**JAVA游戏客户端接入**

* **概述**
* **此文档为Android原生游戏接入**
* **文档对应SDK版本为3.1.3**
* **详细接入步骤请参考Demo**
* **接入准备**

**2.1、工程导入**

**将SDK项目中java客户端目录下assets目录下文件复制到游戏目录assets目录下，将java客户端目录下libs下jar文件复制到游戏目录libs目录下，将java客户端目录下res中资源文件复制到游戏目录res中**

**2.2、配置AndroidManifest.xml文件**

**添加权限：**

**<uses-permission android:name="android.permission.INTERNET"/>**

**<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>**

**<activity**

**android:name="com.funcell.platform.android.game.proxy.FuncellSDKLoginActivity"android:configChanges="orientation|keyboardHidden|screenSize"android:screenOrientation="behind"**

**android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />**

**<activity android:name="com.funcell.platform.android.game.proxy.pay.funcell.FunSdkUiActivity"**

**android:configChanges="orientation|keyboardHidden|screenSize"**

**android:screenOrientation="behind"**

**android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />**

**<meta-data android:name="FUNCELL\_CHANNEL\_TEST" android:value="@string/FUNCELL\_CHANNEL\_TEST" />**

* **接入接口**

**3.1、初始化接口（必接）**

**初始化方法，在游戏的第一个Activity的onCreat()方法中调用**

**FuncellGameSdkProxy.getInstance().Init(this, new IFuncellInitCallBack(){**

**@Override**

**public void onInitFailure(String paramString) {**

**// TODO Auto-generated method stub**

**Log.e(TAG,"-------onInitFailure");**

**}**

**@Override**

**public void onInitSuccess() {**

**// TODO Auto-generated method stub**

**Log.e(TAG,"------onInitSuccess");**

**}**

**}, new IFuncellSessionCallBack(){**

**@Override**

**public void onLoginSuccess(FuncellSession session) {//用户登录成功**

**// TODO Auto-generated method stub**

**Log.e(TAG,"-------onLoginSuccess");**

**Log.e(TAG,"-------session"+session.getmToken());//token信息**

**//session.getmUserID() 用户ID**

**}**

**@Override**

**public void onLoginCancel() {//用户取消登录**

**// TODO Auto-generated method stub**

**Log.e(TAG,"------onLoginCancel");**

**}**

**@Override**

**public void onLoginFailed(String paramString) {//登录失败**

**// TODO Auto-generated method stub**

**Log.e(TAG,"paramString:"+paramString);**

**Log.e(TAG,"------onLoginFailed");**

**}**

**@Override**

**public void onLogout() {//帐号登出**

**// TODO Auto-generated method stub**

**Log.e(TAG,"------onLogout");**

**/\*\***

**\* 游戏需要返回登录界面**

**\*/**

**}**

**@Override**

**public void onSwitchAccount(FuncellSession session) {**

**// TODO Auto-generated method stub**

**/\*\***

**\* 切换帐号回调(使用回调信息进行登录验证)**

**\* 部分渠道在切换帐号回调中，可能是以登录成功回调来 回调给SDK**

**\* 因此当接到该回调信息时候，需要先通知游戏退回登录主界面进行当前角色切换**

**\*/**

**Log.e(TAG,"------onSwitchAccount");**

**}**

**}, new IFuncellPayCallBack(){**

**@Override**

**public void onFail(String paramString) {//支付失败**

**// TODO Auto-generated method stub**

**Log.e(TAG,"IFuncellPayCallBack onFail");**

**}**

**@Override**

**public void onCancel(String paramString) {//支付取消**

**// TODO Auto-generated method stub**

**Log.e(TAG,"IFuncellPayCallBack onCancel");**

**}**

**@Override**

**public void onSuccess(String cpOrder, String sdkOrder,**

**String extrasParams) {**

**// TODO Auto-generated method stub**

**/\*\***

**\* 渠道充值成功，不代表游戏服务器到账，请知晓**

**\* 返回的参数，如果游戏接入者有自己的逻辑，可以不使用以上参数**

**\*/**

**Log.e(TAG,"IFuncellPayCallBack onSuccess");**

**Log.e(TAG,"cpOrder:"+cpOrder); //游戏方的订单，**

**Log.e(TAG,"sdkOrder:"+sdkOrder);//平台订单号**

**Log.e(TAG,"extrasParams:"+extrasParams);//透传参数**

**}**

**});**

**}**

**3.1、生命周期接口（必接）**

**在游戏的主Activity的生命周期中调用SDK生命周期接口：**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**FuncellGameSdkProxy.getInstance().onCreate(this);**

