

MOL Android SDK

Version:1.2.0

**MOL AccessPortal Sdn. Bhd.
2015.05**

前言

1. 面向读者

本文档主要面向需要在Android App中接入MOL支付的android应用开发人员。

2. 读者所需技能

读者需有基本的程序开发背景，掌握java及Android程序开发。

3. 开发环境要求

OS: Windows, Linux, Mac, JRE1.6及以上

Android SDK: SDK2.3及以上

IDE: Eclipse with ADT

Revision History

版本号	作者	内容	日期
1.1.1	Kevin	✧ “EasyToPay” 更改为 ” Easy2Pay”。	2014.09.23
1.1.3	张静	✧ 添加了 PayPal 支付方式。 ✧ 添加了 MOLPay 支付方式 ✧ 修改的混淆打包方式。	2015.03.16
1.2.0	张静	✧ 支付方式升级，统一使用 pay 进行支付 ✧ 删除了以前的 pinpay、Paypal 等支付方式	2015.04.24

目录

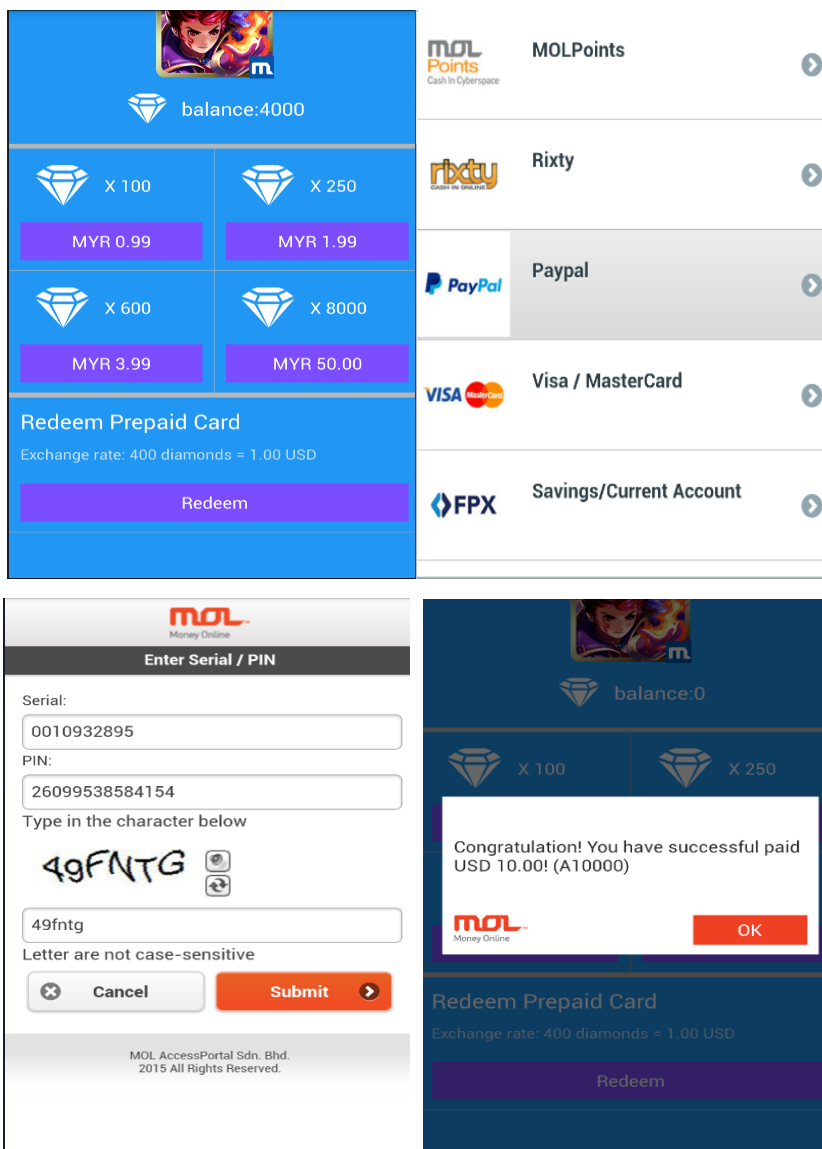
MOL Android SDK	1
第一章 MOL Android SDK 简介	5
1.1 MOL Android SDK 介绍	5
1.2 MOL Android SDK 支付效果图	6
第二章 MOL Android SDK 接入流程	7
2.1 接入前期准备	7
2.1.1 将 SDK 下载到本地，解压后文件与目录说明:	7
2.1.2 开发者账号申请（获取 ApplicationCode 和 SecretKey）	7
2.2 集成	7
2.2.1 导入 jar	7
2.2.2 拷贝需要的资源文件	7
2.2.3 AndroidManifest 需要添加的内容	8
2.2.4 初始化设置	8
2.2.5 Pay 支付	8
2.2.6 Query 查询	10
2.2.7 PaymentListener 回调方法介绍	11
2.2.8 指定屏幕方向	11
2.3 支付接口输入和输出的 Bundle 对象中所包含的字段列举	12
2.4 Bundle 中包含的 Key 和 Value 详细介绍	13
2.5 支付结果参照表（result 和 resultInfo）	14
2.6 支付 Exception 参照表	15
2.7 关于 App 混淆打包支持	16
2.8 支付回调接口（可选）	17
2.8.1 接口介绍	17
2.8.2 支付回调接口登记	17
2.8.3 具体使用	17
2.8.4 回调数据签名验证	17
2.8.5 签名验证例子	18
2.8.6 支付订单相关字段介绍	19
2.9 商户 App MOLStore 支付按钮规范	20

