

# 4399登录SDK接入文档

## 目录

4399登录SDK接入文档

目录

修改记录

导入库与资源

接入登录SDK

- 1. 初始化
- 2. 注册
- 3. 登录
- 4. 记录网页登录历史用户名
- 5. 调用结果回调

接入服务端接口

- 1. 服务端接口与参数
- 2. 服务端成功响应
- 3. 服务端失败响应

附录一、登录SDK授权步骤

附录二、登录接口结果码对照表

登录授权成功

登录授权失败

附录三、自定义SDK界面风格

## 修改记录

版本号	时间	修改人	修改内容
1.2.0.0	2016-01-06	Y.J.Zhou	发布新登录sdk版本
1.2.0.1	2016-01-15	Y.J.Zhou	为配置增加redirect_url接口
1.2.0.2	2016-02-29	Y.J.Zhou	解决用户强行关闭网络权限导致的崩溃问题

版本号	时间	修改人	修改内容
1.2.0.3	2016-03-04	Y.J.Zhou	关闭未释放资源
1.2.0.4	2016-03-08	Y.J.Zhou	加入网页登陆的日志收集

## 导入库与资源

1. 将res下的文件复制到工程相应的资源目录中去。
2. 将 m4399LoginSDK.jar 文件复制到工程的libs目录下
3. 在工程AndroidManifest.xml 中注册 OpeHostActivity :

```
1. <activity
2.     android:name="cn.m4399.operate.controller.OpeHostActivity"
3.     android:configChanges="orientation|screenSize|keyboardHidden"
4.     android:launchMode="singleTask" />
```

4. 在工程AndroidManifest.xml 中加入权限

```
1. < uses-permission android:name="android.permission.INTERNET" />
2. < uses-permission
   android:name="android.permission.READ_PHONE_STATE" />
3. < uses-permission
   android:name="android.permission.ACCESS_NETWORK_STATE" />
```

## 接入登录SDK

### 1. 初始化

建议在 **Appliction** 类的 onCreate 方法中进行初始化。

```
1. public class AppDemo extends Application{
2.
3.     @Override
4.     public void onCreate() {
5.         super.onCreate();
6.         // 获取sdk操作类
7.         OperateCenter opeCenter = OperateCenter.getInstance();
```

```

8.         // sdk配置信息类
9.         OperateConfig opeConfig = new OperateConfig.Builder()
10.            // 登陆界面横竖屏配置（ 当使用游戏盒授权时，登陆强制为竖屏 ）（
    必填 ）
11.            .setOrientation(SCREEN_ORIENTATION)
12.            // app在用户中心分配的 client_id （ 必填 ） 第一次接入的 APP
    可自行向 4399用户中心（厦门） 申请
13.            .setClientID("testNet")
14.            // app在用户中心分配的 client_id 所对应的 redirect_url（ 必
    填 ）
15.            .setRedirectUrl("testRedirectUrl")
16.            // app在游戏盒分配的 id （非必填 默认自动填充为clientid ）
17.            .setGameID("testNet")
18.            // 是否全屏显示登录界面 （ 选填 ）
19.            .setFullScreen(false)
20.            .build();
21.
22.        // 二选一 进行SDK初始化
23.        // 进行sdk初始化（接口一）
24.        mOpeCenter.init(getApplicationContext(), opeConfig);
25.
26.        // 进行sdk初始化（接口二）（有效防止进程被杀而导致的游戏盒无法授权登录 推
    荐使用）
27.        mOpeCenter.init(getApplicationContext(), opeConfig, new Operat
    eCenter.ValidateListener() {
28.            @Override
29.            public void onValidateFinished(SDKResult result) {
30.                Toast.makeText(getApplicationContext(), result.toSt
    ring(), Toast.LENGTH_LONG).show();
31.            }
32.        });
33.    }
34. }

```

```

1. class SDKResult {
2.     // 登录接口结果码，参照（附录二）来确定是哪种登录方式
3.     public void getResultCode();
4.     // 登录接口结果
5.     public void getResultMsg();
6.     // Web 授权登录模式 AuthCode
7.     public void getAuthCode();
8.     // 游戏盒授权登录模式 refresh_token
9.     public void getRefreshToken();
10.    // 游戏盒授权登录模式 uid

