|  |
| --- |
| 邻花钱事业部 |
| LHQPay接入文档 |
| 彩生活接入 |
|  |
|  |
|  |

版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 日期 | 修订者 | 概述 |
| 1.0 | 2017-02-21 | 申飞 | 出版 |

目录

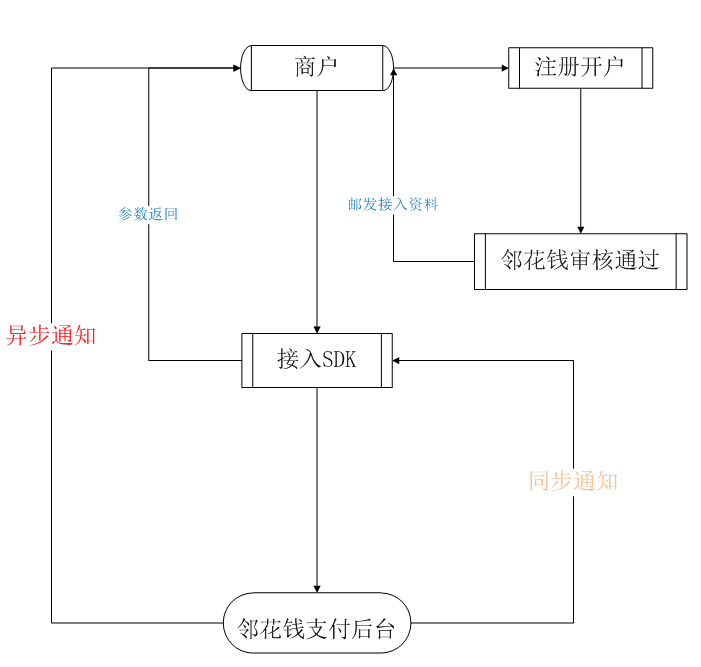
[总体接入流程 3](#_Toc475466018)

[商户注册 3](#_Toc475466019)

[SDK接入 4](#_Toc475466020)

[商户后台回调地址配置及参数验签 16](#_Toc475466021)

# 总体接入流程



\*支付的最终确认：取自“邻花钱支付后台”的异步通知

# 商户注册

1. 注册：<https://merchant.hynpay.com/ewalletMerchant/newMerchantRegist/merRegist.jsp>

等待邮件通知结果。

1. 注册通过后，登录：<https://merchant.hynpay.com/ewalletMerchant/merBackground/merLogin.jsp>
2. 注册通过后，按照所发邮件说明，进行接入开发工作。
3. 特别说明：由于彩生活属于花样年集团公司，开发调测期间，为了推进接入速度，暂免去正式的注册流程。

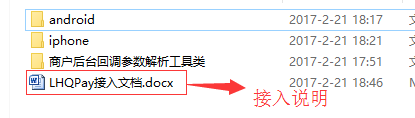
给到临时调测商户号：123456789

# SDK接入

\*API文档见RAP：<http://183.11.217.180:8055/org/index.do>

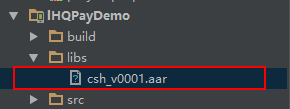
\*测试地址：<http://218.4.234.150:10080/creditsslpay>，如果地址为空，则默认进入邻花钱正式环境。

\*文档目录：



***》》》Android（支持Android studio开发环境）***

1. 将csh\_vxxxx.aar包放入商户应用工程的libs目录下,如果工程下面没有libs文件夹，则自己新建一个，如下图。



PS: csh\_vxxxx.aar中的xxxx代表交付aar文件时的当前版本号，例如：csh\_v0001.aar

1. 修改当前项目的build.gradle配置文件，如下：

添加

repositories {

flatDir {

dirs 'libs' }

}

这个是添加一个本地仓库，并把libs目录作为仓库的地址。

然后在dependencies中添加下面一行代码：

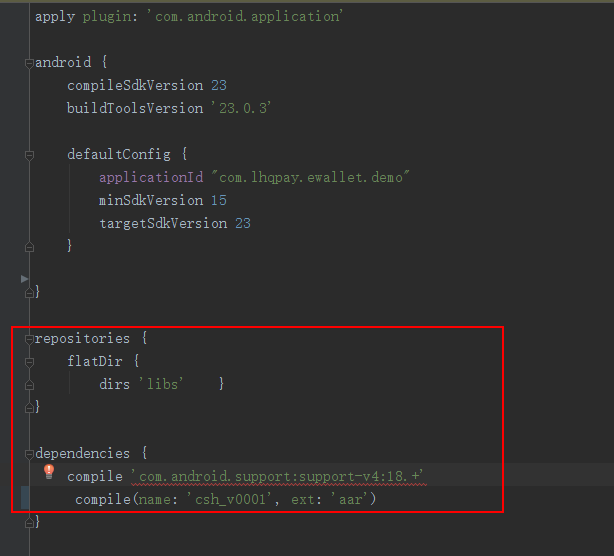
compile(name:'csh­\_vxxxx', ext:'aar')

