U.LU SDK 开发者文档

**Android 版本 v1.2.8**

作者：颜鹏

审阅：钮小明

批准：钮小明

上海游陆信息科技有限公司

2019年10月11日

Changelog：

|  |  |
| --- | --- |
| V1.1.0 | 初版定稿 |
| V1.1.1 | 增加登出接口，接入firebase分析和推送 |
| V1.1.2 | 增加英语国际化 |
| V1.1.3 | 增加进入用户中心，查询商品信息，上传角色名，打开客服 |
| V1.2.0 | 优化对话框独立性，排除cp外部干扰 |
| V1.2.2 | 加入支持竖屏 |
| V1.2.3 | 支付、上传角色传入角色id |
| V1.2.8 | SDK初始化、支付优化，接入文档更新 |

**一、导入相关的库及参数配置**

复制ulusdk.x.x.x.arr文件到项目app/libs目录中

将 minSdkVersion 版本设置为22，targetSdkVersion不小于27

修改结果如图所示



应用级 build.gradle

repositories {  
 flatDir {  
 dirs 'libs'  
 }  
 }  
 productFlavors {  
 }  
}

apply plugin: 'com.google.gms.google-services'

dependencies {  
 implementation(name: 'ulusdk1.1.1', ext: 'aar')  
 implementation 'com.google.android.gms:play-services-auth:16.0.1'

implementation 'com.facebook.android:facebook-login:[4,5)'

implementation 'com.google.firebase:firebase-messaging:17.6.0'  
implementation 'android.arch.work:work-runtime:1.0.1

implementation 'com.google.firebase:firebase-core:16.0.7'

'

项目级build.gradle

buildscript {  
 repositories {  
 maven { url 'https://maven.google.com' }  
 mavenLocal()  
 jcenter()  
   
 }  
 dependencies {  
 classpath 'com.android.tools.build:gradle:3.3.2'  
 classpath 'com.google.gms:google-services:4.0.1'  
  
 // NOTE: Do not place your application dependencies here; they belong  
 // in the individual module build.gradle files  
 }  
}  
  
allprojects {  
 repositories {  
 maven { url 'https://maven.google.com' }  
 mavenLocal()  
 jcenter()  
 }  
}

### 二、配置AndroidManifest.xml

#### 2.1.添加权限

uses-permission android:name="android.permission.INTERNET"/>  
<uses-permission android:name="com.android.vending.BILLING" />  
<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>  
<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"/>

<uses-permission android:name="android.permission.WAKE\_LOCK"/>

#### ****2.2.添加组件****

<meta-data  
 android:name="channel\_id"  
 android:value="渠道id"  
 />

<activity  
 android:name="com.ulusdk.WebviewActivity"  
 android:screenOrientation="portrait"

></activity>

设置页面恒竖屏

android:screenOrientation="portrait"

// portrait为竖屏， sensorLandscape为横屏  
<activity  
 android:name="com.ulusdk.UserCenterActivity"  
 android:configChanges="orientation|keyboardHidden|screenSize"  
 android:screenOrientation="sensorLandscape"></activity>

<activity

注意！！！这个页面不要设置页面方向，Java代码动态设置了  
 android:name="com.ulusdk.ULUMainActivity"  
 android:configChanges="orientation|keyboardHidden|screenSize"  
 android:theme="@style/ulu\_activity"  
  
 ></activity>

<activity  
 android:name="com.ulusdk.BindMailActivity"  
android:configChanges="orientation|keyboardHidden|screenSize"  
 android:screenOrientation="sensorLandscape"></activity>  
<activity  
 android:name="com.ulusdk.ChangePasswordActivity"  
android:configChanges="orientation|keyboardHidden|screenSize"  
 android:screenOrientation="sensorLandscape"></activity>

<service  
 android:name="com.ulusdk.googlepush.ULUFirebaseMessagingService"  
 android:exported="false">  
 <intent-filter>  
 <action android:name="com.google.firebase.MESSAGING\_EVENT" />  
 </intent-filter>  
</service>

<!-- facebook SDK-->  
<meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/facebook\_app\_id"/>  
 <activity android:name="com.facebook.FacebookActivity"  
 android:configChanges= "keyboard|keyboardHidden|screenLayout|screenSize|orientation"  
 android:label="@string/app\_name" />  
 <activity android:name="com.facebook.CustomTabActivity" android:exported="true">  
 <intent-filter> <action android:name="android.intent.action.VIEW" />  
 <category android:name="android.intent.category.DEFAULT" />  
 <category android:name="android.intent.category.BROWSABLE" />  
 <data android:scheme="@string/fb\_login\_protocol\_scheme" />  
 </intent-filter> </activity>

2.3.设置XML参数

Strings.xml中添加游戏从游陆获取到的facebook\_app\_id参数和fb\_login\_protocol\_scheme参数 ，从游陆获取google\_pay\_key，google-service.json

<!--facebook sdk-->  
<string name="facebook\_app\_id">1070581506477121</string>  
<string name="fb\_login\_protocol\_scheme">fb1070581506477121</string>  
  
<string name="google\_pay\_key">MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5gxYJh+OKDOHshIS8VXy/ikqMu8ooqSZhUpRBruo90IvgSLzRzgMB3k1qWh1/d/zMzWoX6kaDNLbUQgWupWNMdRNn4ANFqXfIjJhII3c/wCa/UYrZHbrGudnbGcTi1QHHOYxy7TuPNiFqmsmJ20k7hfj4IsKXPQ3DZrClO/cc67jxz6C+JF+huHouYxGhDtreJl438dvTKmzIBDz2/EIGs2GgfDMxYX7JhSf9nrNgl0XtRjhZuxWXLvZgx359Ql/VtJvBA70QIEgV7lxviNJm/ode4xaAJbO+fN25ebBZP/+IjtfrjfUtHdBtf3N8yHaGvVPobn5lwXr/bgEp4OczwIDAQAB</string>

