支付宝客户端SDK使用

## 1.简介

由于支付宝SDK主要是在Android Studio导入，由于对Android开发并不熟练，所以在导出arr包到unity中无法使用支付宝SDK，这里自己总结出一套可用的方法：在Android Studio中首先导入支付宝SDK写出需要的方法后导出arr包到unity中调用了该方法，调用后在导出为Android工程后在Androi Studio中重新导入支付宝SDK，最后导出APK即可。

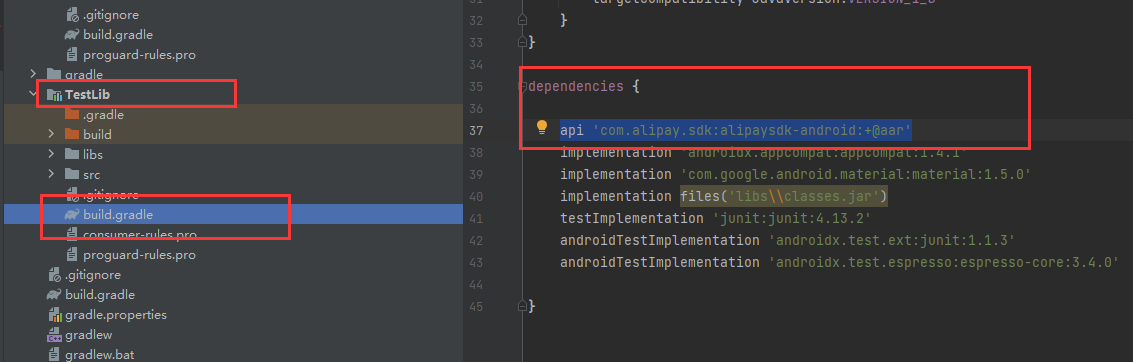
## 2.在Android Studio中导入支付宝SDK

2.1 根据“钉盘->原子比特5->1.4虚拟偶像+ChatGPT中->18 ChatBotAPP->Android Studio到Unity”文件中描述的方法创建一个Android Library在此假设为“TestLib”，同时导入classes.jar。

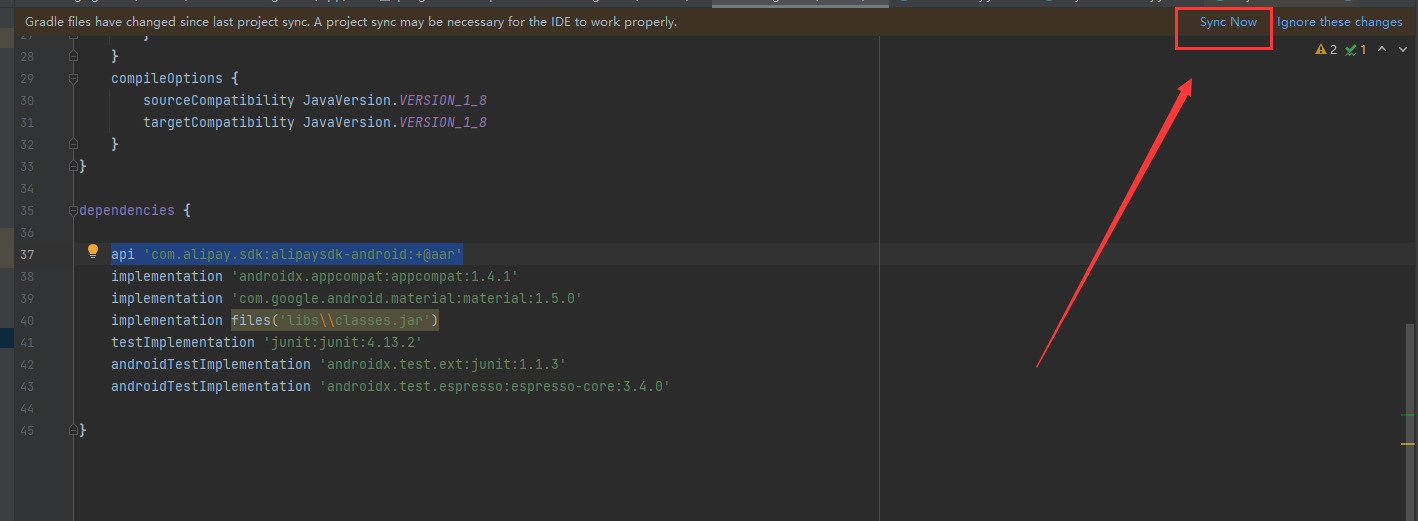
2.2 在[支付宝支付](https://opendocs.alipay.com/open/04km1h)中找到导入导入方法



