具配置相关

1、Formanifest.xml中需要替换某些字符串时，可以把这个字符串设置为ReplaceManifestTagA、ReplaceManifestTagB等，然后在公共后台中配置一样的字符串，打包的时候打包工具会识别并替换字符串。

如下图：

上图中我们想要修改activity的方向，所以我们使用了一个名为ReplaceManifestTagB的占位符，然后到公共后台中配置相对于的占位符，如下图：

游戏在接入的时候就会根据这个占位符填的值去替换为相应的值，达到修改manifest的目的。

2、对应Formanifest.xml中的meta-data标签里的值的修改，我们可以直接以meta-data标签的键作为占位符填到公共后台上，游戏在接入的时候就会根据这个占位符填的值去替换为相应的值。如下图：

相对应于上图的meta-data，我们可以到公共后台中配置，如下图：

3、在需要修改打包后apk中manifest中四大组件标签的增加，删除，修改操作时需要到config.xml添加配置，可以参考文档《SDK添加渠道对接文档(内部)》的附加注意点的第3点。

4、需要修改assets中或者其他非manifest的键值对文件时，需要对config.xml添加配置，具体查考文档《SDK添加渠道对接文档(内部)》的附加注意点的第2点。

5、上面的3和4点中的内容，具体可以参考项目YSDK以及项目乾游YSDK，对比一下它的config.xml以及公共后台中的配置，相信很容易就会发现规律。

6、在仓库中的_Tool文件夹中的一些关于配置的例子：

具体可以留意这两个文件中的config.xml以及ForAssets中的文件。

对内H5文档

公共**SDK-H5 JS**接入文档（内部）

一、项目获取

向小组负责人拿总项目所存放的SVN地址，拉取项目到本地。

可以看到类似上图的目录，其中：

- H5SDKTestPage.html和game_Simulate.js：分别是自测页面UI和代码。
- youaiH5SDK.js：中间层文件，用于提供接口给游戏并衔接我方接入的js文件代码。
- youaiH5SDK.d.ts：用于把js接口声明为ts接口，以便使用ts编写的游戏使用。
- youaiH5Proxy.js：用于接入渠道SDK逻辑，使用内部逻辑，提供接口给中间层，此处的youaiH5Proxy.js文件是模版文件，接入时请复制到相应位置再编辑。
- youaiH5Utils.js：存放了内部逻辑，连接服务端进行各种信息验证，提供接口给我方接入的js文件使用。
- md5.js：顾名思义，是MD加密类，这个文件不要改动。

youaiH5Proxy.js的存放需要注意：要创建一个文件夹以接入的SDK简称命名，再在这个文件夹里创建一个文件夹以接入的SDK的版本号命名，再将youaiH5Proxy.js模版复制到后创建的这个文件夹里再进行接入。

这里提到了SDK的简称以及版本号，需要到公共后台查看，如何查看请看第二小节。

二、公共后台配置项目

1. 公共后台地址也是找小组负责人要，然后点击图中红框按钮找到相应的页面，点击右上角便可以添加SDK。
2. 如上图，填写SDK名称、当前版本（为接入的SDK的版本，每次有版本更新记得修改这个版本号）、SDK简称（最好是SDK的英文名加上"_h5"）、客户端类别选择"H5"，最后记得点击右上角的保存。
3. 添加完后可以看到后台就有了新增的SDK信息如上，其中就可以看到SDK简称（其实就是你自己刚刚填写的），然后点击查看SDK版本。

4. 来到这个页面，这里可以看到该SDK已经添加的历史版本。可以点击对应版本右侧的编辑按钮对版本信息进行编辑，也可以点击复制按钮复制当前版本的复制版（这个操作用于新增版本的配置信息与上个版本的配置信息相同的时候）。点击右上角的新增SDK版本按钮添加新版本。

5. 填写SDK版本号、SDK版本代号（这个服务端需要用到，最好每次都询问一下服务端的接入人员）、SDK插入插件包下载地址栏上传接入文件youaiH5Proxy.js到后台。

三、代码接入

根据渠道SDK的文档在youaiH5Proxy.js文件里编写代码，各种接口都有相应注释，详见youaiH5Proxy.js文件。接入完毕后把youaiH5Proxy.js文件上传到后台。