第一章 MOL Android SDK 简介

1.1 MOL Android SDK 介绍

MOL Android SDK 是MOL公司专门为Android App开发的原生支付SDK，商户通过集成MOL Android SDK 可以非常便捷的在App中实现支付功能。目前MOL Android SDK 提供了丰富的支付方式，能够支持各种用户群体方便完成付费。

1.2 MOL Android SDK 支付效果图



第二章 MOL Android SDK 接入流程

2.1 接入前期准备

2.1.1 将 SDK 下载到本地，解压后文件与目录说明：

MOLPoints Logos	2015/5/13 15:17
res	2015/5/13 15:17
Sample	2015/5/13 15:17
MOL Android SDK Developer Guide V1.2.0-CN.PDF	2015/3/27 16:19
MOL Android SDK Developer Guide V1.2.0-EN.PDF	2015/3/27 16:20
MOLPoints_SDK_V1.2.0.jar	2015/3/25 10:55

- MOL_Android SDK_x.x.x.jar: MOL Android SDK jar 包。
- res: MOL Android SDK 使用到的资源文件，需要拷贝 res 文件夹中的内容到对应集成 MOL Android SDK 的项目工程中的同名文件夹下。
- Sample: 一个简单的集成例子
- MOL Android SDK Developer Guide Vx.x.x.pdf: 提供了集成 MOL Android SDK 的方法以及需要注意的细节。
- MOLPoints Logos: 商户集成时需要的按钮图标，参见 [2.9](#)。

2.1.2 开发者账号申请（获取 ApplicationCode 和 SecretKey）

请联系 MOL 市场部相关人员获取。

2.2 集成

2.2.1 导入 jar

只需要拷贝 MOL Android_SDK_x.x.x.jar 到您项目工程下的 libs 文件夹中即可。（如果没有 libs 文件夹请用以下方法导入：在 eclipse 中选中您的项目工程，右键弹出菜单，选择"Build Path" -> "Add External Archives"，在弹出的文件选择框中选取 MOL Android_SDK_x.x.x.jar，确定后完成导入。）

2.2.2 拷贝需要的资源文件

拷贝 [2.1.1](#) 中介绍的 res 文件夹下面所有文件到您项目工程下的 res 文件夹中，所有的 MOLSDK 资源文件名都是以 mol 开头的，不会覆盖您已有的资源文件。

2.2.3 AndroidManifest 需要添加的内容

- 需要添加的权限

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="com.android.launcher.permission.READ_SETTINGS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.SEND_SMS" />
```

新增权限说明"<uses-permission android:name="android.permission.SEND_SMS" />"是Easy2Pay支付方式必须添加的短信发送权限。

- 需要声明的 Activity

```
<activity android:name="com.mol.payment.MOLPointsActivity"
    android:configChanges="orientation|keyboardHidden|screenSize">
</activity>
```

2.2.4 初始化设置

- 测试模式设置

- 方法名称: *setTestMode*
- 示例代码:

