用户 SDK 接入文档

更新时间：2015.06.23 版本号 2.3.0

爱游戏网游用户体系 SDK 接入事项备注说明：

2、 建议双方建立用户 SDK 接入技术讨论组（包括双方技术人员）。

建议也可通过“爱游戏 SDK 技术咨询”QQ 群（138036160）随时沟通。

2、接入中涉及到的游戏产品接入参数可在爱游戏 OPEN 开放平台

(http://open.play.cn/dev/)获取，可由 CP 商务或爱游戏人员查询。

示例：

游戏名称：XXXX

游戏 ID：5010282

Client_Id：87415562

client_secret：4429730d429741b68ebc3a6752c0cf9d

3、用户 SDK 下载地址，请见 FAQ.1

4、接入联调前，需要开通用户 SDK 的接入权限，详见 FAQ.2

目录

1. 修订记录

版本号	作者	更新时间	更新内容
1.0.0	朱俊宇	2014.5.6	文档初始化
1.0.1	朱俊宇	2014.6.2	整合前后端文档
1.0.2	朱俊宇	2014.6.11	添加获取 token 接口
1.1.0	韩玮	2014.9.4	更新文档格式 更新 onCreate 接口为 start 接口 废除 startUserSdk 接口统一为 start 接口 毋须在 AndroidManifest.xml 中配置参数 简化客户端 SDK 的接口 本文档对应的 20140904 以后发布的 SDK
1.1.1	韩玮 杜彬	2014.9.16	完善文档格式，增加部分名词解释和 FAQ
1.1.2	许成	2014.9.30	完善 OPEN 接入流程说明，包括用户 SDK 下载、用户 SDK 权益申请；
1.1.3	杜彬 姚远	2014.10.13	添加 FAQ.9/FAQ.10/FAQ.11/FAQ.12； 提供打开登录过程异常信息的日志接口
1.2.0	杜彬 姚远	2014.10.23	调整登陆流程，修改客户端、服务器端接入方式。兼容原登陆流程，但不再推荐使用，对应用户 SDK1.7 改动内容。 登录过程的异常信息以错误码的形式返回到对应的接口中
2.0.0	姚远	2014.12.23	完善接口说明，对应用户 SDK2.0.0 改动内容
2.0.1	姚远	2014.12.25	对应用户 SDK2.0.1 改动内容
2.0.2	姚远	2015.01.12	对应用户 SDK2.0.2 改动内容
2.0.3	姚远	2015.01.15	更新日志统计，对应用户 SDK2.0.3 改动内容
2.1.0	姚远、杜彬	2015.01.23	修改客户端接入方式，修改登陆触发方式，登陆接口不变，更新服务端接口说明，对应用户 SDK2.1.0 改动内容
2.1.0	杜彬	2015.02.04	优化数字签名描述；添加服务器端请求示例
2.2.0	姚远	2015.03.04	增加对消费卷的支持，对应用户 SDK2.2.0 改动内容
2.2.1	姚远	2015.04.21	完善接入文档说明，增强嵌入方式的兼容性，添加 FAQ.12，对应用户 SDK2.2.1 改动内容
2.2.1	杜彬	2015.06.03	添加数字签名相关服务端错误码

2.3.0	姚远	2015. 06.23	对应用户 SDK 2.3.0 改动内容，完善文档

术语与缩写解释

术语	定义	备注
ClientId/client_id	应用编号	
ClientSecret/client_secret	应用密码	为 32 为 md5 码，用于应用和爱游戏服务器端通信时，对通信内容进行数字签名
CP	应用提供商	
token	用户给应用的授权令牌信息	包含授权令牌，刷新令牌，有效期，是否失效状态，用户编号，应用编号等信息，爱游戏服务器颁发的令牌有效期为两个月
access_token	授权令牌	为 32 位 md5 码
code	授权码，用于兑换授权令牌	为 32 位 md5 码，有效期十分钟