对内苹果文档

1、目录说明

- channel:存放渠道相关资源
- channel_code:存放接入渠道的代码
- GKChannelSdk:神手，京有切支付sdk源码
- GKChannelSdk_no_third_pay:神手，京有不切支付sdk源码
- publicSdk:公共源码
- publicApp:Demo工程
- mix.py get_word_utils.py config_mix.py word_list.json 混淆相关脚本
- string_mix_public 公共字符串混淆脚本(python3 string_mix.py xxxx)
- string_mix_channel 渠道字符串混淆脚本(神手，京游)

2、渠道接入流程

- 1、在channel目录中创建一个渠道目录名称，用于存放渠道相关的接入资源
- 2、在原始工程的IIII_GKSdkChannel_JJJJ接入渠道的接口
- 3、在原始工程接入完成测试后，运行python3 mix 0 channel_name(渠道名称)
- 4、在1步骤中创建的目录中拿到接入好的资源
- 5、编写注意事项和渠道配置信息后跟4步骤中的资源一起打包发给游戏

乐游/飞跃内部文档

接口接入说明文档

话不多说，接入如下：

添加压缩包下lib下的 **LeYouSDK.aar**

1.application中的 onCreate 方法调用

```
CandySdk.getInstance().applicationStart(Application application,String appsFlyerKey);
```

- 参数说明

参数序号	参数说明
Application	application实例
String	appsFlyer申请的key

- 调用事例

```
CandySdk.getInstance().applicationStart(this,"xxx");
```

2.初始化

```
CandySdk.getInstance().init(Activity activity, CandySdkParams params, CandySdkCallBack callBack,CandyEventCallBack eventCallBack);
```

- 参数说明

参数序号	参数说明
Activity	上下文
CandySdkParams	初始化所需要的参数，详解如下
CandySdkCallBack	sdk回调，详解见附录
CandyEventCallBack	sdk上报给公共后台的事件回调，此参数为重载参数，不需要可以不用

- CandySdkParams参数说明

参数序号	参数说明
appId	运营给的参数
appKey	运营给的参数
styleName	样式，运营给的参数
serverUrl	请求头地址，后台给
isUseArea	跟运营商量是否使用地区功能，true 使用 false 不使用

- 调用事例

```

CandySdkParams params = new CandySdkParams();
params.setAppId("5ffd3ed2b6ee064e678d4a67");
params.setAppKey("5e77406f4e0710735f7795ec46550009");
params.setStyleName("sdk1_0_0");
params.setServerUrl("https://sdk.hahplatform.com");
params.setUseArea(true);
CandySdk.getInstance().init(this, params, new CandySdkCallBack() {
    @Override
    public void callback(int code, Object result) {
        switch (code){
            case CandySdkCallBack.INIT_SUCCESS:
                //初始化成功
                break;
            case CandySdkCallBack.INIT_FAIL:
                //初始化失败
                break;
            case CandySdkCallBack.LOGIN_SUCCESS:
                //登录成功
                CandyUserInfo userInfo = (CandyUserInfo) result;
                break;
            case CandySdkCallBack.QUERY_WITH_PRODUCTS_SUCCESS:
                //查询google后台商品成功
                break;
            case CandySdkCallBack.SWITCH_ACCOUNT_SUCCESS:
                //切换账号成功
                break;
            case CandySdkCallBack.AREA_DATA_CALLBACK:
                //切换区域回调
                break;
            case CandySdkCallBack.LOGIN_FAIL:
                //登录失败
                break;
        }
    }
}

```

```
        default:
            //默认不处理
            break;
    }
}
}, new CandyEventCallback() {
    @Override
    public void eventReport(String eventType, String msg, long
        timeInterval) {
        //公共事件上报回调，可不处理此回调
        //eventType 事件类型
        //msg 事件信息
        //timeInterval 事件消耗时间
    }
});
```

3. 登录

```
CandySdk.getInstance().login(this);
```

4. 支付

```
CandySdk.getInstance().pay(Activity activity, CandyPayInfo payInfo);
```