在新创建的Android Library “TestLib”中找到“build.gradle”，双击打开后在dependencies中输入 api 'com.alipay.sdk:alipaysdk-android:+@aar' 命令，即可添加支付宝SDK。

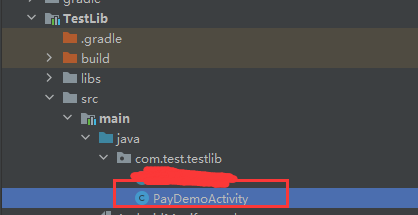


记住输入后点击一下顶端出现的Sync Now。

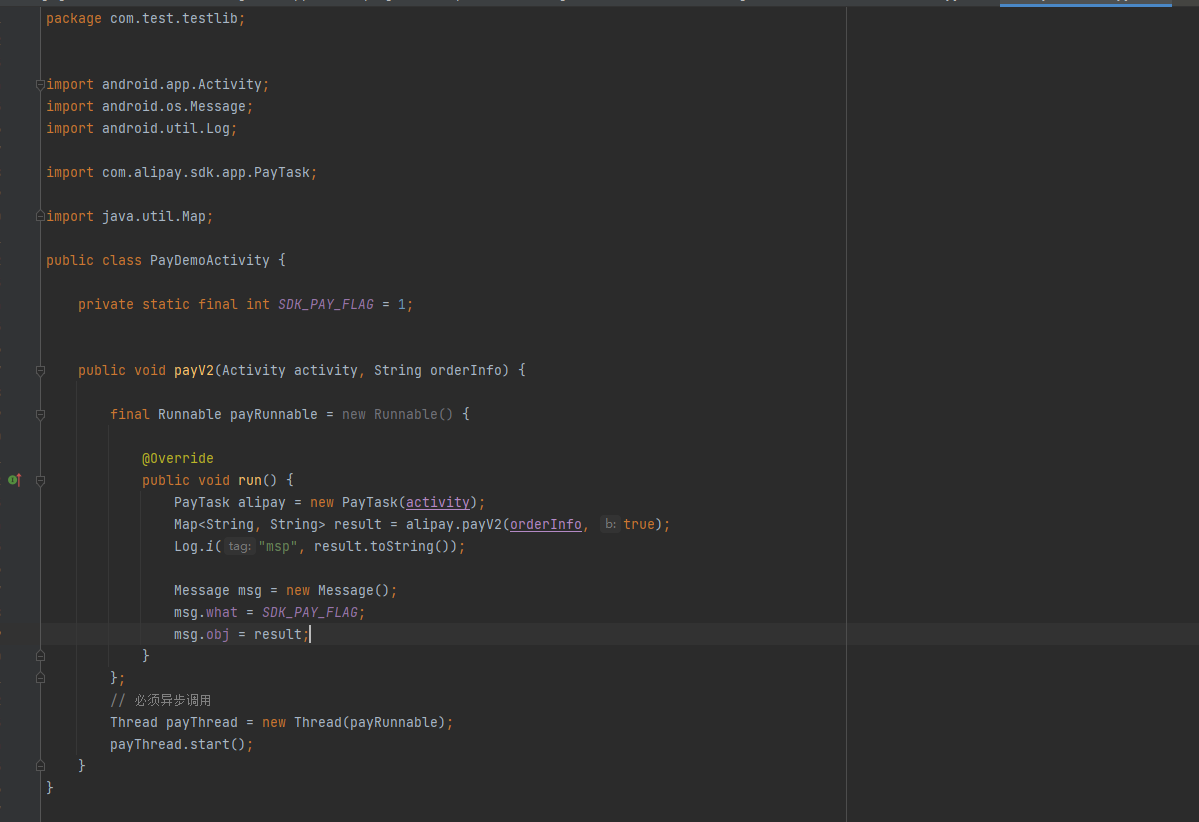


加载完成后即可使用。

2.3 在“TestLib->src->main->java->（包名）”下创建java类，这里可以随便命名，我命名为“PayDemoActivity”