**}**

**public void onStop() {**

**super.onStop();**

**FuncellGameSdkProxy.getInstance().onStop(this);**

**}**

**public void onDestroy() {**

**super.onDestroy();**

**FuncellGameSdkProxy.getInstance().onDestroy(this);**

**}**

**public void onResume() {**

**super.onResume();**

**FuncellGameSdkProxy.getInstance().onResume(this);**

**}**

**public void onPause() {**

**super.onPause();**

**FuncellGameSdkProxy.getInstance().onPause(this);**

**}**

**public void onStart() {**

**super.onStart();**

**FuncellGameSdkProxy.getInstance().onStart(this);**

**}**

**public void onRestart() {**

**super.onRestart();**

**FuncellGameSdkProxy.getInstance().onRestart(this);**

**}**

**protected void onActivityResult(int requestCode, int resultCode, Intent data) {**

**super.onActivityResult(requestCode, resultCode, data);**

**FuncellGameSdkProxy.getInstance().onActivityResult(this, requestCode, resultCode,data);**

**}**

**public void onNewIntent(Intent intent){**

**super.onNewIntent(intent);**

**FuncellGameSdkProxy.getInstance().onNewIntent(intent);**

**}**

**public void onRequestPermissionsResult(int requestCode,String[] permissions, int[] grantResults) {**

**// TODO Auto-generated method stub**

**FuncellGameSdkProxy.getInstance().onRequestPermissionsResult(requestCode, permissions, grantResults);}**

**3.2、登录接口（必接,主线程调用）**

**FuncellGameSdkProxy.getInstance().Login(MainActivity.this);**

**3.3、登出接口（必接）**

**FuncellGameSdkProxy.getInstance().Logout(MainActivity.this);**

**3.4、支付接口（必接,主线程调用）**

**FuncellRoleInfo roleInfo = new FuncellRoleInfo(); //角色信息**

**roleInfo.setGameGoldBalance("888");//角色剩余元宝数**

**roleInfo.setGameUnionName("游戏工会");//游戏工会**

**roleInfo.setRoleId(“123”);//角色ID**

**roleInfo.setRoleLevel("2");//角色等级**

**roleInfo.setRoleName("test");//角色名**

**roleInfo.setServerId("998");//区服ID**

**roleInfo.setServerName("天下");//区服名**

**roleInfo.setVipLevel("3");//角色VIP等级**

**FuncellPayParams PayParams = new FuncellPayParams();**

**PayParams.setmExtrasParams("test"); //透传参数，支付成功后，返回给客户端的数据**

**PayParams.setmItemAmount("10"); //商品价格**

**PayParams.setmItemCount(10);//钻石数量 ,10个元宝**

**PayParams.setmItemDescription("商品描述");**

**PayParams.setmItemId("100002"); //商品ID**

**PayParams.setmItemName("10元宝"); //商品名**

**PayParams.setmItemType("元宝"); //商品类型**

**PayParams.setmCurrency("RMB");//货币设置**

**PayParams.setmOrderId("游戏接入方订单号"); //游戏方订单号，若无则不传该参数或者不设置，该字段为支付成功后，返回的数据**

**PayParams.setmRoleInfo(roleInfo);**

**FuncellGameSdkProxy.getInstance().Pay(MainActivity.this, PayParams);**

**3.5、数据统计接口（必接）**

**创建角色事件：创建角色后调用(无法设置的字段设置成空字符串)**

**ParamsContainer pc = new ParamsContainer();**

**pc.putString("usermark", mSession.getmUserID());//用户ID，登录成功后返回的数据**

**pc.putString("role\_id", "123");//角色ID**

**pc.putString("serverno", "998");//区服ID**

**pc.putString("server\_name", "天下");//区服名**

**pc.putString("role\_name", "test");//角色名**

**pc.putString("role\_gamegold\_balance", "游戏玩家金币余额"); //游戏金币，并非充值后的钻石数量**

**pc.putString("role\_gameunion\_name", "公会名字");**

**pc.putString("role\_vip\_level", "1");//VIP等级**

**pc.putString("role\_level", "1"); //角色等级**

**pc.putString("role\_creat\_time", "服务器时间戳");//角色创建时间**

**pc.putString("role\_upgrade\_time", "服务器时间戳"); //角色升级时间**

**pc.putString("role\_recharge\_balance", "游戏玩家充值游戏币余额"); //充值后的游戏币余额(钻石、元宝等余额)**

**FuncellGameSdkProxy.getInstance().setDatas(MainActivity.this,FuncellDataTypes.DATA\_CREATE\_ROLE, pc);**

**登录事件：登录游戏并选择了服务器后调用**

**ParamsContainer pc = new ParamsContainer();**

**pc.putString("usermark", mSession.getmUserID());//用户ID，登录成功后返回的数据**