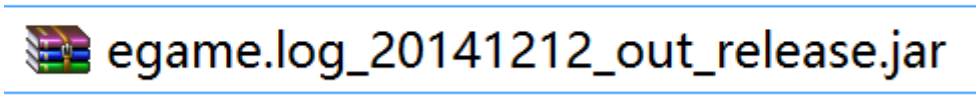
2． 客户端接入方式

3.1 使用方法

3.1.1 文件清单

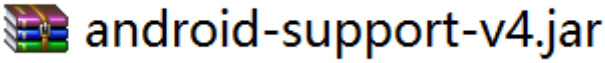
SDK 本体：（注：文件大小可能略有不同）

统计支持包：



V4 支持包：

(注：默认 v4 包在 Android 源码中，用户 sdk 暂不提供，请自行添加)



3.1.2 如何在项目中配置

第一步：导入 SDK

将 SDK 作为 library 项目引用到自己的项目中

将统计包和 v4 包添加到自己的项目中

第二步：在自己项目的 AndroidManifest.xml 中加入如下权限：

```
<!-- 需要添加的权限  wei.han BEGIN -->
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.GET_TASKS" />
<!-- 需要添加的权限  wei.han END -->
```

第三步：添加 activity 声明 ：

```
<activity
    android:name="egame.terminal.usersdk.customview.FloatViewActivity"
    android:screenOrientation="portrait" >
</activity>
<activity
     android:name="egame.terminal.usersdk.customview.floatview.EgameBrowserActivity
     "
     android:screenOrientation="portrait" >
   </activity>
```

注意：此处 2 个的 activity 的 LauncherMode 不能设置成 singleTop 或者 singleInstance ， 关 于 activity 主 题 ， 请 设 置 成 android:Theme.Light 或其子主题。

第 四 步 ： 你 需 要 写 一 个 基 类 Activity 继 承 egame.terminal.usersdk.EgameUserActivity

须重写 initLogin 方法

第五步：在 initLogin 方法中调用登陆接口

示例：

必要参数 *CLIENT_ID*：

```
/**
 * 游戏的clientId
 */
private static final int CLIENT_ID = 84106431;
```

调用方法：

```
public void initLogin(){
   EgameUser.start(this, CLIENT_ID, new CallBackListener() {
         public void onSuccess(String code) {
             Toast.makeText(MainActivity.this, "登录成功，授权码是"+code,
                    Toast.LENGTH_SHORT).show();
         }
         public void onFailed(int code) {
           Toast.makeText(MainActivity.this, "登录失败，错误码是"+code,
                    Toast.LENGTH_SHORT).show();
         }
         public void onCancel() {
             if (!EgameUser.getLoginWindowState()) {
               Toast.makeText(MainActivity.this, "登陆窗消失", 0).show();
              } else {
                 Toast.makeText(MainActivity.this, "取消登录", 0).show();
              }
         }
      });
}
```

第六步：然后让你的其他 activity 继承自这个基类

第七步：登录成功后根据授权码在有效时间内请自行换取 token

另附：若项目中有 activity 已无法继承其他类，请在每个 activity 的 onResume 和 onPause 方法中分别调用 EgameUser.onResume(Activity activity)和 EgameUser.onPause(Activity activity)

```
@Override
public void onResume(){
    EgameUser.onResume(MainActivity.this);
    super.onResume();
}

@Override
public void onPause(){
     EgameUser.onPause(MainActivity.this);
     super.onPause();
}
```

在自身的 **onActivityResult** 方法中捕获 **resultCode**,并做如下处理

```
@Override
public void onActivityResult(int requestCod,int resultCode,Intent data){
      if(resultCode==ErrorCode.TOKEN_INVALID){
         //在此处重新调用登陆接口
          EgameUser.start(MainActivity.this,clientId,callBacklistener);
      }
      super. onActivityResult (requestCod,resultCode,data);
```

详细请看 **demo**

3.2 API 文档

3.2.1 接口概要

EgameUser 接口概要

方法	
限定符和类型	**方法和说明**
static void	**start**(Activity activity, int clientId, **CallBackListener** callbackListener) 触发用户 SDK 的登录流程

- **start**

```
public static void start(Activity activity,
        int clientId,
        CallBackListener callbackListener)
               throws java.lang.IllegalArgumentException
```

触发用户 SDK 的登录流程

参数:

activity - 当前页面的 Activity 对象

clientId - 游戏的 clientId

callbackListener - 回调监听

抛出:

java.lang.IllegalArgumentException - 传递非法参数时会抛出参数异常,如 activity 为 null, clientId 为负数