代码如下：



类中有个payV2 （）方法，这个方法即是需要调用的方法，需要传入两个参数，其中activity 可以理解为当前Android界面，orderInfo 即是由服务器生成的可以理解为跳转码。需要将此码给SDK才可以实现跳转支付，由服务器生成。

package com.test.testlib;

import android.app.Activity;

import android.os.Message;

import android.util.Log;

import com.alipay.sdk.app.PayTask;

import java.util.Map;

public class PayDemoActivity {

private static final int SDK\_PAY\_FLAG = 1;

public void payV2(Activity activity, String orderInfo) {

final Runnable payRunnable = new Runnable() {

@Override

public void run() {

PayTask alipay = new PayTask(activity);

Map<String, String> result = alipay.payV2(orderInfo, true);

Log.i("msp", result.toString());

Message msg = new Message();

msg.what = SDK\_PAY\_FLAG;

msg.obj = result;

}

};

// 必须异步调用

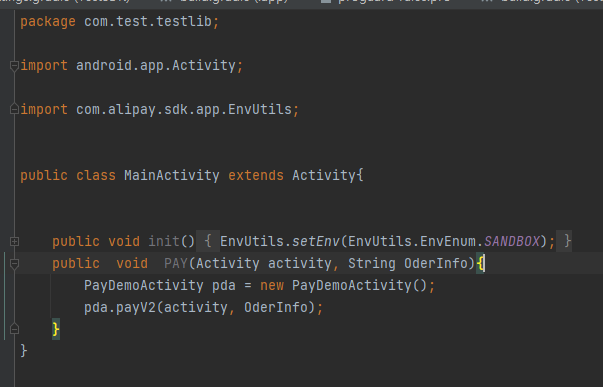
Thread payThread = new Thread(payRunnable);

payThread.start();

}

}

2.4 编写与Unity连接的类，在同一位置创建一个java类，这里名为“MainActivity”



此类的注意事项都在“Android Studio到Unity”文件中，这里只解释方法。该有两个方法，其中init()是初始化方法，“EnvUtils.setEnv(EnvUtils.EnvEnum.SANDBOX);”表示为沙箱环境，正式使用时不需要该方法。导出arr包到unity，具体注意事项与方法也在“Android Studio到Unity”文件中。

package com.test.testlib;

import android.app.Activity;

import com.alipay.sdk.app.EnvUtils;

public class MainActivity extends Activity{

public void init(){

EnvUtils.setEnv(EnvUtils.EnvEnum.SANDBOX);

}

public void PAY(Activity activity, String OderInfo){

PayDemoActivity pda = new PayDemoActivity();

pda.payV2(activity, OderInfo);