```
MOLPayment.setTestMode(true);
```

- 方法描述: 集成测试时首先需要设置 MOLSDK 的运行模式, MOLSDK 运行模式有两个: 测试模式 (true) 和正式发布模式 (false)。如果不设置, 默认为正式发布模式 (false)。

- MOLPayment 支付对象创建

- 支付对象类名: *MOLPayment*
- 实例代码:

```
MOLPayment molPayment = new MOLPayment(this, Secret_Key, Application_Code);
```

- 描述: 需要传递 Context 对象, 以及 [2.1.2](#) 得到的 SecretKey 和 ApplicationCode

2.2.5 Pay 支付

- 方法名称: *pay*

- 方法介绍: *pay* 支付有两种支付方式, 一种是**指定商品价格**的方式, 另一种是**指定虚拟货币兑换率**的方式。

- a) 指定商品价格的方式: (MOL 电子钱包和在线网银支付)

Example #1 :Product A 5000(Diamonds)= MYR 39.99.

- (1) 需要提供的参数: currencyCode=MYR, amount=3999.

(2) 用户可以选择 MOLPoints, PayPal, Rixty, FPX 等等支付渠道进行支付，支付截图如图：2.2.5-1.

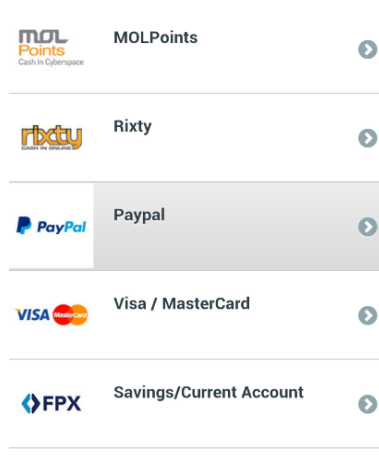


图 2.2.5-1

b) 指定虚拟货币兑换率的方式，采用虚拟币支付（EasytoPay 和其他在线银行支付）

Example #2： 商户需要指定充值 1USD 对应的虚拟货币数量，如"300 Diamonds =1 USD"。

(1) 需要传递参数：virtualCurrencyName=" Diamonds ",virtualCurrencyRate=300

(2) 用户可以选择 Easy2Pay、Direct Top Up、Zest Card、MOL Points Card 等支付渠道进行支付。如图 2.2.5-2

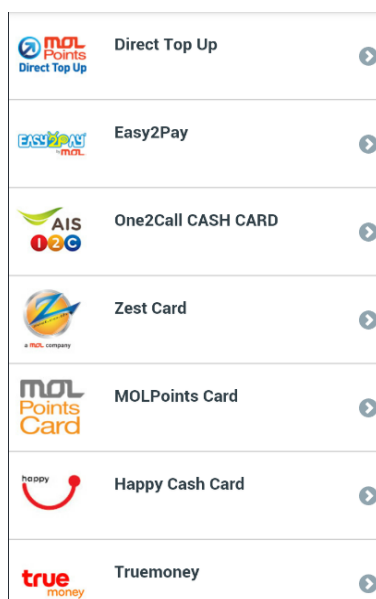


图 2.2.5-2

● 示例代码：

■ 指定商品价格，Bundle 对象必须包含 currencyCode、amount 参数：

```
Bundle inputBundle = new Bundle();
inputBundle.putString(MOLConst.B_Key_ReferenceId, "TRX1708901"); // Must
inputBundle.putLong(MOLConst.B_Key_Amount, 3999);
inputBundle.putString(MOLConst.B_Key_CurrencyCode, "MYR");
inputBundle.putString(MOLConst.B_Key_Description, "5000 diamonds"); //Optional
inputBundle.putString(MOLConst.B_Key_CustomerId, "12321144221"); // Must
try {
```

```

molPayment.pay(this, inputBundle, new PaymentListener() {
    @Override
    public void onBack(int action, Bundle outputBundle) {
        // TODO Auto-generated method stub
        showInfo(outputBundle.toString());
    }
});
} catch (Exception e) {
    showInfo(e.getMessage());
}

```

- 指定虚拟货币兑换率，Bundle 对象建议包含 `virtualCurrencyName` 、`virtualCurrencyRate` 参数：