PS:这里的aar名一定要与接入的arr包名相同

其中name就是libs目录下csh\_v0001.aar文件名称，

ext，就是usericonchooserutil.aar的扩展名

完整添加如下图：



3、重新编译项目：

从工具栏依次选择："Build"-->"rebuild project"

4、修改Manifest

在商户应用工程的AndroidManifest.xml文件里面添加声明：

<activity android:name="com.lhqpay.ewallet.activity.CSHActivity"></activity>  
  
<activity android:name="com.lhqpay.ewallet.activity.CSHCardActivity"  
 android:windowSoftInputMode="stateHidden|adjustPan" ></activity>  
  
<activity android:name="com.lhqpay.ewallet.activity.CSHCardListActivity"></activity>  
  
<activity android:name="com.lhqpay.ewallet.activity.CSHBillActivity"  
 android:windowSoftInputMode="stateHidden|adjustPan"></activity>

<activity android:name="com.lhqpay.ewallet.activity.SupportBankListActivity"

android:windowSoftInputMode="stateHidden|adjustPan"></activity>

和权限声明：

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"/>  
<uses-permission android:name="android.permission.READ\_PHONE\_STATE"/>  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>  
<uses-permission android:name="android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"/>

5、添加混淆规则

在商户应用工程的proguard-project.txt里添加以下相关规则：

-keep class com.lhqpay.ewallet.\*\*{ \*; }

-keep class com.lhqpay.ewallet.adapter.\*\* { \*; }

-keep class com.lhqpay.ewallet.basil.\*\* { \*; }

-keep class com.lhqpay.ewallet.entity.\*\* { \*; }

-keep class com.lhqpay.ewallet.utils.\*\* { \*; }

-keep class com.lhqpay.ewallet.wedgit.\*\* { \*; }

-keep class com.lhqpay.ewallet.lhqpay.listener.\*\* { \*; }

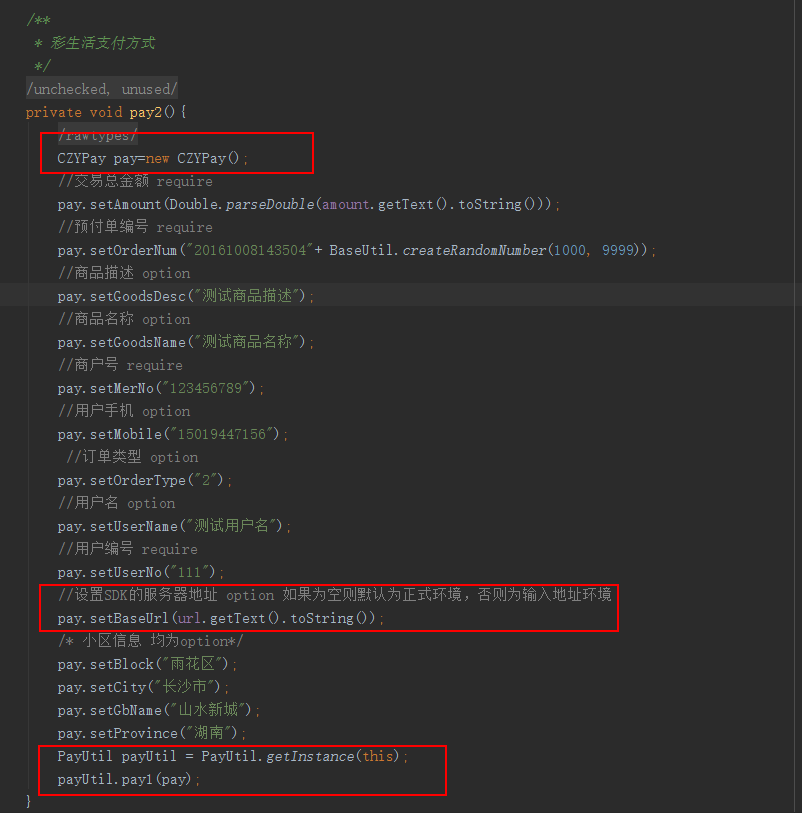
-keep public class com.lhqpay.ewallet.R$\*{

public static final int \*;