```

```
11.         public String geUID();
12.     }
```

## 2. 注册

```
1.     // 注册接口一经调用，无论原先是否已经登录一律清除原有登录信息，重新进行注册并且登录
    步骤。
2.     mOpeCenter.register(this, new OperateCenter.ValidateListener() {
3.         @Override
4.         public void onValidateFinished(SDKResult result) {
5.             Toast.makeText(getApplicationContext(), result.toSt
ring(), Toast.LENGTH_LONG).show();
6.         }
7.     });
```

## 3. 登录

```
1.     // 注册接口一经调用，无论原先是否已经登录一律清除原有登录信息，重新进行注册并且登录
    步骤。
2.     mOpeCenter.login(this, new OperateCenter.ValidateListener() {
3.         @Override
4.         public void onValidateFinished(SDKResult result) {
5.             Toast.makeText(getApplicationContext(), result.toSt
ring(), Toast.LENGTH_LONG).show();
6.         }
7.     });
```

## 4. 记录网页登录历史用户名

```
1.         mOpeCenter.recordAccountName(String accountName);
```

## 5. 调用结果回调

```
1.     public interface ValidateListener {
2.         /**
3.          * @param result    用户验证结果
4.          */
5.         public void onValidateFinished(SDKResult result);
6.     }
```

## 接入服务端接口

## 1. 服务端接口与参数

web授权登录

**GET** <https://ptlogin.4399.com/oauth2/token.do>

参数名	内容
grant_type=AUTHORIZATION_CODE	( 固定字段 )
client_id	用户中心分配的 client_id
redirect_uri	用户中心分配的 redirect_uri
client_secret	用户中心分配的 secret
code	Web登录获取到的 <b>AuthCode</b>

游戏盒授权登录

**POST** <https://ptlogin.4399.com/oauth2/token.do>

参数名	内容
grant_type=REFRESH_TOKEN	( 固定字段 )
client_id	用户中心分配的 client_id
client_secret	用户中心分配的 secret
redirect_uri	用户中心分配的 redirect_uri
refresh_token	游戏盒授权登录获取到的 <b>refresh_token</b>

## 2. 服务端成功响应

web授权登录

参数名	内容
access_token	access_token
expired_at	accesstoken过期时间点
expires_in	accesstoken有效期
refresh_token	refresh_token
scope	权限范围，现在都是basic
uid	用户身份识别码
username	用户名
display_name	用于显示的名字
ext_nick	第三方账号昵称
bound_phone	绑定手机号
account_type	账号类型，4399，qq，weixin，weibo
ext_token_expired_at	第三方账号token过期时间点

## 游戏盒授权登录

参数名	内容
uid	用户身份识别码
username	用户名
scope	权限范围，现在都是basic
nick	第三方账号昵称
expires_in	accesstoken有效期
access_token	access_token