- 参数说明

参数序号	参数说明
Activity	上下文
CandyPayInfo	支付所需要的参数，详解如下

- CandyPayInfo参数说明

参数序号	参数说明
amount	金额
productId	内购id
roleId	角色id
roleName	角色名

serverId	区服id
serverName	区服名称
custom	自定义参数
cpOrderId	游戏的订单号
queryId	公共生成的订单号
exChange	比率
productIds	内购的所有商品id的集合，字符串数组

- 调用事例

```
CandyPayInfo payInfo = new CandyPayInfo();
payInfo.setProductId("1");
payInfo.setCpOrderId(System.currentTimeMillis()+"");
payInfo.setCustom("5286522f2dsfds");
payInfo.setRoleId("1234232");
payInfo.setRoleName("first time");
payInfo.setServerId("16");
payInfo.setServerName("test server");
payInfo.setAmount("10");
payInfo.setExChange(10);
CandySdk.getInstance().pay(MainActivity.this, payInfo);
```

5.查询商品信息

```
CandySdk.getInstance().querySkuDetailsAsync(要查询的商品id集合);
```

- 调用事例

```
List<String> list = new ArrayList<>();
list.add("1");
list.add("2");
list.add("3");
CandySdk.getInstance().querySkuDetailsAsync(list);
```

6.提交角色信息(玩家进入游戏的时候传)

```
CandySdk.getInstance().enterGame(Activity activity, CandyRoleInfo roleInfo);
```

- 参数说明

参数序号	参数说明
Activity	上下文
CandyRoleInfo	提交角色信息所需要的参数，详解如下

- CandyRoleInfo参数说明

参数序号	参数说明
serverId	区服id
serverName	区服名称
roleId	角色id
roleName	角色名

- 调用事例

```
CandyRoleInfo roleInfo = new CandyRoleInfo();
roleInfo.setRoleId("角色id");
roleInfo.setRoleName("角色名字");
roleInfo.setServerId("服务器id");
roleInfo.setServerName("服务器名字");
CandySdk.getInstance().enterGame(paramActivity, roleInfo);
```

7.打开个人中心

```
CandySdk.getInstance().openUserCenter(Activity activity);
```

- 参数说明

参数序号	参数说明
Activity	上下文

- 调用事例

```
CandySdk.getInstance().openUserCenter(this);
```

8.打开悬浮窗

```
CandySdk.getInstance().showFloatView(Activity activity);
CandySdk.getInstance().setFloatViewDefaultXY(int x , int y);//设置悬浮窗默认位置
```

- 参数说明

参数序号	参数说明
Activity	上下文

- 调用事例

```
CandySdk.getInstance().showFloatView(this);
CandySdk.getInstance().setFloatViewDefaultXY(0,150);
```

9.关闭悬浮窗

```
CandySdk.getInstance().hideFloatView();
```

10.onDestroy生命周期

```
CandySdk.getInstance().hideFloatView();
CandySdk.getInstance().onDestroySdk();
```

11.onActivityResult

```
CandySdk.getInstance().onActivityResult(int requestCode, int resultCode, Intent data);
```

- 参数说明

参数序号	参数说明
requestCode	请求码
resultCode	返回码
data	返回意图

- 调用事例

```
CandySdk.getInstance().onActivityResult(0, 0, data);
```

12.区域地址获取接口(选接)

```
String gameAreaUrl = CandySdk.getInstance().getGameAreaUrl();
```

附录

- CandySdkCallBack回调的code说明

code	参数说明
200	初始化成功
201	登录成功
202	支付成功
203	查询google内购商品成功
204	切换账号
205	facebook分享成功
100	初始化失败
103	支付失败
104	支付取消
105	查询google内购商品失败
106	facebook分享失败
401	区域选择结果回调

前后端联调文档

乐游服务端请求以及和前端交互的接口如下：

登录(和前端交互)

请求链接：https://sdk.hahplatform.com/static/sdk1_0_0/client.html?action=login&app_id=6007e26eb6ee064428e16d96&app_key=4c3ce5d31f10cfb3922c652063f519ed&isSwitch=true&uuid=&idfv=&device_info=10&ver=20210126×tamp=1611655228785&lng=en-US

参数	说明
action	动作，登录就为 login
app_id	运营给的参数
app_key	运营给的参数
isSwitch	是否是切换账号，是 true ， 不是 false
timestamp	时间戳
lng	当前用户的语言

第三方账号登录返回乐游账号(和后端交互)