CallBackListener 接口概要

方法	
限定符和类型	**方法和说明**
public void	**onSuccess(String code)** 登陆成功后回调的接口，返回授权码，需要根据授权码自行换取 token
限定符和类型	**方法和说明**
public void	**onFailed(int code)** 登陆流程中发生的异常时，回调的接口，返回错误码
限定符和类型	**方法和说明**
public void	**onCancle()** 中断登陆过程时回调的接口，主要是在切换用户时触发

4. 服务端接入方式

4.1 数字签名(重要)

4.1.1 签名策略

爱游戏开放平台的数字签名级别分三种，安全级别由低到高依次为：**无签名；基本签名；业务签名**。每个接口都会根据数据的安全要求，对数字签名级别进行限制，在服务请求过程中，如果签名的级别与要求的不一致，或者签名信息与请求参数不匹配，则会被认定为非法请求并阻止调用。所以请您在开发过程中，确保每个接口的签名都与接口所要求的级别、算法一致。

另外，我们为您提供了用于服务器交互的 jar 包及 demo，包含发起 http/https 请求、数字签名等功能，如果您的服务器采用 JAVA 语言开发，可以直接引用，无需开发，如果您的服务器采用其他语言，请参考文档和 demo 自主实现相关功能。

4.1.2 签名算法

基本签名

基本签名只对固定的基本参数进行签名，是 CP 身份的证明。

必须进行签名的字段：统一为 client_id、sign_method、version、timestamp、client_secret

具体算法如下：

在调用接口时，如果需要使用基本签名，需要在 https 请求参数中加入如下参数，且参数名必须保持一致：client_id、sign_method、version、timestamp、sign_sort、signature。

client_id: 应用向开放平台注册时分配的应用编号

sign_method: 数字签名的编码算法，如 MD5，HmacSha1，HmacMD5 等，SDK 目前版本暂时只支持 MD5 算法

version: 所使用的 SDK 版本号，例如 v2.1.0

timestamp: 请求发起时的时间戳，如：1385345938378

sign_sort: 用于指定请求参数在进行数字签名时的排列顺序。把 client_id、sign_method、version、timestamp、client_secret 的参数名，按照任意顺序排列，然后拼接起来，参数名之间以“&”符号连接。例如：sign_sort=client_id&sign_method&version&timestamp&client_secret。另外请在签名后，请求发送前对此字段进行 urlencode

signature: 数字签名。计算步骤如下：

1. 把 client_id、sign_method、version、timestamp、client_secret 参数的值，按照 sign_sort 中指定的顺序排列，并拼接成字符串，参数之间无需分隔符。
2. 采用 jdk 的类 java.security.MessageDigest，对步骤 1 生成的字符串进行 md5 计算，生成 byte 数组。
3. 将步骤 2 生成的 byte 数组转换成 16 进制字符串，此字符串即为数字签名。计算过程中字符集采用 utf-8，16 进制字符串转换算法必须与爱游戏提供的 ByteFormat.java 中算法保持一致。

示例：某应用（client=1001；client_secret=a1b2c3），请求某接口
业务参数：token= aaaaaaaa,
签名级别：基本签名

则加入签名信息后最终请求参数如下：
token=aaaaaaaa
client_id=1001
sign_method=MD5
version=v2.1.0
timestamp= 1423018253631
sign_sort= timestamp&sign_method&client_secret&client_id&version
signature= bytesToHexString
(MD5.encode("1001v2.1.0MD5a1b2c31385345938378"))

业务签名

业务签名需要对固定的基本参数和接口业务参数进行签名。不仅能验证 CP 身份，而且能够提高数据安全性，防止在请求过程中业务参数被第三方篡改。所以采用业务签名的接口，会在接口清单中明确约定需要签名的字段。