```

Bundle inputBundle = new Bundle();
inputBundle.putString(MOLConst.B_Key_ReferenceId, "TRX1708901"); // Must
inputBundle.putString(MOLConst.B_Key_VirtualCurrencyName, "Diamond");
inputBundle.putFloat(MOLConst.B_Key_VirtualCurrencyRate, 300f);
inputBundle.putString(MOLConst.B_Key_Description, "Product A"); //Optional
inputBundle.putString(MOLConst.B_Key_CustomerId, "12321144221"); //Must
try {
    molPayment.Pay(this, inputBundle, new PaymentListener() {
        @Override
        public void onBack(int action, Bundle outputBundle) {
            // TODO Auto-generated method stub
            showInfo(outputBundle.toString());
        }
    });
} catch (Exception e) {
    showInfo(e.getMessage());
}

```

- 需要传入的参数：pay 方法需要传递
 - Context 对象
 - Bundle 对象：传递支付相关信息
 - PaymentListener 回调对象：用于接收支付结果
- 当传递的参数不合法时（比如 Context 为空、Bundle 中缺少 ReferenceId 等）会抛出 Exception 异常，错误信息可通过 `e.getMessage()` 得到。
- 更多：[Bundle 中的具体内容](#)、[抛出 Exception 列表](#)、[PaymentListener 详细介绍](#)

2.2.6 Query 查询

- 方法名称：`paymentQuery`
- 示例代码：

```

Bundle inputBundle = new Bundle();
//ReferenceId or PaymentId is required
inputBundle.putString(MOLConst.B_Key_ReferenceId, "TRX1708901");
//inputBundle.putString(MOLConst.B_Key_PaymentId, "MPO101913");

```

```
try {
    molPayment.paymentQuery(this, inputBundle, new PaymentListener() {
        @Override
        public void onBack(int action, Bundle outputBundle) {
            // TODO Auto-generated method stub
            showInfo(outputBundle.toString());
        }
    },false);
} catch (Exception e) {
    showInfo(e.getMessage());
}
```

- 是否后台查询的 boolean 值：ture 为后台查询，false 为显示 LoadingDialog。
- 当传递的参数不合法时（比如 Context 为空、Bundle 中缺少 ReferenceId 等）会抛出 Exception 异常，错误信息可通过 e.getMessage()得到。
- 更多：[Bundle 中的具体内容](#)、[抛出 Exception 列表](#)、[PaymentListener 详细介绍](#)

2.2.7 PaymentListener 回调方法介绍

- 描述：SDK 支付或查询完成后调用 PaymentListener 对象的 onBack()方法向商户返回操作结果。
- onBack 的参数：
 - action: 表示操作类型（目前有两种类型：[MOLConst.Action_Pay](#)、[MOLConst.Action_Query](#) 分别表示 pay 支付和查询操作），主要是为了支持一个 PaymentListener 对象在多个接口中使用。
 - resultdata: 为操作的结果，具体请参考 [2.3 Bundle 对象中所包含的内容列举](#)和 [2.4 Bundle 中包含的 Key 和 Value 详细介绍](#)。

2.2.8 指定屏幕方向

为了保证支付过程不被屏幕方向改变而中断，您需要在调用支付接口的 Activity 中实现下面两项中的一项：

- **<推荐>**您在 AndroidManifest 文件中指定该 Activity 的屏幕方向，例如：

```
<activity android:name=".PinPayActivity" android:screenOrientation="landscape"></activity>
```

或者

```
<activity android:name=".PinPayActivity" android:screenOrientation="portrait"></activity>
```

- 您的 Activity 处理屏幕改变事件，例如：
 - 1).AndroidManifest 文件中指定 android:configChanges