}

6、支付接口调用

在调用支付接口的时候，最好新启一个线程，然后创建一个支付实例pay，pay设置参数后调用PayUtil payUtil = PayUtil.*getInstance*(this);最后调用pay1()方法并传入pay实例，代码示例：



7、支付结果获取和处理

本SDK插件采用intent进行跳转传值与返回结果，商户应用客户端通过onActivityResult来接收来自SDK的反馈信息，其中resultCode为10010，实例代码如下：



特别提示：SDK返回的同步通知，不一定准确，因为银行处理有一定的延迟性。为了准确起见，请采用后台的异步通知回调结果。

***》》》iPhone（请参考Demo接入）***

1.文档开放接口说明

1**.1.1** 在进行所有邻花钱钱包操作前, 先配置默认的服务器参数**,**且只配置一次:(方法如下)

/\*\* 邻花钱测试服务器调试模式

\* @param 用户传送需要请求的接口,进行数据请求

\*

\* 注意:当未使用该方法或URL为空时, 默认启用正式服务器

测试环境地址为”http://218.4.234.150:10080/creditsslpay/”

\*/

+(void)setSDKServiceURL:(NSString \*)URL;

调用如下:

1. [ControllerActivity setSDKServiceURL:URL];

2**.2** 方法使用说明**:**

**2.2.1 接口调用**

#import "ControllerActivity.h"

**然后用户再调用ControllerActivity中的类方法**

**2.2.2:**

**(1)**

/\*\* 邻花钱SDK接口调用

\* @param dic 用户提供的NSDictionary数据

需要传的数据

\* @param signBlock 用户签名或加密字符串block:

\* block的参数中signInfo是需要用户进行处理的字符串数据,isSignWithPrivateKey用来判断签名或者加密

\* 当isSignWithPrivateKey=2时,用户需要对signInfo做签名处理

\*/

+(void)setParams:(NSDictionary \*)dic

event:(MMMEventType)event

signInfoBlock:(NSString \*(^)(NSString \*signInfo, BOOL isSignWithPrivateKey))signBlock;

**(2)**

/\*\* 接口调用

\* @param dic 用户提供的NSDictionary数据

\* @param event 用户需要选择的事件:

\* @param signBlock 用户签名或加密字符串block:

\* block的参数中signInfo是需要用户进行处理的字符串数据,isSignWithPrivateKey用来判断签名或者加密

\* 当isSignWithPrivateKey=2时,用户需要对signInfo做签名处理

\* @param resultBlock 用户事件回调block,返回目标事件的数据结果

\* (如用户进行充值事件时,当用户点击充值按钮并从服务器成功返回数据后,执行该block.

\* 用户可根据block提供的NSdictionary参数,自行进行下一步结果处理)

\*/

+(void)setParams:(NSDictionary \*)dic

event:(MMMEventType)event

signInfoBlock:(NSString \*(^)(NSString \*signInfo, BOOL isSignWithPrivateKey))signBlock

resultBlock:(void (^)(NSDictionary \*result, MMMEventType event))resultBlock;

2**.3.3**参数签名方法**.**

关于RSA签名与加密, iOS不进行具体说明, 可参考提供的Demo中RSA签名及加密方法.具体如何操作, 还需用户自由选择

接口调用

\* @param dic 用户提供的NSDictionary数据

需要传的数据

\* @param signBlock 用户签名或加密字符串block:

\* block的参数中signInfo是需要用户进行处理的字符串数据,isSignWithPrivateKey用来判断签名或者加密

\* 当isSignWithPrivateKey=2时,用户需要对signInfo做签名处理

\* @param resultBlock 用户事件回调block,返回目标事件的数据结果

\* (如用户进行充值事件时,当用户点击充值按钮并从服务器成功返回数据后,执行该block.