必须进行签名的字段：具体接口指定。

具体算法如下：
在调用接口时，如果需要使用业务签名，需要在 https 请求参数中加入如下参数，且参数名必须保持一致：client_id、sign_method、version、timestamp、sign_sort、signature

client_id: 应用向开放平台注册时分配的应用编号

sign_method: 数字签名的编码算法，如 MD5，HmacSha1，HmacMD5 等，SDK 目前版本暂时只支持 MD5 算法

version: 所使用的 SDK 版本号，如 v2.1.0

timestamp: 请求发起时的时间戳，如：1385345938378

sign_sort：用于指定请求参数在进行数字签名时的排列顺序。把接口要求进行签名的参数，按照任意顺序排列，然后拼接起来，参数名之间以**“&”**符号连接。例如：某接口约定使用业务签名，签名字段为 client_id、sign_method、version、timestamp、client_secret、username、password，则可以是 sign_sort=password&username&client_secret&timestamp&version&sign_method&client_id。另外请在签名后，请求发送前对此字段进行 urlencode

signature：数字签名。计算步骤如下：

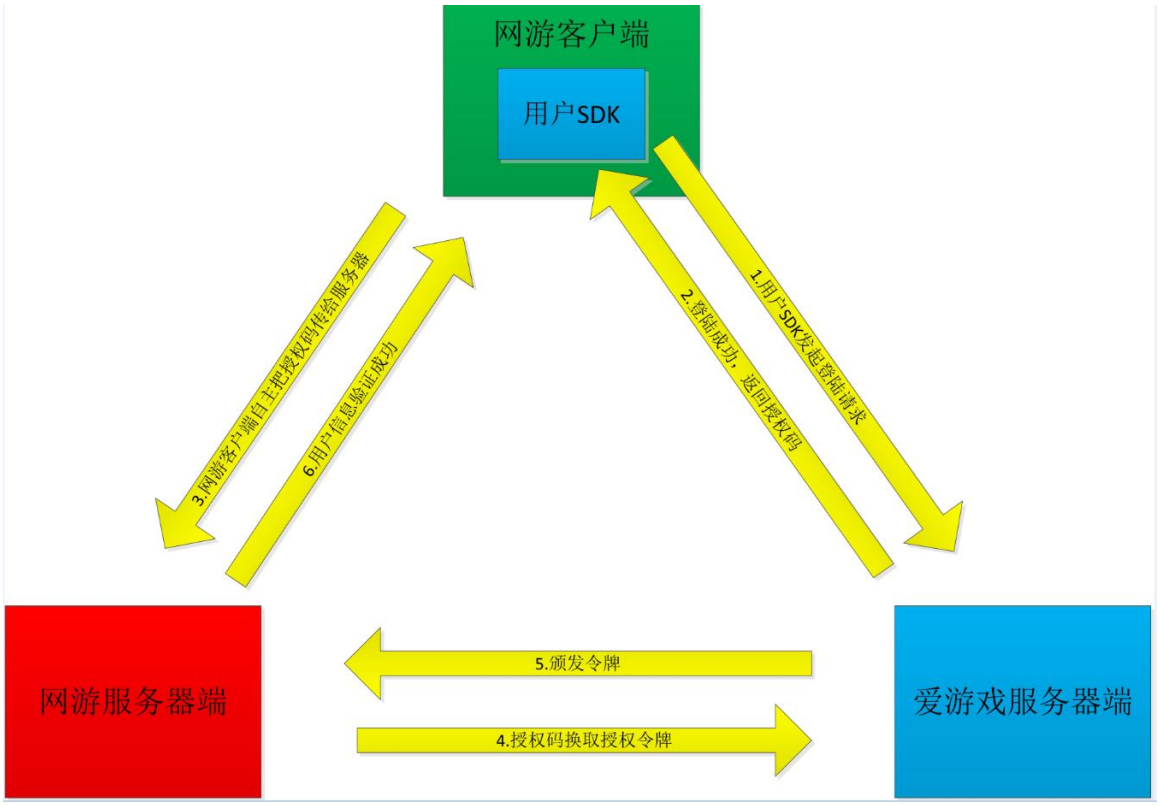
1. 把所有需要进行签名的参数的值，按照 sign_sort 中指定的排列，并拼接成字符串，参数之间无需分隔符。
2. 采用 jdk 的类 java.security.MessageDigest，对步骤 1 生成的字符串进行 md5 计算，生成 byte 数组。
3. 将步骤 2 生成的 byte 数组转换成 16 进制字符串，此字符串即为数字签名。计算过程中字符集采用 utf-8，16 进制字符串转换算法必须与爱游戏提供的 ByteFormat.java 中算法保持一致。