**pc.putString("serverno", "998");//区服ID**

**FuncellGameSdkProxy.getInstance().setDatas(MainActivity.this, FuncellDataTypes.DATA\_LOGIN, pc);**

**角色升级事件：角色升级后调用(无法设置的字段设置成空字符串)**

**ParamsContainer pc = new ParamsContainer();**

**pc.putString("usermark", mSession.getmUserID());//用户ID，登录成功后返回的数据**

**pc.putString("serverno", "998");//区服ID**

**pc.putString("role\_level", "999");//等级**

**pc.putString("role\_id", "123");//角色ID**

**pc.putString("role\_name", "test");//角色名**

**pc.putString("server\_name","天下");//区服名**

**pc.putString("role\_gamegold\_balance", "游戏玩家金币余额"); //游戏金币，并非充值后的钻石数量**

**pc.putString("role\_gameunion\_name", "公会名字");**

**pc.putString("role\_vip\_level", "1");//VIP等级**

**pc.putString("role\_creat\_time", "服务器时间戳");//角色创建时间**

**pc.putString("role\_upgrade\_time", "服务器时间戳"); //角色升级时间**

**pc.putString("role\_recharge\_balance", "游戏玩家充值游戏币余额"); //充值后的游戏币余额(钻石、元宝等余额)**

**FuncellGameSdkProxy.getInstance().setDatas(MainActivity.this,FuncellDataTypes.DATA\_ROLE\_LEVELUP, pc);**

**角色服务器事件：可在进入游戏主场景时调用(无法设置的字段设置成空字符串)**

**ParamsContainer pc = new ParamsContainer();**

**pc.putString("usermark", mSession.getmUserID());//用户ID，登录成功后返回的数据**

**pc.putString("role\_id", "123");//角色ID**

**pc. putString ("role\_level", "999"); //等级**

**pc.putString("role\_name", "test");//角色名**

**pc.putString("role\_gameunion\_name", "工会");//工会名**

**pc.putString("role\_vip\_level", "VIP等级");//VIP等级**

**pc.putString("serverno", "998");//区服ID**

**pc.putString("server\_name", "天下");//区服名**

**pc.putString("role\_gamegold\_balance", "游戏玩家金币余额"); //游戏金币，并非充值后的钻石数量**

**pc.putString("role\_creat\_time", "服务器时间戳");//角色创建时间**

**pc.putString("role\_upgrade\_time", "服务器时间戳"); //角色升级时间**

**pc.putString("role\_recharge\_balance", "游戏玩家充值游戏币余额"); //充值后的游戏币余额(钻石、元宝等余额)**

**FuncellGameSdkProxy.getInstance().setDatas(MainActivity.this,FuncellDataTypes.DATA\_SERVER\_ROLE\_INFO, pc);**

**3.6、游戏退出接口（必接,主线程调用）**

**查询渠道是否有退出界面接口：**

**FuncellGameSdkProxy.getInstance().GetExitUI(this);**

**0代表渠道无退出界面，其他值代表渠道有退出界面**

**SDK退出接口：**

**FuncellGameSdkProxy.getInstance().Exit(this, new IFuncellExitCallBack(){**

**@Override**

**public void onChannelExit() {**

**// TODO Auto-generated method stub**

**Log.e(TAG, "-------onChannelExit");**

**Toast.makeText(MainActivity.this, "渠道界面退出",Toast.LENGTH\_LONG).show();**

**finish();**

**android.os.Process.killProcess(android.os.Process.myPid());**

**}**

**@Override**

**public void onGameExit() {**

**// TODO Auto-generated method stub**

**Log.e(TAG, "-------onGameExit");**

**AlertDialog.Builder builder = new Builder(**

**MainActivity.this);**

**builder.setTitle("游戏退出界面");**

**builder.setPositiveButton("退出",**

**new DialogInterface.OnClickListener() {**

**@Override**

**public void onClick(DialogInterface dialog,**

**int which) {**

**dialog.dismiss();**

**/\*\*\*\* 退出逻辑需确保能够完全销毁游戏 \*\*\*\*/**

**finish();**

**android.os.Process**

**.killProcess(android.os.Process**

**.myPid());**

**/\*\*\*\* 退出逻辑请根据游戏实际情况，勿照搬Demo \*\*\*\*/**

**}**

**});**

**builder.setNegativeButton("取消",**

**new DialogInterface.OnClickListener() {**

**@Override**

**public void onClick(DialogInterface dialog,**

**int which) {**

**dialog.dismiss();**

**}**

**});**

**dialog = builder.create();**

**dialog.show();**

**}**

**});**

**以下接口为选接：**

**3.7、扩展方法接口（选接）**

**FuncellGameSdkProxy.getInstance().callFunction(MainActivity.this, "functionTest");**

**参数详解：**

|  |  |
| --- | --- |
| **MainActivity.this** | **上下文** |
| **functionTest** | **调用方法名** |