```
<activity android:name=".PinPayActivity" android:configChanges="orientation/screenSize"></activity>
```

- 2).Activity 中重载 onConfigurationChanged 方法

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

2.3 支付接口输入和输出的 **Bundle** 对象中所包含的字段列举

Method	Bundle	Value	Required
Pay	Input Bundle	<i>referenceId</i>	Must
		<i>amount</i>	Optional(must when currencyCode exist)
		<i>currencyCode</i>	Optional(must when amount exist)
		<i>customerId</i>	Must
		<i>description</i>	Optional
		<i>virtualCurrencyName</i>	Optional(must when virtualCurrencyRate exist)
		<i>virtualCurrencyRate</i>	Optional(must when virtualCurrencyName exist)
	Output Bundle	<i>result</i>	Must
		<i>resultInfo</i>	Must
		<i>referenceId</i>	Must
		<i>paymentId</i>	Must(result==MOLConst.Result_Success)
		<i>currencyCode</i>	Optional (result==MOLConst.Result_Success)
		<i>amount</i>	Optional (result==MOLConst.Result_Success)
		<i>paymentStatusDate</i>	Must(result==MOLConst.Result_Success)
paymentQuery	Input Bundle	<i>referenceId</i>	Must(referenceId or paymentId)
		<i>paymentId</i>	
	Output Bundle	<i>result</i>	Must
		<i>resultInfo</i>	Must
		<i>referenceId</i>	Must
		<i>paymentId</i>	Must(result==MOLConst.Result_Success)
		<i>currencyCode</i>	Must(result==MOLConst.Result_Success)
		<i>amount</i>	Must(result==MOLConst.Result_Success)
		<i>paymentStatusDate</i>	Must(result==MOLConst.Result_Success)
		<i>virtualCurrencyAmount</i>	Optional
		<i>customerId</i>	Optional

2.4 Bundle 中包含的 Key 和 Value 详细介绍

- **referenceId:**
 - 描述: Reference Id is a unique identifier generated by merchant for each distinct transaction.
 - 数据类型: String (Max Length 50 Characters)
 - Bundle Key: [MOLConst.B_Key_ReferenceId](#)
- **customerId:**
 - 描述: Customer Id is a unique identifier of customer generated by the merchant.
 - 数据类型: String (Max Length 50 Characters)
 - Bundle Key: [MOLConst.B_Key_CustomerId](#)
- **description:**
 - 描述: Payment description refers to statement that describes the payment. The statement will be displayed in UI of the several payment providers.
 - 数据类型: String (Max Length 50 Characters)
 - Bundle Key: [MOLConst.B_Key_Description](#)
- **amount:**
 - 描述: Amount refers as payment/redemption amount of the transaction in fractional unit (lowest common denominator) of the respective currency code. Thousand comma separator should be removed before assign value to this parameter.
 - 数据类型: Integer (Max Length 20 numbers)
 - Bundle Key: [MOLConst.B_Key_Amount](#)
- **result:**
 - 描述: 支付或查询操作返回的状态码，具体参见返回状态码表格。
 - 数据类型: String
 - Bundle Key: [MOLConst.B_Key_Result](#)
- **resultInfo:**
 - 描述: 支付或查询操作返回的操作信息，具体参见返回状态码表格。
 - 数据类型: String
 - Bundle Key: [MOLConst.B_Key_Result_Info](#)
- **paymentId:**
 - 描述: Payment Id is a unique identifier given by MOL Payout for transaction references purpose.
 - 数据类型: String (Max Length 50 Characters)
 - Bundle Key: [MOLConst.B_Key_PaymentId](#)
- **paymentStatusDate:**
 - 描述: Payment Status Date indicates the last updated date of the payment's status. The date will be in UTC (Coordinated Universal Time) format.
 - 数据类型: String
 - 格式: yyyy-MM-ddTHH:mm:ssZ
 - Bundle Key: [MOLConst.B_Key_PaymentStatusDate](#)
- **VirtualCurrencyName**
 - 描述: 虚拟货币的名称。例如 400 Diamonds = 1 USD, VirtualCurrencyName= Diamonds
 - 数据类型: String
 - Bundle Key: [MOLConst.B_Key_VirtualCurrencyName](#)
- **VirtualCurrencyRate**

- 描述: VirtualCurrencyRate 是 1 美元对应的虚拟货币的转化率, 例如: 400 Diamonds = 1 USD, VirtualCurrencyRate=400.0;
- 数据类型: Float
