

海游支付SDK 对接文档

- 一、说明
- 二、接入SDK
- 三、商品参数
- 四、google支付参数说明

一、说明

支付 SDK v1.2.2

二、接入SDK

1. 导入相关jar包

Android Studio 工程

将sdk android_studio 目录下的 libs 中文件复制到你项目libs中

项目中引用 aar 包

在 android 同级标签中增加以下代码

```
repositories { flatDir { dirs 'libs' }
```

引用

```
implementation(name: 'haiyou_pay_xx', ext: 'aar')
implementation(name: 'haiyou_common_xx', ext: 'aar')
implementation files('libs/google_pay.jar')
// google_pay.jar是google支付SDK 如果导入时出现包冲突 需要将程序中google支付代码删除
```

完整配置参考demo中的配置

注 `google_pay_xx.jar` 是封装了谷歌支付的jar包，简化Google支付

2. 配置工作

1. 权限

```
<uses-permission android:name="android.permission.INTERNET" />
<!-- 谷歌支付权限 -->
<uses-permission android:name="com.android.vending.BILLING" />
```

2. 配置 appid 和 支付使用的 Activity

在AndroidManife.xml 中配置

```
<application>
    <activity
        android:name="com.example.paysdk.ui.HaiYouPayActivity"
        android:configChanges="screenSize|keyboardHidden|orientation"
        android:theme="@style/walletPayTheme" />
    <activity
        android:name="com.walletfun.common.app.HaiYouShowActivity"
        android:theme="@style/walletPayTheme"
        android:configChanges="screenSize|keyboardHidden|orientation"
        />
    <receiver
        android:name="com.walletfun.common.HaiYouReceiver"
        android:exported="true">
        <intent-filter>
            <action
                android:name="com.android.vending.INSTALL_REFERRER" />
        </intent-filter>
    </receiver>
</application>
```

3. 初始化

在Activity 中 或者 Application 中进行初始化操作

在Activity 中需要在 onDestroy 中进行解绑

```
private void initHaiYouPay() {
    // 在公共模块增加HaiYouPaySDK
    walletHelp.addSDK(HaiYouPaySDK.instance());
    // 设置是否只使用google play 支付
    // HaiYouPaySDK.ALL
    // HaiYouPaySDK.ALL
    // 设置google play 支付公钥
    HaiYouPaySDK.setGooglePublicKey(String GooglePublicey);
    // 设置 运行环境 正式上线时需要设置为false
    walletHelp.setEnvDebug(true);
    // 初始化 参数: context AppID key: 秘钥 sandbox:沙盒开关 (上线时一定要设置为
    SANDBOX_CLOSE)
    walletHelp.init(Context context, String APPID, String Key,
        int walletHelp.SANDBOX_CLOSE);
}
```

```

        // 线上环境时 必须达到额定关卡 等级 是才能显示充值选项 (后台配置 参照服务器配置文档)
        walletHelp.updatePlayer( String grade, String level);
    }

    @Override
    protected void onDestroy() {
        walletHelp.onDestroy(this);
        super.onDestroy();
    }

```

4. 设置支付模式

```

/**
 * 设置支付模式 默认模式 ALL
 * HaiYouPaySDK.ALL : google和第三方支付
 * HaiYouPaySDK.ONLYHAIYOU PAY : 只支持第三方支付
 * HaiYouPaySDK.ONLYGOOGLE : 只支持google 支付
 */
public void setPayMode() {

    HaiYouPaySDK.setPayMode(HaiYouPaySDK.ALL);

}

```

5.支付及监听支付结果,

```

private HaiYouPayHelp haiYouPayHelp;

// 初始化, 此处 Context 可以是任意 Context
haiYouPayHelp = new HaiYouPayHelp(this);

haiYouPayHelp.addHaiYouPayCallBack(haiYouPayCallBack);

/* 开始支付
 * goodsID: 商品ID (google商品ID)
 * money : 商品金额
 * currency : 商品货币单位 美元: USD、韩币: KRW 、人民币: CNY
 * ExtraString : 额外参数
 */
haiYouPayHelp.startPay(String goodsID,Double money,String currency,
                        String ExtraString);

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    // 并且在使用 startActivityForResult 时 requestCode 不要设置为20001 和20002, 防止出现意外情况
    super.onActivityResult(requestCode, resultCode, data);
    if (haiYouPayHelp != null) haiYouPayHelp.onActivityResult(requestCode, resultCode,
data);
}