### 3. 服务端失败响应

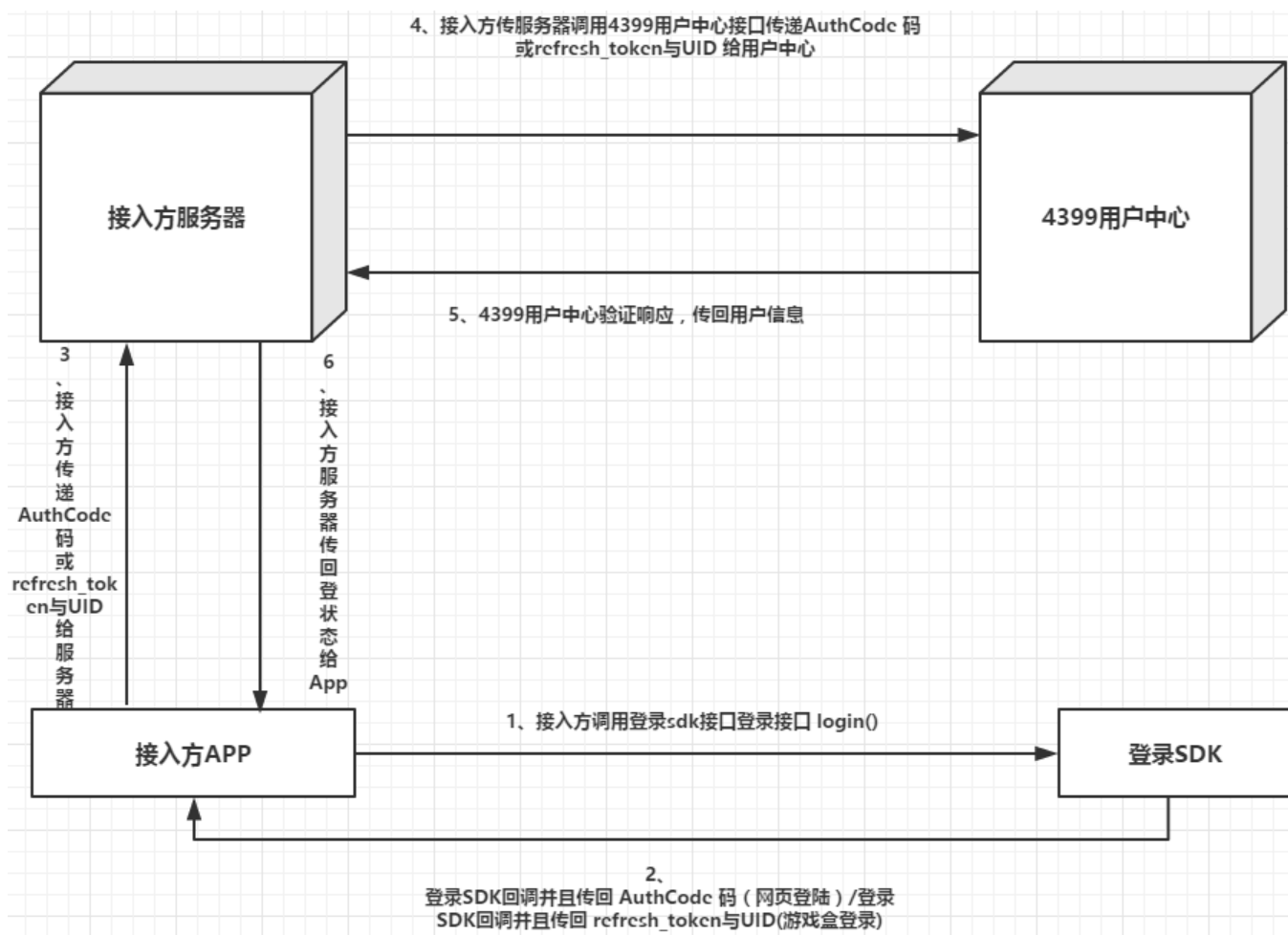
web授权登录

```
1. {"error" : "invalid_request" , "error_description" : "Code unauthorized" }
```

游戏盒授权登录

```
1. {"error" : "invalid_request" , "error_description" : "Invalid refresh token" }
```

## 附录一、登录SDK授权步骤



## 附录二、登录接口结果码对照表

登录授权成功

Result_code	Result_msg
0x000	Web 登录返回 <b>AuthCode</b> 码
0x001	Web 注册返回 <b>AuthCode</b> 码
0x002	游戏盒授权返回 <b>Refresh_Code</b> 与 <b>UID</b>

登录授权失败

Result_code	Result_msg
0x101	网页登录网络异常
0x102	网页注册网络异常
0x103	游戏盒授权无结果
0x104	用户取消登录

附录三、自定义SDK界面风格





登录SDK的头部一共有三张 9-patch 背景图片：

图**红色**部分背景图为：

导航背景图：m4399loginsdk\_9patch\_title\_bg.9.png

图**蓝色**部分按钮背景图：

按钮正常态：m4399loginsdk\_9patch\_title\_btn\_normal.9.png

按钮选定态：m4399loginsdk\_9patch\_title\_btn\_active.9.png

接入时可以对这些图片进行替换以达到更改背景的目的。