- Bundle Key: [MOLConst.B_Key_VirtualCurrencyRate](#)
- VirtualCurrencyAmount
 - 描述: VirtualCurrencyAmount 是服务器返回的虚拟货币充值数量, 可供商户参考。
 - 数据类型: Double
 - Bundle Key: [MOLConst.B_Key_VirtualCurrencyAmount](#)
- currencyCode:
 - 描述: Currency Code refers to three characters global currencies code as refer to ISO 4217.
 - 数据类型: String (Max Length 3 Characters)
 - Bundle Key: [MOLConst.B_Key_CurrencyCode](#)
 - 目前支持的 currencyCode 类型:

Currency Code	Decimal Places	Example	Amount in Fractional Unit
USD	2	USD 1.00	100
MYR	2	MYR 1.00	100
AUD	2	AUD 1.00	100
BRL	2	BRL 1.00	100
IDR	2	IDR 1.00	100
INR	2	INR 1.00	100
NZD	2	NZD 1.00	100
PHP	2	PHP 1.00	100
SGD	2	SGD 1.00	100
THB	2	THB 1.00	100
TWD	2	TWD 1.00	100
VND	2	VND 1.00	100

2.5 支付结果参照表 (result 和 resultInfo)

Result	MOLConst value	resultInfo	备注
A10000	Result_Success	{success info}	支付成功
A10001	Result_TimeOut	Network timeout.	联网超时, 请通过查询接口确认订单状态
A10002	Result_User_CancelPayment	User cancel the payment.	用户取消了支付

A10004	<i>Result_NetWork_Fail</i>	<i>network failed</i>	网络错误， 请通过查询接口确认订单状态
A10005	<i>Result_InComplete</i>	<i>Payment has not complete or in middle of processing</i>	订单还没完成， 请稍后通过查询接口确认订单状态
A10006	<i>Result_Payment_Expired</i>	<i>Payment has been failed as expired.</i>	订单超时， 支付失败
A10007	<i>Result_Proceed_Fail</i>	<i>Payment for the given transaction failed.</i>	订单支付失败
A10019	<i>Result_UnSupport</i>	<i>The device unSupport.</i>	用户设备不支持，基本不会发生这个错误（比如不支持 UTF-8 编码）
A10020	<i>Result_SDK_Error</i>	<i>MOL SDK error.</i>	MOL SDK 错误，基本不会发生，如果发生了，请重试，或者升级 MOL SDK
40003	<i>Result_CurrencyCode_Invalid</i>	<i>Invalid CurrencyCode</i>	货币类型有误或者不支持
40004	<i>Result_Duplicate_ReferenceId</i>	<i>Duplicate Reference Id.</i>	该订单号已存在，请传入新的订单号进行支付，也可以通过查询接口查询该订单号的状态
40008	<i>Result_InsufficientBalance_Invalid id</i>	<i>Insufficient Balance</i>	余额不足
40101	<i>Result_ApplicationCode_Invalid</i>	<i>Invalid Application Code.</i>	<i>Invalid Application Code.</i>
40103	<i>Result_Signature_Invalid</i>	<i>Invalid secret key.</i>	<i>Invalid secret key.</i>
40106	<i>Result_NotTransact_Wallet_Invalid</i>	<i>User does not has matched wallet to transact.</i>	<i>User does not has matched wallet to transact.</i>
40107	<i>Result_Transaction_expired</i>	<i>User does not has matched wallet to transact.</i>	<i>Payment has been failed</i>
40010	<i>Result_Invalid_CarrierCode</i>	<i>Invalid Carrier Code or not supported.</i>	错误或不支持的运营商编号
40011	<i>Result_Invalid_AmountOrCurrencyCode</i>	<i>Invalid Amount or Currency Code not supported.</i>	充值金额或者货币类型不支持、可以向市场部相关人员获取支持的金额列表。
40109	<i>Result_BlackList_TelNo</i>	<i>MSISDN is blacklisted.</i>	用户电话号码被列入黑名单
40012	<i>Result_WrongFormat_TelNo</i>	<i>MSISDN is giving in a wrong format</i>	用户电话号码输入错误
40013	<i>Result_Exceed_Accept_Amount</i>	<i>Exceed channel accepted amount</i>	金额超过最大值
40014	<i>Result_Below_Accept_Amount</i>	<i>Below channel accepted amount.</i>	金额小于最小值

2.6 支付 Exception 参照表

错误类型	Exception message
<i>callback listener is null</i>	<i>The paymentListener is null.</i>
<i>applicationCode invalid(null or len>50)</i>	<i>Invalid Application Code.</i>
<i>secretkey invalid(null or len>50)</i>	<i>Invalid SecretKey.</i>
<i>Android Context is null</i>	<i>The context is null.</i>
<i>referenceID invalid(null or len>50)</i>	<i>Invalid Reference Id.</i>
<i>customerID invalid(len>50)</i>	<i>The customerId's length is exceeds the maximum length(50)!</i>

currencyCode invalid(len>3)

Invalid CurrencyCode

No SIM Card available.

No SIM Card available.

2.7 关于 App 混淆打包支持

如果您的 App 发布时需要混淆打包 apk，请在你的混淆配置文件中添加：