```

```

    }

@Override
protected void onDestroy() {
    // 需要销毁
    if (haiYouPayHelp != null) {
        haiYouPayHelp.onDestroy();
    }

    super.onDestroy();
}

// 支付回调
haiYouPayCallBack = new HaiYouPayCallBack() {
    @Override
    public void onQueryResult(HaiYouPayResult payResult) {
        LogUtils.e(" 订单详情: " + payResult.toString());
        if (payResult != null && payResult.isValid()) {
            switch (payResult.status) {
                case HaiYouPayResult.PAY_STATUS_PENDING:
                    // 支付中
                    LogUtils.e("支付回调结果 : " + "支付中");
                    break;
                case HaiYouPayResult.PAY_STATUS_SUCCESS:
                    // 支付成功
                    LogUtils.e("支付回调结果 : " + "支付成功");
                    break;
                case HaiYouPayResult.PAY_STATUS_FAIL:
                    // 支付失败
                    LogUtils.e("支付回调结果 : " + "支付失败");
                    break;
                case HaiYouPayResult.PAY_STATUS_REFUND:
                    // 已经退款
                    LogUtils.e("支付回调结果 : " + "已经退款");
                    break;
                default:
                    // 未知状态, 请重新查询或者从自己的服务器上获取数据
                    LogUtils.e("支付回调结果 : " + "未知状态");
                    break;
            }
        }
    }
}

@Override
public void onPayStart(String orderId) {
    textHistory.append("点击支付: orderId= " + orderId + " \n");
    LogUtils.e(" 点击支付: orderId= " + orderId);
}

@Override
public void onGoogleResult(boolean b, Purchase purchase, String s) {
    // 使用google支付回调 (可以在自己服务器处理google支付逻辑)
}

```

```

        LogUtils.e("google支付: 状态= " + b + " \n" + "mes: " + s + " \n");
        Log.e("", "google支付: purchase = " + (purchase == null ? "null" :
        purchase.toString()) + " \n");
    }
};

```

6.引用的class 路径

```

import com.example.paysdk.api.HaiYouPayCallBack;
import com.example.paysdk.api.HaiYouPayResult;
import com.example.paysdk.ui.HaiYouPayHelp;
import com.example.paysdk.ui.HaiYouPaySDK;
import com.walletfun.common.app.WalletHelp;
import com.walletfun.common.floatwindow.WalletFloatWindowListener;
import com.walletfun.common.util.LogUtils;

import cn.chrisx.google.pay.util.Purchase;

```

1. 其他参见接口文档和demo

三、商品参数

```

public class HaiYouPayResult {
    /**
     * 订单 id
     */
    public String orderId;

    /**
     * 额外参数
     */
    public String extraString;
    //商品ID
    public String productId;

    public int resultCode;
    // 货币价格
    public double price;
    // 货币单位
    public String currency;
    // 平台名称
    public String platformName;

    /**
     * 订单状态
     * {@link #PAY_STATUS_PENDING}
     * {@link #PAY_STATUS_SUCCESS}
     * {@link #PAY_STATUS_FAIL}
     * {@link #PAY_STATUS_REFUND}

```

```
* {@link #PAY_STATUS_UNKNOWN}  
*/  
public String status = PAY_STATUS_UNKNOWN;
```

四、google支付参数说明

```
public class Purchase implements Serializable {  
    private String mItemType;           //商品类型 购买或订阅 ITEM_TYPE_INAPP or  
ITEM_TYPE_SUBS  
    private String mOrderId;           // 订单ID  
    private String mPackageName;       // 包名  
    private String mSku;               // 商品Id  
    private long mPurchaseTime;        // 购买时间  
    private int mPurchaseState;        // 购买状态  
    private String mDeveloperPayload;  // 自定义参数  
    private String mToken;             // token  
    private String mOriginalJson;      // INAPP_PURCHASE_DATA （数据集合）  
    private String mSignature;         // INAPP_DATA_SIGNATURE  
    private boolean mIsAutoRenewing;
```