### 三、SDK接入流程

#### 3.1.获取Sdk实例方法

ULUSDKManager.*getInstance*()

**3.2.添加生命周期接口（必接）**

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 ULUSDKManager.*getInstance*().ULUOnCreat(this);

}

@Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 ULUSDKManager.*getInstance*().ULUOnActivityResult(requestCode, resultCode, data);  
  
 }  
  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 ULUSDKManager.*getInstance*().onRequestPermissionsResult(requestCode,permissions,grantResults);  
 }

@Override  
protected void attachBaseContext(Context newBase) {  
 super.attachBaseContext(newBase);  
 ULUSDKManager.*getInstance*().ULUattachBaseContext(newBase);  
}

@Override  
 protected void onStart() {  
 super.onStart();  
 ULUSDKManager.*getInstance*().ULUOnStart(this);  
 }  
  
 @Override  
 protected void onStop() {  
 super.onStop();  
 ULUSDKManager.*getInstance*().ULUOnStop(this);  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
 ULUSDKManager.*getInstance*().ULUOnPause(this);  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
 ULUSDKManager.*getInstance*().ULUOnResume(this);  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 ULUSDKManager.*getInstance*().ULUOnDestory(this);  
 }

#### ****3.3.Sdk初始化接口（必接）****

String gameId="1";  
ULUSDKManager.*getInstance*().init(this, gameId, new ULUInitListener() {  
 @Override  
 public void onInitSuccess() {  
 Toast.*makeText*(MainActivity.this,"初始化成功",Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onInitFail(String errorMsg) {  
 Toast.*makeText*(MainActivity.this,"初始化失败---"+errorMsg,Toast.*LENGTH\_SHORT*).show();  
  
 }  
});

#### ****3.4.Sdk登录接口（必接）****

登录失败返回“-1” 表示用户取消登录

ULUSDKManager.*getInstance*().login(new ULULoginListener() {  
 @Override  
 public void onLoginSuccess(User user) {  
 String uid=user.getUid();  
 String token=user.getToken();  
 Toast.*makeText*(MainActivity.this,"登录成功",Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onLoginFail(String erroMsg) {  
 Toast.*makeText*(MainActivity.this,"登录失败"+erroMsg,Toast.*LENGTH\_SHORT*).show();  
 }  
 });

#### ****3.5.Sdk支付接口（必接）****

支付失败返回“-2”表示googleplay没登陆

ULUOrder uluOrder=new ULUOrder();  
uluOrder.setProductId(SKU\_GAS);   
  
ULURole uluRole=new ULURole();  
uluRole.setRoleName("RoleName");  
uluRole.setServerName("ServerName");  
uluRole.setServerId("ServerId");

uluRole.setRoleId("RoleId");

ULUSDKManager.*getInstance*().uluGooglePay

(uluOrder,uluRole,new ULUPayListenter() {  
 @Override  
 public void onPaySuccess(String orderId, String extrasParams) {  
 LogUtil.*getInstance*().i(this.getClass().getSimpleName(),"paySuccess");  
 Toast.*makeText*(MainActivity.this,"支付成功",Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onPayFail(String orderId, String errorMsg) {  
 //errorMsg LogUtil.*getInstance*().i(this.getClass().getSimpleName(),"onPayFail");  
 Toast.*makeText*(MainActivity.this,"支付失败"+errorMsg,Toast.*LENGTH\_SHORT*).show();  
 }  
});

#### ****3.7.进入用户中心（必接）****

ULUSDKManager.*getInstance*().setUluLogoutListener(new ULULogoutListener() {  
 @Override  
 public void onLogoutSuccess() {  
 Toast.*makeText*(MainActivity.this, "退出成功", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onLogoutFail(String error) {  
 Toast.*makeText*(MainActivity.this, "退出失败", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
ULUSDKManager.*getInstance*().enterUserCenter();

#### ****3.8.查询谷歌内购所有商品信息（选接）****

查询失败返回“-2”表示googleplay没登陆

ArrayList<String> skuList = new ArrayList<String> ();  
skuList.add(SKU\_GAS);  
ULUSDKManager.*getInstance*().queryProductInfo(skuList, new ULUQueryProductListener() {  
 @Override  
 public void onQuerySuccess(Map<String, SkuDetails> map) {  
 SkuDetails skuDetails=map.get(SKU\_GAS);  
 Toast.*makeText*(MainActivity.this,"查询成功 map.size()==="+map.size(),Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onQueryFail(String s) {

Toast.*makeText*(MainActivity.this,"查询失败 ",Toast.*LENGTH\_SHORT*).show();  
  
 }  
});

#### ****3.9.上传角色信息（选接）****

ULURole：角色信息

游戏在创建角色、登录后进入游戏和角色升级三个事件中调用此接口。

uluRole = new ULURole();  
uluRole.setRoleLevel("RoleLevel");  
uluRole.setRoleName("RoleName");  
uluRole.setServerName("ServerName");  
uluRole.setServerId("ServerId");  
uluRole.setVipLevel(1);   
ULUSDKManager.*getInstance*().updateRoleInfo(uluRole);

#### ****4.0.打开客服（必接）****

uluRole = new ULURole();  
uluRole.setRoleName("RoleName");  
uluRole.setServerName("ServerName");  
uluRole.setServerId("ServerId");  
uluRole.setVipLevel(1);  
uluRole.setRoleId("RoleId");

ULUSDKManager.*getInstance*().openCustomerService(uluRole);