请求事例：

https://sdk.hahplatform.com/sdk/sso/login?app_id=6007e26eb6ee064428e16d96&openid=112304055357683206858&sso_name=ACTION_S0FA&sso_type=google&sign=2a087a5bcaef15c3c41656186975c1ce

请求方式： GET

参数	说明
app_id	运营给的参数
openid	登录返回的

sso_name	第三方平台返回的用户名
sso_type	要绑定的平台，google 或 facebook
sign	请求链接加上 appkey的参数 再md5的值

返回事例：

```
{
  "code": 0,
  "msg": {
    "share_str": "",
    "username": "google_06858_67",
    "google": "112304055357683206858",
    "apple": "",
    "bind_phone": "",
    "open_id": 1549,
    "timestamp": 1611735800,
    "login_count": 4,
    "mail": "",
    "bind_mail_status": 0,
    "token": "5db0e83b4a69da9a6e2b6d1cf3d185b0",
    "notify_url": "https://sdk.hahplatform.com/service/confirm/google/5458c012e138231304af4216",
    "facebook": "",
    "login_key": "131e82317d50eb55cd35be161d5ea53d",
    "float_button": 0,
    "pay_channel_id": "5458c012e138231304af4216",
    "is_guest": 0,
    "kefu_info": ["Facebook: @Leyou"],
    "nickname": "google_06858_67",
    "share_pic": "",
    "bind_status": 0
  }
}
```

参数	说明
code	0 代表成功
float_button	悬浮球
login_count	登录次数
login_key	login_key
notify_url	确认订单地址
open_id	open_id
pay_channel_id	pay_channel_id
share_pic	facebook分享的图片链接
share_str	facebook分享的图片描述
timestamp	时间戳
token	token

乐游账号绑定第三方账号(和后端交互)

请求事例：

```
https://sdk.hahplatform.com/sdk/sso/bind?app_id=6007e26eb6ee064428e16d96&login_key=297fa17cfee1e02913603b1ddfecf5ff&open_id=1614&openid=112304055357683206858&sso_name=&sso_type=google&sign=bc7ead9d99ea86941c7ce35c14a312c2
```

请求方式： GET

参数	说明
app_id	运营给的参数
login_key	登录返回的
openid	登录返回的
sso_name	第三方平台返回的用户名
sso_type	要绑定的平台， google 或 facebook
sign	请求链接加上 appkey的参数 再md5的值

返回事例：

```
{"code":0, "msg": "success"}
```

参数	说明
code	0 代表成功

第三方账号解绑(和后端交互)

请求事例：

```
https://sdk.hahplatform.com/sdk/sso/bind?action=unbind&app_id=6007e26eb6ee064428e16d96&login_key=131e82317d50eb55cd35be161d5ea53d&open_id=1549&openid=112304055357683206858&sso_name=&sso_type=google&sign=61e6556112acedfd5d6e10d00e14a5a8
```

请求方式： GET

--	--

参数	说明
action	unbind
app_id	运营给的参数
login_key	登录返回的
openid	登录返回的
sso_name	第三方平台返回的用户名
sso_type	要绑定的平台，google 或 facebook
sign	请求链接加上 appkey的参数 再md5的值

返回事例：

```
{"code":0,"msg":"解绑成功"}
```

参数	说明
code	0 代表成功

请求语言数据(和后端交互)

请求链接

```
http://10.19.200.42:9009/static/sdk1_0_0/locales/en-US.json
```

请求方式：GET ， en-US 代表请求的是英文数据，zh-TW 代表繁体 ， 其他的请自行和运营商量

返回：json字符串，自行解析所需数据

支付(和后端交互)

创建订单

请求事例：

```
role_name=WalkerPag&encode=0&login_key=95f5d5d214cf66e661851ec8c55b2576&
```

pay_channel_id=5458c012e138231304af4216&productid=beauty_00099_1&money=0.99&open_id=1609&role_id=2100297844&callback=202101271450147228852699&game_amount=9&serverid=2003&app_id=6007e26eb6ee064428e16d96

请求方式：POST 表单提交

参数	说明	备注
callback	公共订单号	
open_id	登录返回的数据	
login_key	登录返回的数据	
serverid	游戏区服	
productid	内购id	
role_name	玩家角色名称	
role_id	玩家角色id	
pay_channel_id	登录返回的数据	
game_amount	游戏传过来的金额和比率之积	
money	游戏传过来的金额	
app_id	运营给的参数	
encode	是否加密，0 不加，1 加	