示例：某应用（client=12；client_secret=cs），请求某接口
业务参数：username=open； password=123； imsi=189；
签名级别：业务签名
签名字段：client_id、sign_method、version、timestamp、client_secret、username、password、imsi

则加入签名信息后最终请求参数如下：

```
username=open
password=123
imsi=189
client_id=12
sign_method=MD5
version=v2.1.0
timestamp=1385345938378
sign_sort=client_id&sign_method&version&timestamp&client_secret&username&password&imsi
signature= bytesToHexString
(MD5.encode(“12MD5v2.1.01385345938378csopen123189”))
```

4.2 登陆流程



登陆流程：

1. 用户触发登陆功能，用户 SDK 判断用户是否满足自动登陆条件，如果不满足，引导用户进入登陆界面。
2. 用户身份通过验证后，爱游戏服务端会生成待登陆应用的授权码，并把授权码返回给客户端，提示登陆成功。
3. 网游客户端需要自主把授权码传给网游服务器，用于网游服务器换取授权令牌，确认用户信息。
4. CP 服务器端调用爱游戏服务器端“授权码兑换 token”接口（详见 4.3.1）
5. 爱游戏服务器端校验授权码通过后，生成授权令牌并返回给 CP 服务器端。
6. CP 服务器端收到令牌后，确认用户信息，之后便可以进行自己的业务逻辑（如绑定账号等）

4.3 服务端接口 API(接口清单)

4.3.1 授权码兑换令牌

服务地址：https://open.play.cn/oauth/token

签名级别：2

签名策略：基本签名

提交方式：POST

功能说明：获取访令牌

请求头参数：Content-Type=application/x-www-form-urlencoded

请求体参数：

编号	参数名	必填	类型/长度	备注
1	client_id	yes	String	对应于应用注册时分配的 client_id
2	client_secret	yes	String	对应于应用注册时分配的 client_secret
3	code	yes	String	用于兑换令牌的授权码
4	grant_type	yes	String	此值固定为 authorization_code
5	scope	no	String	访问请求的作用域
6	state	no	String	维护请求和响应的状态，传入值与返回值保持一致

返回结果

编号	参数名	类型	备注
1	access_token	string	访问令牌
2	token_type	string	令牌类型，目前只有“Bearer”
3	refresh_token	string	刷新令牌
4	expires_in	string	访问令牌过期时间
5	re_expires_in	string	刷新令牌过期时间
6	scope	string	授权范围
7	user_id	string	爱游戏用户 Id

可能的错误码：

错误码	错误描述
invalid_request	非法请求
unauthorized_client	未授权的应用
access_denied	请求被拒绝
unsupported_response_type	非法的 response_type
invalid_scope	非法的权限范围
invalid_grant	非法的 grant_type 类型
expired_token	令牌过期
insufficient_scope	超出授权范围

scope limits	不合法授权范围
app status is invalid	应用状态不可用
appkey is overdue	appkey 超出有效时间
app is overdue	应用超出有效时间
authorize_code is overdue	authorize_code 超出有效时间
access_token is overdue	access_token 超出有效时间
refresh_token is overdue	refresh_token 超出有效时间
request method must be get	请求方式必须为 get
request method must be post	请求方式必须为 post
client_id is empty	client_id 为空
response_type is empty	response_type 为空
redirect_uri is empty	redirect_uri 为空
grant type is empty	grant_type 为空
authorize code is empty	授权码为空
client_secret is invalidate	应用的 client_secret 无效
app business exception occured	CP 私有业务处理异常
wrong version	SDK 版本号为空或错误
wrong sign_sort	签名顺序为空或错误
wrong clientId	Client_Id 为空或错误
wrong sign_method	签名算法为空或错误
wrong timestamp	时间戳为空或错误
failed to check signature	校验数字签名失败(爱游戏服务端错误)
wrong signature	签名错误(与所传签名参数不匹配)
url_write(get /oauth/token) is out of the config.	请求方式错误，请使用 post 请求

请求示例：

```
https://open.play.cn/oauth/token?client_id=111111&client_secret=12341234473a415f9bd3a3e76609c15f&code=b57fe9b13125c13dc359864953759a95&grant_type=authorization_code&sign_method=MD5&timestamp=1423019541343&sign_sort=timestamp%26sign_method%26client_secret%26client_id%26version&signature=173E82FEE2CFD50A63DA0AD58C78B910&version=2.1.0
```

