U.LU SDK 开发者文档

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17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 |
| V2.0 | SDK重构 |

**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

apply plugin: 'com.android.application'  
apply plugin: 'com.google.gms.google-services'  
android {  
  
 def currentVersionCode = getVersionCode()  
 signingConfigs {  
 uluGame {  
 keyAlias 'ulugame'  
 keyPassword 'uluGame'  
 storeFile file('../uluGame.keystore')  
 storePassword 'uluGame'  
 }  
 }  
 compileSdkVersion 28  
 defaultConfig {  
 // applicationId "com.ulusdk.demo"  
 applicationId "com.ulusdk.ulu"  
 minSdkVersion 18  
 targetSdkVersion 27  
 multiDexEnabled true  
 versionCode currentVersionCode  
 versionName "2.5"  
 testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"  
 signingConfig signingConfigs.uluGame  
 }  
 buildTypes {  
 release {  
 minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 signingConfig signingConfigs.uluGame  
 }  
 debug {  
 signingConfig signingConfigs.uluGame  
 }  
 }  
 repositories {  
 flatDir {  
 dirs 'libs'  
 }  
 }  
 productFlavors {  
 }  
 android.applicationVariants.all { variant ->  
 variant.outputs.all {  
 def fileName = "ULUSDKDemo"+'.apk'  
 outputFileName = fileName  
 }  
 }  
}  
  
def getVersionCode() {  
 def versionCodeFile = file('version\_code.properties')  
 if (versionCodeFile.canRead()) {  
 Properties properties = new Properties()  
 properties.load(new FileInputStream(versionCodeFile))  
 def versionCode = properties['VERSION\_CODE'].toInteger()//读取version\_code.properties文件存放的版本号。  
 def runTasks = gradle.startParameter.taskNames  
 def task = 'assembleDebug'  
 if (task in runTasks) {  
 properties['VERSION\_CODE'] = (++versionCode).toString()  
 properties.store(versionCodeFile.newWriter(), null)  
 }  
  
 return versionCode  
 } else {  
 throw new GradleException("无法读取version\_code.properties文件！")  
 }  
}  
  
  
dependencies {  
 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'  
 implementation 'com.appsflyer:af-android-sdk:4.9.0'  
 implementation fileTree(include: ['\*.aar'], dir: 'libs')  
 implementation 'com.android.support:appcompat-v7:28.0.0'  
 implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
 implementation 'com.android.support:design:28.0.0'  
 testImplementation 'junit:junit:4.12'  
 androidTestImplementation 'com.android.support.test:runner:1.0.2'  
 androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
  
 implementation 'com.android.support:multidex:1.0.3'  
}

项目级build.gradle

// Top-level build file where you can add configuration options common to all sub-projects/modules.  
  
buildscript {  
 repositories {  
 google()  
 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 {  
 google()  
 jcenter()  
   
 }  
}  
  
task clean(type: Delete) {  
 delete rootProject.buildDir  
}

### 二、配置AndroidManifest.xml

#### 2.1.添加权限

<uses-permission android:name="android.permission.INTERNET"/>  
  
<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="0"  
 />

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

设置页面恒竖屏

android:screenOrientation="portrait"

// portrait为竖屏， sensorLandscape为横屏

注意！！！这个页面不要设置页面方向，Java代码动态设置了

<activity  
 android:name="com.ulusdk.ui.ULUMainActivity"  
 android:configChanges="orientation|keyboardHidden|screenSize"  
 android:theme="@style/ulu\_activity"  
></activity>

2.3.设置XML参数

Strings.xml中添加SDK环境配置，从游陆获取google-service.json

<!-- 1 ULUSDK沙盒环境 ，0 ULUSDK生产环境 -->  
<string name="ulu\_env">1</string>

### 三、SDK接入流程

#### 3.1.获取Sdk实例方法

ULUManager.*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*);  
 ULUManager.*getInstance*().ULUOnCreat(this);

}

@Override  
protected void attachBaseContext(Context newBase) {  
 super.attachBaseContext(newBase);  
 ULUManager.*getInstance*().ULUAttachBaseContext(newBase);  
}  
  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 ULUManager.*getInstance*().ULUOnActivityResult(requestCode, resultCode, data);  
  
}  
  
  
@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 ULUManager.*getInstance*().ULUOnRequestPermissionsResult(requestCode,permissions,grantResults);  
}  
  
@Override  
protected void onStart() {  
 super.onStart();  
 ULUManager.*getInstance*().ULUOnStart(this);  
}  
  
@Override  
protected void onStop() {  
 super.onStop();  
 ULUManager.*getInstance*().ULUOnStop(this);  
}  
  
@Override  
protected void onPause() {  
 super.onPause();  
 ULUManager.*getInstance*().ULUOnPause(this);  
}  
  
