

海游SDK 对接文档

一、说明

二、接入SDK

- 1、导包
- 2、配置
- 3、初始化

三、支付

- 1、设置支付模式
- 2、获取支付权限
- 3、获取支付工具
- 4、设置支付监听
- 5、绑定声明周期
- 6、调用支付
- 7、Android支付返回结果

四、登录

- 1、获取登录工具类
- 2、设置登录结果监听
- 3、绑定声明周期
- 4、调用登录
- 5、查询上次登录信息
- 6、退出登录

一、说明

海游 SDK 包含支付、登录、海游数据统计功能。

二、接入SDK

1、导包

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' }
```

以下两种方法 根据自己开发环境选择一种

方法1、使用gradle 引用

```
implementation(name: 'haiyou_sdk_xx', ext: 'aar')
implementation files('libs/google_pay.jar')

implementation 'com.facebook.android:facebook-android-sdk:4.29.0'
implementation 'com.facebook.android:facebook-login:4.29.0'

implementation 'com.google.android.gms:play-services-auth:16.0.1'
// 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
```

方法2、引用本地文件

```
implementation(name: 'haiyou_sdk_xx', ext: 'aar')
implementation files('libs/google_pay.jar')
// google登录
implementation(name: 'play-services-auth-16.0.1', ext: 'aar')
implementation(name: 'play-services-auth-base-16.0.0', ext: 'aar')
implementation(name: 'play-services-base-16.0.1', ext: 'aar')
implementation(name: 'play-services-basement-16.0.1', ext: 'aar')
implementation(name: 'play-services-tasks-16.0.1', ext: 'aar')
// facebook 登录
implementation(name: 'facebook-core', ext: 'aar')
implementation(name: 'facebook-common', ext: 'aar')
implementation(name: 'facebook-login', ext: 'aar')
implementation(name: 'cardview-v7-28.0.0', ext: 'aar')
```

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

1、权限

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

2、在AndroidManife.xml 中配置

```

<application>
    <!-- 支付页面 -->
    <activity
        android:name="com.example.paysdk.ui.HaiYouPayActivity"
        android:configChanges="screenSize|keyboardHidden|orientation"
        android:theme="@style/walletPayTheme" />
    <!-- 活动页面 -->
    <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>

    <!-- 海游登录sdk 配置 -->
    <activity
        android:name="com.walletfun.login.ui.RegisterOrForgotActivity"
        android:configChanges="screenSize|keyboardHidden|orientation" />

    <activity
        android:name="com.walletfun.login.ui.LoginActivity"
        android:configChanges="screenSize|keyboardHidden|orientation" />
    <!-- 还有活动页面 如果已经注册 无需重复-->
    <activity
        android:name="com.walletfun.common.app.HaiYouShowActivity"
        android:theme="@style/walletPayTheme"
        android:configChanges="screenSize|keyboardHidden|orientation" />

    <!-- facebook 登录 -->
    <meta-data
        android:name="com.facebook.sdk.ApplicationId"
        android:value="@string/facebook_app_id" />

    <!-- 注意: 【FBID】这个值请更换成 facebook_app_id 的值-->
    <provider
        android:name="com.facebook.FacebookContentProvider"
        android:authorities="com.facebook.app.FacebookContentProvider 【FBID】"
        android:exported="true" />

    <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>

</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

1、设置日志开关

参数 true 为打开log日志 false 为关闭日志

```
walletHelp.setEnvDebug(true);
```

2、设置支付和登录参数

```

// googlewebcode: google登录使用 值为google后台的 web客户端参数
HaiYouLoginSdk.setGoogleServerClientId(String googlewebcode);

// googlePulickey: google支付使用 值为google后台的许可密钥
HaiYouPaySDK.setGooglePublicKey(String googlePulickey);

```

3、初始化

```

/**
 * @param context :上下文对象
 * @param appid : 应用标识 值为海游运营提供
 * @param key : 网络密钥 值为海游运营提供
 * @param sandbox :沙盒模式 1 打开 0 关闭 (打开时可海游支付开发测试 不包含Google支付)
 */
walletHelp.init(Context context, String APPID , String APPKEY , int sandbox);