结果示例

成功响应：

```
{
    "scope": "all",
    "re_expires_in": 15552000,
    "user_id": 956877,
    "token_type": "Bearer",
    "expires_in": 5184000,
    "refresh_token": "2c639e8c1cbfeee5fb07e968163d0343",
    "access_token": "2cd0a6f9c8ce81ada335f1989413ca08"
}
```

失败响应：

```
{
    "error_uri":null,
    "error":"invalid_request",
    "state":null,
    "error_description":"Missing grant_type parameter value"
}
```

4.3.2 访问令牌鉴权

服务地址：https://open.play.cn/oauth/token/validator

签名级别：2

签名策略：基本签名

提交方式：POST

功能说明：访问令牌鉴权（此接口主要用于鉴定用户身份，当应用客户端需要操作应用的数据时，如游戏道具，游戏金币等，应用服务器可以调用此接口鉴定用户身份，从而确定用户是否有权限操作相关数据）

请求体参数：

编号	参数名	必填	类型/长度	备注
1	client_id	yes	String	对应于应用注册时返回的 client_id
2	access_token	yes	String	访问令牌

返回结果：

编号	参数名	类型	备注
1	code	string	返回码,0 表示成功
2	text	string	返回码描述信息
3	ext	string	业务数据
4	ext->user_id	string	爱游戏用户编码
5	ext->access_token	string	访问令牌
6	ext->expires_in	string	令牌有效时间

可能的错误码：

-5,-260,-261,-262,-263,-264,-265

请求示例：

http://open.play.cn/oauth/token/validator?client_id=111111&access_token=3e8dee90ae7884cff0dfb1fa3818fa42&sign_method=MD5&timestamp=1423019541343&sign_sort=timestamp%26sign_method%26client_secret%26client_id%26version&signature=173E82FEE2CFD50A63DA0AD58C78B910&version=2.1.0

结果示例：

成功响应：


```
{
    "code": 0,
    "ext": {
        "access_tpken": "6e87e2dd56dd92a9e5d0322ec9ecbaca",
        "expires_in": 4945807,
        "user_id": 317506
    },
    "text": "success"
}
```


失败响应：


```
{
"code":-261,
"text":"ErrorCode:-261 / Message:access_Token is invalid",
"ext":null
}
```

I 附录

服务端错误代码

错误代码	错误信息
-5	请求参数异常
-260	oauth 访问令牌 accessToken 过期
-261	oauth 访问令牌 accessToken 无效
-262	oauth 访问令牌 accessToken 失效
-263	应用授权过期
-264	应用访问受限
-265	应用不存在
-267	数字签名或签名参数错误

签名级别代码

签名级别代码	含义
1	无签名
2	基本签名
3	业务签名

客户端错误代码

错误代码	错误信息
-10000	服务器异常
-10001	令牌无效或过期
-10002	重定向失败
-10003	无法解析数据
-10004	无法获取令牌
-10005	手机所在运营商未知
-10006	无法获取短信中心的服务号码
-10007	无法注册手机号码
-10008	短信上行参数非法
-10009	手机号码已存在
-10010	手动注册网络异常
-10011	手机网络异常
-10012	用户名密码错误
-10013	手机号码非法