```
-keep class mol.payment.test.R${*};
-keepattributes InnerClasses,*Annotation*
-keep class com.mol.payment.* {
    <fields>;
    <methods>;
}
```

其中“-keep class mol.points.sample.R\${*};”中的包名“mol.points.sample”需修改为自己定义的包名。
例如：

```
-keep public class * extends android.app.Activity
-keep public class * extends android.app.Application
-keep public class * extends android.app.Service
-keep public class * extends android.content.BroadcastReceiver
-keep public class * extends android.content.ContentProvider
-keep public class * extends android.app.backup.BackupAgentHelper
-keep public class * extends android.preference.Preference
-keep public class com.android.vending.licensing.ILicensingService
-keep public class * extends android.os.IInterface
```

```
-keep class mol.points.sample.R${*};
-keepattributes InnerClasses,*Annotation*
-keep class com.mol.payment.* {
    <fields>;
    <methods>;
}
-keepattributes InnerClasses,*Annotation*
-keepclasseswithmembernames class * {
    native <methods>;
}

-keepclasseswithmembers class * {
    public <init>(android.content.Context, android.util.AttributeSet);
}
```


2.8 支付回调接口（可选）

2.8.1 接口介绍

支付回调接口是商户服务器接收支付通知的接口,用于 MOL 服务器向商户服务器发送已支付完成的用户订单信息。

2.8.2 支付回调接口登记

请联系 MOL 市场部相关人员进行登记。

2.8.3 具体使用

Name	Parameters
HTTP Method	POST /{ callback URL }
Request Parameters in HTTP Body (x-www-form-urlencoded format)	Format: applicationCode ={ applicationCode }& referenceId ={ referenceId }& paymentId ={ paymentId }& version ={ version }& amount ={ amount }& currencyCode ={ currencyCode }& paymentStatusCode ={ paymentStatusCode }& paymentStatusDate ={ paymentStatusDate }& customerId ={ customerId }& signature ={ signature } Example: applicationCode =3f2504e04f8911d39a0c0305e82c3301& referenceId =TRX1708901& paymentId =MPO000000000001& version =v1& amount =1000& currencyCode =MYR& paymentStatusCode =00& paymentStatusDate =2012-12-31T14%3A59%3A59Z& customerId =12321144221& signature =5e2a170eabcb54db0b2937874c39549b
Response	<i>Merchant server just need to response HTTP 200 to MOL server when receive callback notification.</i>

2.8.4 回调数据签名验证

- 支付回调接口可用 MD5 来验证接收到的数据是否合法有效。
- 签名字符串是参数名为“signature”对应的参数值“5e2a170eabcb54db0b2937874c39549b”，是通过
对支付订单信息和 SecretKey 进行 MD5 加密产生的 MD5 串。
- 签名验证方法：对接收到的支付订单信息进行签名得到 MD5 串（签名具体步骤见下面），将得到的 MD5 串与接收到的 **signature** 字段值进行比对，相同则合法，不同则非法。
- 签名验证步骤：
 - 对接收到的所有参数名（[2.8.3](#) 表格中 request 内容）按字母顺序进行排序。**注意：**

- ◆ 值为空的参数名不参与排序
- ◆ **signature** 不参与排序
- 将已排好序的参数名，根据顺序依次取出对应的参数值拼接成一个“字符串 A”。**注意：**
 - ◆ 参数值需要原始值，所以需要对每个参数值进行 URL decode
 - ◆ 参数值区分大小写
 - ◆ 参数值头尾如果有空格需要去除
- 将 ApplicationCode 对应的 SecretKey 添加到“字符串 A”的尾部（“字符串 A”= “字符串 A ” + {SecretKey}）。
- 对“字符串 A”进行 MD5 加密，得到“字符串 B”。
- “字符串 B”与 **signature** 字段值进行比对，相同则合法，不同则非法。

2.8.5 签名验证例子

- 支付回调接口接收到的数据：