**3.8、获取登录状态（选接）**

**FuncellGameSdkProxy.getInstance().GetLoginFlag();**

**参数详解：**

**返回值：boolen类型**

**3.9、获取EVE数据（选接）**

**FuncellGameSdkProxy.getInstance().GetEveData();**

**参数详解：**

**返回值：json数据类型**

**3.10、获取公告列表（选接）**

**FuncellGameSdkProxy.getInstance().GetNoticeList(MainActivity.this, new IPlatformNoticeListCallBack(){}, "type", "server\_id");**

**参数详解：**

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必传** | **参数说明** |
| **MainActivity.this** | **Y** | **上下文** |
| **IPlatformNoticeListCallBack** | **Y** | **回调接口** |
| **type** | **Y** | **类型(如：登录公告、活动公告、退出公告类型)** |
| **server\_id** | **N** | **缺省参数，服务器ID** |

**公告类型(type字段详解)：**

|  |  |
| --- | --- |
| **general** | **游戏普通公告** |
| **login** | **游戏登录公告** |
| **activity** | **游戏活动公告** |
| **update** | **游戏更新公告** |
| **sdk\_pop** | **SDK登陆弹窗** |
| **sdk\_exit** | **SDK退出弹窗** |

**注：可以一次性获取多种类型的公告，以逗号”,”分割**

**3.11、获取服务器列表（选接）**

**FuncellGameSdkProxy.getInstance().GetServerList(this, new IPlatformServerListCallBack(){}, "white\_key");**

**参数详解：**

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必传** | **参数说明** |
| **this** | **Y** | **上下文** |
| **IPlatformServerListCallBack** | **Y** | **回调接口** |
| **white\_key** | **N** | **缺省参数，白名单Key** |

**3.12、获取商品列表（选接）**

**FuncellGameSdkProxy.getInstance().GetPayList(this, true, "category", new IFuncellPayListCallBack(){});**

**参数详解：**

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必传** | **参数说明** |
| **this** | **Y** | **上下文** |
| **true** | **Y** | **是否刷新数据的标志** |
| **category** | **Y** | **请求商品列表 类型** |
| **IFuncellPayListCallBack** | **Y** | **回调接口** |

**3.13、获取平台参数（选接）**

**FuncellGameSdkProxy.getInstance().GetPlatformParams(this, PlatformParamsType.GameID);**

**参数详解：**

|  |  |  |
| --- | --- | --- |
| **参数** | **是否必传** | **参数说明** |
| **this** | **Y** | **上下文** |
| **PlatformParamsType** | **Y** | **获取平台参数类型** |

**PlatformParamsType类型解释**

|  |  |
| --- | --- |
| **GameID** | **平台对应的游戏ID** |
| **PlatformID** | **平台对应的渠道ID** |
| **PlatformType** | **平台对应的渠道类型(如，UC，360)** |
| **AppVersion** | **平台对应的游戏版本号** |
| **ClientId** | **平台ClientId** |

**3.14、设置初始化参数（选接）**

**FuncellGameSdkProxy.getInstance().setConfigParams(this,jsonObject.toString());**

**以下参数均为json数据中对应Key值（String类型）**

**若游戏设置了以下字段，默认使用游戏设置参数**

|  |  |
| --- | --- |
| **参数(Key)** | **参数说明** |
| **datacenter** | **平台分配（数据中心）** |
| **gameId** | **平台分配（游戏ID）** |
| **platformId** | **平台分配（渠道ID）** |
| **eve** | **平台分配（eve地址）** |
| **ChannelVersionCode** | **平台分配（渠道版本）** |
| **report** | **平台分配（BI上报地址）** |
| **platformType** | **平台分配（渠道类型）** |
| **area** | **平台分配（地区）** |
| **resVersion** | **打包工具填写（资源版本）** |
| **appVersion** | **打包工具填写（应用版本）** |

* **混淆**

**若需要混淆java代码，请勿将SDK代码混淆,可在proguard中进行如下配置：**

-dontwarn com.funcell.platform.android.game.proxy.\*\*

-keep class com.funcell.platform.android.\*\*{\*;}   
-keep interface com.funcell.platform.android.\*\*{\*;}   
-keep enum com.funcell.platform.android.\*\*{\*;}