}

}

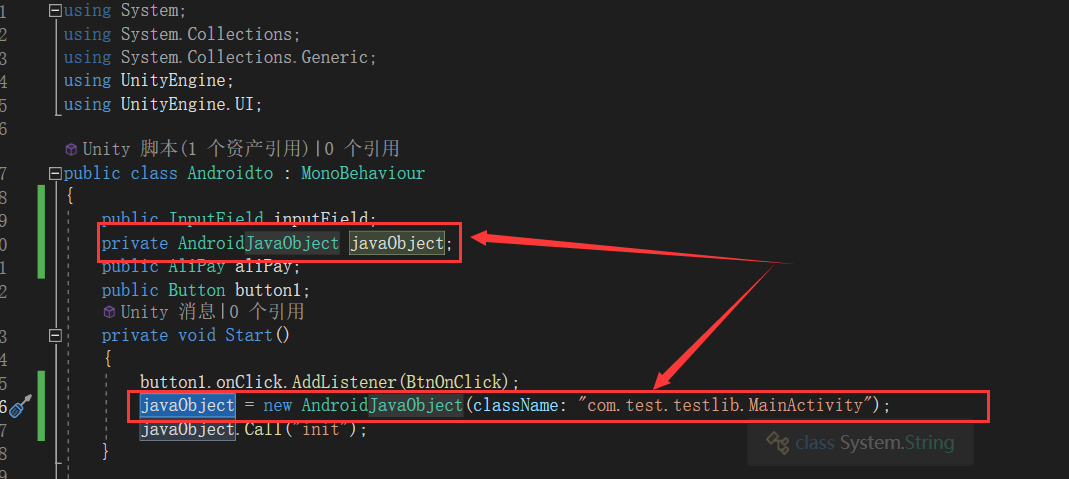
## 3 Unity中调用支付方法

3.1 支付在unity中需要实现的方法很少，包括：

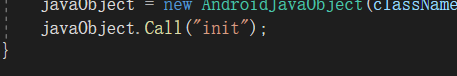
* 获取到需要支付的数据，一般为“商品名”，“订单号”和“金额”，将这些信息发送给服务器返回OrderInfo。
* 将OrderInfo和当前页面传入在Android Studio中编写的方法。
* 若是沙箱环境，需要先调用一下初始化方法。

这里主要说明一下调用Android Studio中的方法，发送给服务器返回OrderInfo将在下面服务器SDK中说明。

3.2 新建一个脚本，在脚本中需要定义的是一个AndroidJavaObject类型的变量



javaObject = new AndroidJavaObject(className: "com.test.testlib.MainActivity");的含义时获取我们在Android Studio中创建的MainActivity类，需要注意的是，格式为：AndroidJavaObject(className: "包名.MainActivity")，这句代码便可以与arr包中的脚本MainActivity连接上。



连接后首先调用一下初始化，不是沙箱环境不需要此操作。

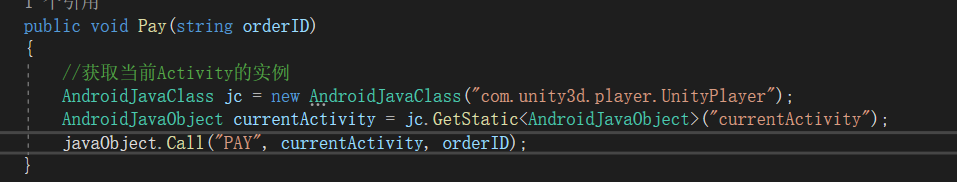
3.3 需要传入的参数不仅有，通过服务器获取的OrderInfo，还需要传入当前页面，在unity中的获取方法为：

//获取当前Activity的实例

AndroidJavaClass jc = new AndroidJavaClass("com.unity3d.player.UnityPlayer");

AndroidJavaObject currentActivity = jc.GetStatic<AndroidJavaObject>("currentActivity");

currentActivity即为当前Android页面，再调用MainActivity中的PAY方法即可



## 4 Unity导出Android工程到Android Studio中重新添加支付宝SDK

4.1在unity中打开Building Settings，在里面勾选“Export Project”，“Build”按钮会变成“Export”，单击“Export”按钮，可以导出为Android Studio文件。

图形用户界面, 应用程序

描述已自动生成

4.2在Android Studio打开该工程，需要注意打开后可能会出现JDK问题的报错，解决方法在setting更改JDK，选择菜单“file”->“setting”

图形用户界面, 文本

描述已自动生成

在“settings”窗口中选择“Build, Execution, Deployment”下的的“Build Tools->Gradle”

文本

描述已自动生成

更改Gradle JDK，与unity中的一致，unity一般为1.8

文本

描述已自动生成

没有的话需要添加，unity下载Android环境是带的JDK位置如图。



这样导入Android Studio成功。

4.3重新添加支付宝SDK，切换到Android视图

文本

描述已自动生成

在如图所示的两个build.gradle中的dependencies添加

api 'com.alipay.sdk:alipaysdk-android:+@aar'

文本

描述已自动生成

文本

描述已自动生成

这样在Android Studio中打包APK之后就可以使用了。