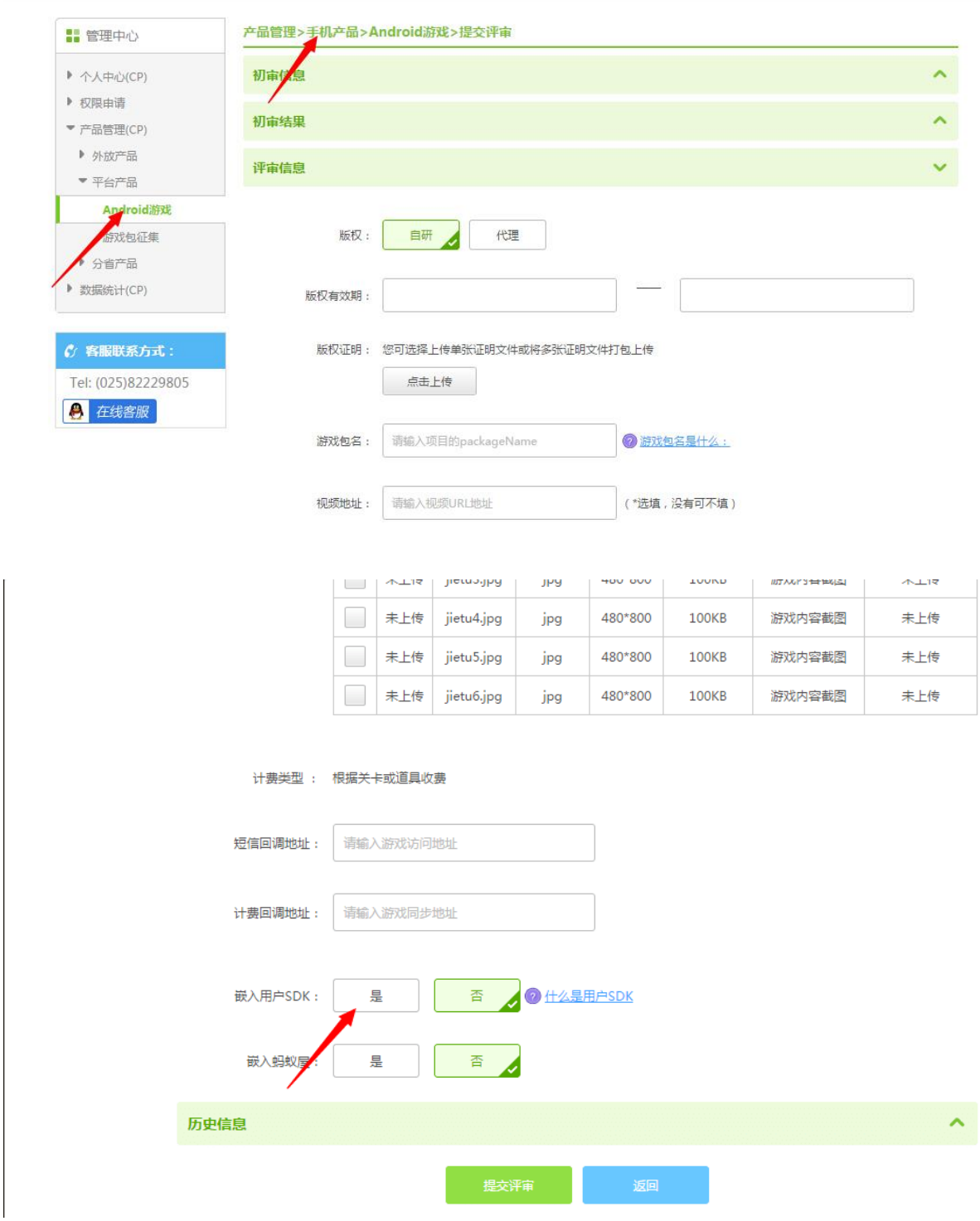
II 常见问题 FAQ

1．用户 SDK 包在哪里下载？

登陆 OPEN 后台，进入管理中心，按下图指示下载用户 SDK 包；

2．如何申请用户 SDK 的接入权限？

在爱游戏的 OPEN 后台管理中心，创建游戏并提交初审通过后，在申报产品信息页面，勾选嵌入用户 SDK 即可。按以下几张图指示申请：



3．移动和联通的手机号可以注册吗？
可以。

4．登录时可以切换账号吗？
可以。在登录过程中，屏幕的右上角显示切换账号的按钮，点击即可进入到手动登录界面。

5．为什么屏幕始终 s 显示在登录中的状态中？

请检查 AndroidManifest.xml 中相关权限是否完整，特别是下面的权限是否已经声明：

```
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
```