\* 用户可根据block提供的NSdictionary参数,自行进行下一步结果处理)

\*/

+(void)setParams:(NSDictionary \*)dic

event:(MMMEventType)event

signInfoBlock:(NSString \*(^)(NSString \*signInfo, BOOL isSignWithPrivateKey))signBlock

resultBlock:(void (^)(NSDictionary \*result, MMMEventType event))resultBlock;

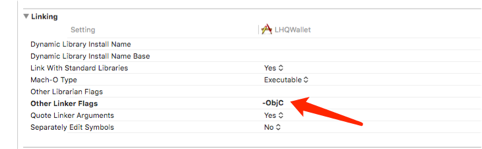
2**.3.3**参数签名与加密方法**.**

关于RSA签名与加密, iOS不进行具体说明, 可参考提供的Demo中RSA签名及加密方法.具体如何操作, 还需用户自由选择

3.1 集成邻花钱插件的工程须知（参照Demo开发）

1.在targets->build Setting 中配置.pch 的路径为当前文件夹的路径

2.在targets->build Setting 中配置Other Link Flags 中的参数设定为-ObjC



3.在targets->General中配置libc++.tbd文件

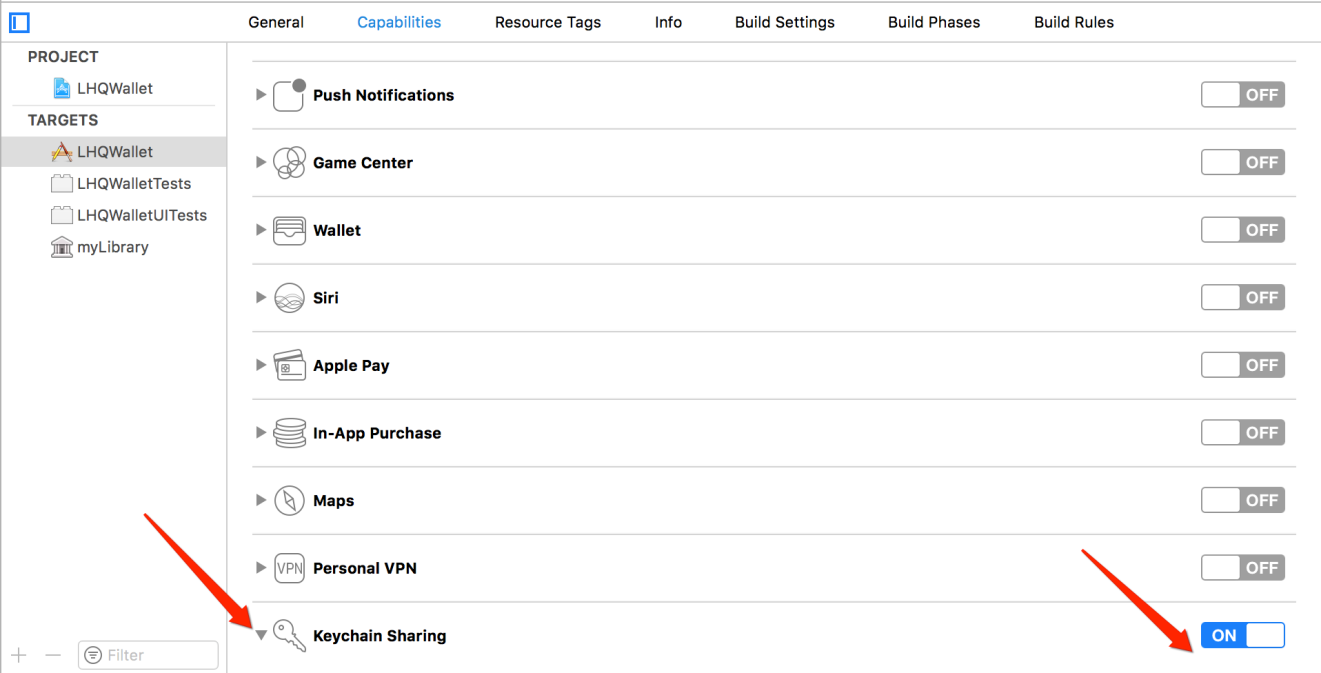


4.将libcrypto.a和libssl.a两个类库拖进工程中，这两个属于openssl库的静态库