```
applicationCode=3f2504e04f8911d39a0c0305e82c3301&referenceId=TRX1708901&paymentId=MPO0000000000001&version=v1&amount=1000&currencyCode=MYR&paymentStatusCode=00&paymentStatusDate=2012-12-31T14%3A59%3A59Z&customerId=12321144221&signature=5e2a170eabcb54db0b2937874c39549b
```

- 得到参数名和参数值映射表（参数值已 URL decode）：

Parameter Name	Value
applicationCode	3f2504e04f8911d39a0c0305e82c3301
referenceId	TRX1708901
paymentId	MPO0000000000001
version	v1
paymentStatusCode	00
paymentStatusDate	2012-12-31T14:59:59Z
amount	1000
currencyCode	MYR
customerId	12321144221

- 对参数名进行排序：

```
{ amount } + { applicationCode } + { currencyCode } + { customerId } + { paymentId } + { paymentStatusCode } + { paymentStatusDate } + { referenceId } + { version }
```

- 根据排序拼接参数值：

```
10003f2504e04f8911d39a0c0305e82c3301MYR12321144221MPO0000000000001002012-12-31T14:59:59ZTRX1708901v1
```

- 添加 SecretKey:

```
10003f2504e04f8911d39a0c0305e82c3301MYR12321144221
MPO0000000000001002012-12-31T14:59:59ZTRX1708901v1Ziu61T9xY227aazS530Pk8
C5424v663r
```

- 得到 MD5 字符串:

```
MD5(10003f2504e04f8911d39a0c0305e82c3301MYR12321144221
MPO0000000000001002012-12-31T14:59:59ZTRX1708901v1Ziu61T9xY227aazS530Pk8
C5424y663r) = 5e2a170eabcb54db0b2937874c39549b
```

- 进行比对:

```
接收到的签名: 5e2a170eabcb54db0b2937874c39549b
计算出的签名: 5e2a170eabcb54db0b2937874c39549b
比对结果: 签名相同
```

2.8.6 支付订单相关字段介绍

字段名	介绍															
applicationCode	Application Code is uniquely identifying merchant application or portal which integrating with MOL Payout API.															
referenceId	Reference Id is a unique identifier generated by merchant for each distinct transaction.															
paymentId	Payment Id is a unique identifier given by MOL Payout for transaction references purpose.															
amount	Amount refers as payment/redemption amount of the transaction in fractional unit (lowest common denominator) of the respective currency code. Thousand comma separator should be removed before assign value to this parameter.															
currencyCode	Currency Code refers to three characters global currencies code as refer to ISO 4217.															
paymentStatusCode	Payment Status Code refers to two characters status indicator for Success, Failure, or Pending of a payout payment transaction. <table><tr><th>Code</th><th>Message</th><th>Description</th></tr><tr><td>00</td><td>Success</td><td>Payment completed and paid.</td></tr><tr><td>01</td><td>Incomplete</td><td>Payment has not complete or in middle of processing.</td></tr><tr><td>02</td><td>Session Expired</td><td>Payment has been failed as expired.</td></tr><tr><td>99</td><td>Failure</td><td>Payment for the given transaction failed.</td></tr></table>	Code	Message	Description	00	Success	Payment completed and paid.	01	Incomplete	Payment has not complete or in middle of processing.	02	Session Expired	Payment has been failed as expired.	99	Failure	Payment for the given transaction failed.
Code	Message	Description														
00	Success	Payment completed and paid.														
01	Incomplete	Payment has not complete or in middle of processing.														
02	Session Expired	Payment has been failed as expired.														
99	Failure	Payment for the given transaction failed.														

<u>paymentStatusDate</u>	Payment Status Date indicates the last updated date of the payment's status. The date will be in UTC (Coordinated Universal Time) format.
<u>customerId</u>	Customer Id is a unique identifier of customer generated by the merchant.
signature	Value returned by MOL for Merchant to validate against the returned parameters.

2.9 商户 App MOLStore 支付按钮规范

商户 App 中支付方式选择页面，MOLSDK 支付操作按钮图片请使用如下图标：



Money Online

(sdk rar 压缩包中有原图)