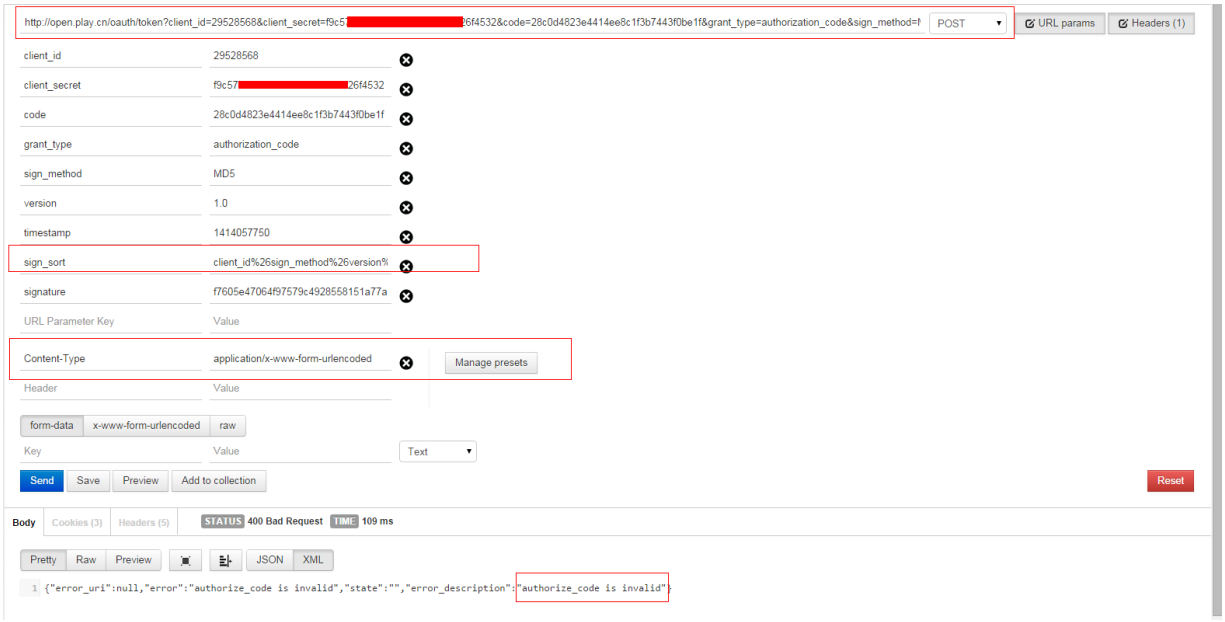
6. 为什么调用爱游戏授权码兑换令牌接口时，虽然做了签名，但是仍跑不通？

首先确定请求头里包含编码格式信息：

Content-Type = application/x-www-form-urlencoded

然后确定请求体里的参数已经做过 UrlEcode 处理，即参数值里包含的“&”等特殊字符已经转义。

如果按照以上方法仍无法跑通，请尝试把请求参数直接拼到 url 后面，而不是放到请求体里，如下图所示：



如果报错信息与图中一致，恭喜你，接口已经调通了，换一个正确的 code 即可。

7. 为什么做签名时，要把签名转换成 16 进制字符串？

因为 MD5 加密后的签名为 byte 数组，传输时需转换成对应字符串

8. 为什么我的网游服务端配置没有问题但是却拿不到客户端请求的数据？

因为虽然客户端重定向到网游服务端的提交方式是 post，但是请求参数是拼接在 url 地址上的

9. 为什么我的注册了号码却无法登陆？并且显示的错误码是-10001 或者-10010？

请联系贵方商务，去爱游戏平台上查询接入的游戏是否已经开通了用户体系

10. 为什么我的登录出现了图片按钮适配问题？

请确保调用 SDK 的 activity 使用的主题为 android:Theme 及其子主题，并且不要继承 ActionBarActivity

11. 如何验证我的数字签名 MD5 算法是否与爱游戏一致？
可用您的 MD5 算法对如下串进行加密：
1421986517277MD5abc2d0c88900417c88e4f6d678c6d04f111111112.1.0
正确结果：4066426956C959ECBC84597786428313

12. 为什么我调用登陆接口却无法启动或者报 can't create handle thread that Has not called Looper.prepare()?
请在主线程中调用登陆接口，可能在某些游戏引擎下，使用conext.runOnUiThread()方法也是不行的，这时候，请通过以下方式调用

```
new Handler(Looper.getMainLooper()).post(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub
               //在这里调登陆
            }
        });
```

13. 为什么我的手机上看不到切换用户的按钮和消息推送的提示？
如果是小米手机，请在应用设置里找到对应的应用，开启显示悬浮窗权限就可以了；
在游戏正式上线之前，消息推送会在登陆成功后提示网络异常，这是正常提示。