```

4 完整调用方法

```

private void initHaiYouPay() {
    walletHelp.setEnvDebug(true);
    HaiYouLoginSdk.setGoogleServerClientId(svngoogLeLoginID);
}

```

```

        HaiYouPaySDK.setPayMode(HaiYouPaySDK.ALL);
        HaiYouPaySDK.setGooglePublicKey(VPNgooglePublicKey);
        WalletHelp.init(this, "8", "31hriti1133zc2nk", walletHelp.SANDBOX_OPEN);
    }

    // Activity 中调用
    @Override
    protected void onDestroy() {
        walletHelp.onDestroy(this);
        super.onDestroy();
    }

```

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、支付

1、设置支付模式

```

/**
 * 设置支付模式 默认模式 ALL
 * HaiYouPaySDK.ALL : 值为3 google和第三方支付
 * HaiYouPaySDK.ONLYHAIYOU PAY : 值为2 只支持第三方支付
 * HaiYouPaySDK.ONLYGOOGLE : 值为1 只支持google 支付
 */

HaiYouPaySDK.setPayMode(int payMode);

```

2、获取支付权限

```

// grade 关卡 level 等级 (只有达到后台配置的要求才能跳转第三方支付, 否则使用google支付)
walletHelp.updatePlayer(String grade, String level);

```

3、获取支付工具

```
HaiYouPayHelp haiYouPayHelp = new HaiYouPayHelp(this);
```

4、设置支付监听

```
HaiYouPayCallBack 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;

                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");
    }
};

haiYouPayHelp.addHaiYouPayCallBack(haiYouPayCallBack);
```

5、绑定声明周期

```
@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();
}

@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    walletHelp.onConfigurationChanged();
}
```

6、调用支付

```
/* 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);
```

7、Android支付返回结果

```
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;

public class Purchase implements Serializable {
    private String itemType;           //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;

```

四、登录

####

1、获取登录工具类

```
HaiYouLoginHelper helper = HaiYouLoginHelper.getInstance(this);
```

2、设置登录结果监听

```
// 设置登录回调方法
```



```

helper.setLoginCallback(new HaiYouLoginCallback() {
    @Override
    public void onResult(HaiYouLoginResult result) {
        // 登录结果
        switch (result.getStatus()) {
            case HaiYouLoginStatus.LOGIN_SUCCESS:
                // 成功
                resultEdit.append("\n 登录成功");
                loginSuccess(result.getUser());
                break;
            case HaiYouLoginStatus.LOGIN_CANCEL:
                // 用户取消了登录
                resultEdit.append("\n用户取消了登录" +
                    helper.platformName(result.getPlatform()));
                break;
            default:
                // 登录失败
                resultEdit.append("\n登录失败: " +
                    helper.platformName(result.getPlatform()));
                loginFailure(result.getPlatform(), result.getFailed());
                break;
        }
    }
});

```

3、绑定声明周期

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    // 很重要的一步, 自己使用 startActivityForResult 时, requestCode 请不要设置成 10001 到 10005
    super.onActivityResult(requestCode, resultCode, data);
    if (helper != null) helper.handlerActivityResult(requestCode, resultCode, data));
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (helper != null) helper.destroy();
}

```

4、调用登录

```
/* 开始支付
 * LOGIN_PLATFORM_FACEBOOK: 调用facebook登录
 * LOGIN_PLATFORM_GOOGLE : 调用google登录
 * LOGIN_PLATFORM_HAIYOU : 调用hiayou登录
 * LOGIN_PLATFORM_GUEST : 调用游客登录
 * LOGIN_DEFAULT: 调用默认悬浮框登录
 */
helper.showLogin(HaiYouLoginHelper.LOGIN_DEFAULT);
```

5、查询上次登录信息

```
// 查询上次登录是否有效
helper.hrtUser(new HaiYouUserCallback() {
    @Override
    public void onUserResult(HaiYouUser user) {
        // 如果为 null 则说明登录过期, 或者是谷歌登录
        if (user == null) {
            resultEdit.append("请重新登录\n");
        } else {
            loginSuccess(user);
        }
    }
});
```

6、退出登录

```
helper.logout();// 目前退出登录只针对客户端 调用后hrtUser获得的返回值为null
```