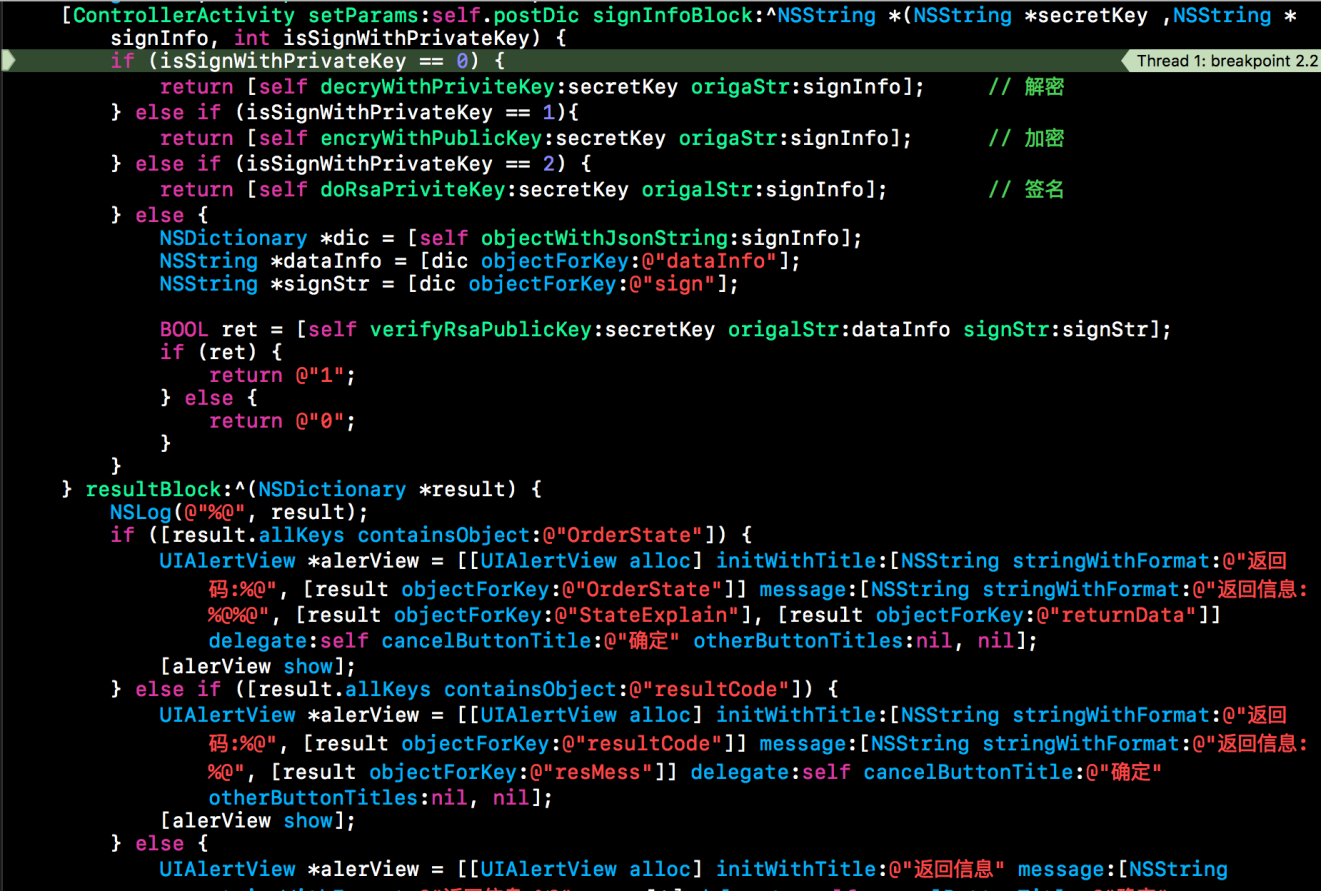
5.将Demo中的CER文件夹拖进工程，如果出现头文件找不到的问题，请在targets->build Setting 中配置Header Search Path 配置CER文件夹的路径

6.请将邻花钱钱包插件拖进项目中，并将图片库LHQResources.bundle库拖进工程

7. xcode8更新后，插件加密方法完全失效，需要打开targets -> Capabilities -> Keychain Sharing



8.上面图片为传递参数的主方法，集成的时候需要配置加密和签名方法，由后台返回，目前解密的方法应商户要求已经删除，商户只需要配置加密和签名方法即可，resultBlock里面为回调参数，插件里面客户操作以后，成功与否，或者客户主动取消交易，都会返回，请商户自己接受判断下一步操作。回调字段，可见，先判断是否有OrderState，这个字段是交易字段，只有在交易后返 回，具体参数见后端接口返回，resultCode为移动端自己定义回调，具体参数暂定：





# 商户后台回调地址配置及参数验签

回调地址，及参数见RAP：<http://183.11.217.180:8055/org/index.do，CSH>模块

回调参数解析：

POST参数过来是明文。商户需要取‘sign’参数值进行解签对比校验，判断合法性，解签的公钥由“邻花钱”提供。

参数解析代码，取“商户后台回调参数解析工具类”文件夹下的工具类，代码示例：