返回事例：

```
{"code":0,"msg":{"youai_star": 99, "pay_param": {"next_url": "", "query_id": "60110d281a65d320ffdc4f95"}, "amount": 0.99, "game_unit": "个", "kefu_info": ["Facebook: @Leyou"], "game_unit_name": "金币", "trade_id": "60110d281a65d320ffdc4f94", "game_amount": 9}}
```

参数	说明
code	0 代表成功
query_id	支付校验订单号

确认订单

请求事例：

```
{
  "user_info": "1609|2100297844|2003",
  "unique_id": "60110d281a65d320ffdc4f95",
  "open_id": "1609",
  "signeddata": "{\n\"orderId\":\n\"GPA.3382-9280-8672-73304\n\", \n\"packageName\":\n\"com.xml.palace.beauty\n\", \n\"productId\":\n\"beauty_00099_1\n\", \n\"purchaseTime\":1611730221231, \n\"purchaseState\":0, \n\"purchaseToken\":\n\"hmenkbjjojiickomocmmdfbf.A0-J10zuUszV_sTqUl_xjFInC0Zjhh0l70UnKjcwW7mGgdTgYc1SGW_LZnb1TSDGRwAdacba81J2jVX7Eks2gUoC8MTIwTf16w\n\", \n\"obfuscatedAccountId\":\n\"1609|2100297844|2003\n\", \n\"obfuscatedProfileId\":\n\"60110d281a65d320ffdc4f95\n\", \n\"acknowledged\":false\n}",
  "signature": "A3Hr2ptyCj88E0o1f1F4LDRaI9PZ43vx0UDJLSn3aKcUtcWc4Tcx8Xx\\3VYZYZkiiogmcxyt0Wqic7dEQ5LAuYYt5\\juckUh3nXZ31u0RIP2f8voj0lZZXMNGaw00kNhZh7kCMJxljoG9G1PzSEScxjnsGB6TFfucKvxBBP8+rwVe5Z14L6xHrAQK1LYcbdeD1GRXl60Ghk06TE2\\W3I\\gd0kd7l3GfrtmZ04qdrKZZpyy9c1bHnnj7NfktTQg5nsd6QxnfftYZ+OR0PjzVw1Hqvl8DlGx5DdH8vHeSwcXismyKLJitcAUD20nWnzMZjwNHZyExVz5V6ztBaLrMuJQ==",
  "query_id": "60110d281a65d320ffdc4f95",
  "product_id": "beauty_00099_1",
  "app_id": "6007e26eb6ee064428e16d96"
}
```

请求方式： POST json字符串

参数	说明	备注
open_id	登录返回的数据	
signeddata	公共订单号	google内购传
signature	公共订单号	google内购传
query_id	创建订单返回	
product_id	内购id	
app_id	运营给的参数	

返回事例：

success

参数	说明
success	支付成功

打开个人中心(和前端交互)

请求链接：`https://sdk.hahplatform.com/static/sdk1_0_0/client.html?action=user&app_id=6007e26eb6ee064428e16d96&app_key=4c3ce5d31f10cfb3922c652063f519ed&open_id=1605&login_key=ffc8ee95ab85efe4fc4d60a74535cd3c&uuid=&idfv=&device_info=10&ver=20210127×tamp=1611716804191&lng=en-US`

参数	说明
action	动作，打开个人中心就为 user
app_id	运营给的参数
app_key	运营给的参数
open_id	登录返回的
login_key	登录返回的
timestamp	时间戳
lng	当前用户的语言

前端返回

说明：一般情况返回的数据格式为：`sdk://exit?action=xxx&type=xxx&其他参数=xxx`，当 host 为 `exit` 时，需要你处理结果后关闭页面，通过 `action` 确定要处理的是何种行为的数据，通过 `type` 确定当 `action` 一样 时的 子行为 数据（`type`不一定有）