@Override  
protected void onResume() {  
 super.onResume();  
 ULUManager.*getInstance*().ULUOnResume(this);  
}  
  
@Override  
protected void onDestroy() {  
 super.onDestroy();  
 ULUManager.*getInstance*().ULUOnDestory(this);  
   
}

#### ****3.3.Sdk初始化接口（必接）****

String gameId="1";  
ULUManager.*getInstance*().init(this, gameId, new ULUListener() {  
 @Override  
 public void onInitSuccess() {  
 Toast.*makeText*(MainActivity.this, getString(R.string.*init\_success*), Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 @Override  
 public void onInitFail(String errorMsg) {  
 Toast.*makeText*(MainActivity.this, getString(R.string.*init\_fail*)+ errorMsg, Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 @Override  
 public void onLogoutSuccess() {  
 Toast.*makeText*(MainActivity.this, getString(R.string.*logout\_success*), Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 @Override  
 public void onLogoutFail(String erroMsg) {  
 Toast.*makeText*(MainActivity.this, getString(R.string.*logout\_fail*)+erroMsg, Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 @Override  
 public void onLoginSuccess(ULUUser uluUser) {  
 String uid=uluUser.getUid();  
 String token=uluUser.getToken();  
 Toast.*makeText*(MainActivity.this,getString(R.string.*login\_success*),Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 public void onLoginFail(String erroMsg) {  
 Toast.*makeText*(MainActivity.this,getString(R.string.*login\_fail*)+erroMsg,Toast.*LENGTH\_SHORT*).show();  
  
 }  
});

#### ****3.4.Sdk登录接口（必接）****

ULUManager.*getInstance*().login();

#### ****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.setExtraData("ExtraData");  
uluOrder.setUluProductId(SKU\_GAS);  
  
ULURole uluRole=new ULURole();  
uluRole.setRoleName("RoleName");  
uluRole.setServerName("ServerName");  
uluRole.setServerId("ServerId");  
uluRole.setRoleId("roleId");  
ULUManager.*getInstance*().pay(uluOrder,uluRole,new ULUPayListenter() {  
  
 @Override  
 public void onPaySuccess(String orderId) {  
 Toast.*makeText*(MainActivity.this,getString(R.string.*pay\_success*),Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 @Override  
 public void onPayFail(String orderId, String errorMsg) {  
 Toast.*makeText*(MainActivity.this,getString(R.string.*pay\_fail*)+errorMsg,Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
  
});

#### ****3.7.进入用户中心（必接）****

uluRole = new ULURole();  
uluRole.setRoleLevel("RoleLevel");  
uluRole.setRoleName("RoleName");  
uluRole.setServerName("ServerName");  
uluRole.setServerId("ServerId");  
uluRole.setVipLevel(1);  
uluRole.setRoleId("RoleId");  
ULUManager.*getInstance*().openUserCenter(uluRole);

#### ****3.8.查询游陆内购所有商品信息（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);  
ULUManager.*getInstance*().getProductList(skuList, new ULUGetProductListener() {  
 @Override  
 public void onQuerySuccess(List<ULUProduct> list) {  
 Toast.*makeText*(MainActivity.this,getString(R.string.*query\_success*)+" list.size()==="+list.size(),Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
  
 @Override  
 public void onQueryFail(String errorMsg) {  
 Toast.*makeText*(MainActivity.this,getString(R.string.*query\_fail*),Toast.*LENGTH\_SHORT*).show();  
  
 }  
});

#### ****3.9.检查是否有漏单情况（选接）****

ULUManager.*getInstance*().checkOrder(new ULUcheckOrderListenter() {  
 @Override  
 public void onCheckOrderSuccess(boolean hasOrder) {  
//true 表示有漏单情况，并已经通知后台发货

//false 表示无漏单情况 Toast.*makeText*(MainActivity.this,getString(R.string.*check\_order\_success*)+ " hasOrder=="+hasOrder, Toast.*LENGTH\_SHORT*).show();  
  
 }  
  
 @Override  
 public void onCheckOrderFail(String erroMsg) {  
 Toast.*makeText*(MainActivity.this, getString(R.string.*check\_order\_fail*)+" erroMsg=="+erroMsg, Toast.*LENGTH\_SHORT*).show();  
  
 }  
});

#### ****4.0.打开客服（必接）****

uluRole = new ULURole();  
uluRole.setRoleLevel("RoleLevel");  
uluRole.setRoleName("RoleName");  
uluRole.setServerName("ServerName");  
uluRole.setServerId("ServerId");  
uluRole.setVipLevel(1);  
uluRole.setRoleId("RoleId");  
ULUManager.*getInstance*().openCustomerService(uluRole);