事件返回

返回的链接：`sdk://event?action=event&eventName=jump&msg=guest&eventTime=0`

参数	说明
action	event
eventName	事件名
msg	事件描述
eventTime	消耗时间

登录返回

返回的链接: `sdk://exit?action=login&birthday=&float_button=1&kefu=&login_count=1&login_key=32263921f4de55eb8399794a2fb84a59¬ify_url=https%3A%2F%2Fsdk.hahplatform.com%2Fservice%2Fconfirm%2Fgoogle%2F5458c012e138231304af4216&open_id=1607&pay_channel_id=5458c012e138231304af4216&share_pic=&share_str=×tamp=1611718040&token=5fe0a1b6b3d4b408f9c4661289c4f120`

参数	说明
action	返回的事件名
birthday	生日日期
float_button	悬浮球
kefu	客服
login_count	登录次数
login_key	login_key
notify_url	确认订单地址
open_id	open_id
pay_channel_id	pay_channel_id
share_pic	facebook分享的图片链接
share_str	facebook分享的图片描述
timestamp	时间戳
token	token

切换账号返回

返回的链接: `sdk://exit?action=user&type=changeAccount`

参数	说明
action	user
type	changeAccount

页面的x关闭页面返回

返回的链接: `sdk://exit?action=finish`

参数	说明
action	finish

选择语言返回

返回的链接: `sdk://exit?action=user&type=changeLng&language=en-US`

注意: 语言切换成功后需重新打开个人中心页面

参数	说明
action	user
type	changeLng
language	当前用户选择时哪种语言

游客账号登录点击去绑定返回

返回的链接: `sdk://exit?action=user&type=HighAccount`

注意: 收到通知后需重新打开个人中心页面并在链接后台加上参数 `open=HighAccount`

参数	说明
action	user
type	HighAccount

前端通过js调用原生接口

方法名	说明
facebookLogin	facebook登录
googleLogin	google登录
twitterLogin	推特登录

bindFacebook	绑定facebook
unbindFacebook	解绑facebook
bindGoogle	绑定google
unbindGoogle	解绑谷歌
bindTwitter	绑定推特
unbindTwitter	解绑推特

选择大区接口说明

区域匹配接口

1) 请求地址

https://sdk.hahplatform.com/sdk/location?app_id=5ffd3ed2b6ee064e678d4a67

2) 调用方式：HTTP get

3) 接口描述：

- 服务端根据玩家IP匹配返回对应玩家区域的区服数据

4) 请求参数：

GET参数：

字段名称	字段说明	类型	必填	备注
app_id	sdk的appId参数	string	Y	-

5) 请求返回结果：

```
{
  "code": 0,
  "msg": {
    "list": [
      {
        "url": "http://rose-static-test.hnwanqi.cn",
        "key": "AS",
        "title": "国服内测(亚洲)"
      },
      {
        "url": "http://rose-static-test.hnwanqi.cn",
        "key": "AS1",
        "title": "国服内测(亚洲)1"
      },
      {
        "url": "http://rose-static-test.hnwanqi.cn",
        "key": "AS2",
```

```

        "title": "国服内测(亚洲)2"
    }
},
"match": {
    "url": "http://rose-static-test.hnwanqi.cn",
    "key": "AS",
    "title": "国服内测(亚洲)"
},
"kefu_info": [
    "客服服务中心: "
]
}
}

```

6) 请求返回结果参数说明:

字段名称	字段说明	类型	必填	备注
code	0:成功 其他失败	string	Y	-
msg	消息内容	string	Y	-
list	配置的大区列表数据	string	Y	-
url	大区地址	string	Y	-
key	大区key	string	Y	-
title	大区名称	string	Y	-
match	对应IP匹配到的区服数据	string	Y	-
kefu_info	客服信息	string	Y	-

上报事件到公共后台

上报事件对应的key

事件名	后台key
初始化请求耗时事件	init
打开登录页面	login_open
公共登录验证耗时事件	login_check
绑定接口耗时事件	third_bind
解绑定耗时	third_unbind
创建订单耗时	create_order_sdk(sdk内部),create_order_public(接入层)
确认订单耗时	confirm_order
用户中心页面	user_center_open
第三方账号生成乐游	bind_candy
点击facebook登录	facebook_login_action
点击google登录	google_login_action
facebook登录返回	facebook_login_return
google登录返回	google_login_return
获取区域地址	game_area
点击游客登录	guest_login_action