

百度钱包对外接入手册

Android 端

文件版本：2.2

百度在线网络技术（北京）有限公司

(版权所有,翻版必究)

修改记录

No	版本号	修改内容简介	修改日期	修改人
1	1.0	初稿	2014.7.16	陈锋、李博
2	1.1	开放话费充值接口	2014.07.31	陈锋
3	1.3	补充新增接口——getLoginData()	2015.2.4	杨文燕
4	1.4	补充参数长度描述	2015.2.13	杨文燕
5	1.5	增加返现的参数	2015.11.5	熊雷
6	2.0	去掉 ios 部分，两个端文档分开，附录合入到主文档，商服版	2015.12.7	熊雷
7	2.1	增加了主动绑卡的业务	2015.12.23	王鼎淳
8	2.2	增加银行卡扫描和指纹支付模块	2016.06.07	田征绿

目 录

1 百度钱包简介	1
2 百度钱包支付集成	1
2.1 百度钱包支付业务流程	1
2.2 支付流程	1
2.3 百度钱包 SDKDemo 集成示例	2
2.3.1 Demo 配置运行	2
2.3.2 Demo 结构说明	5
2.4 百度钱包 SDK 集成	6
2.4.1 接口说明	6
2.4.2 接入步骤	8
2.5 支付参数列表	12
2.6 主动绑卡参数列表	14
3 签名机制	15
3.1 签名原理	15
3.2 签名步骤	15
4 通知结果	16
4.1 同步返回结果	16

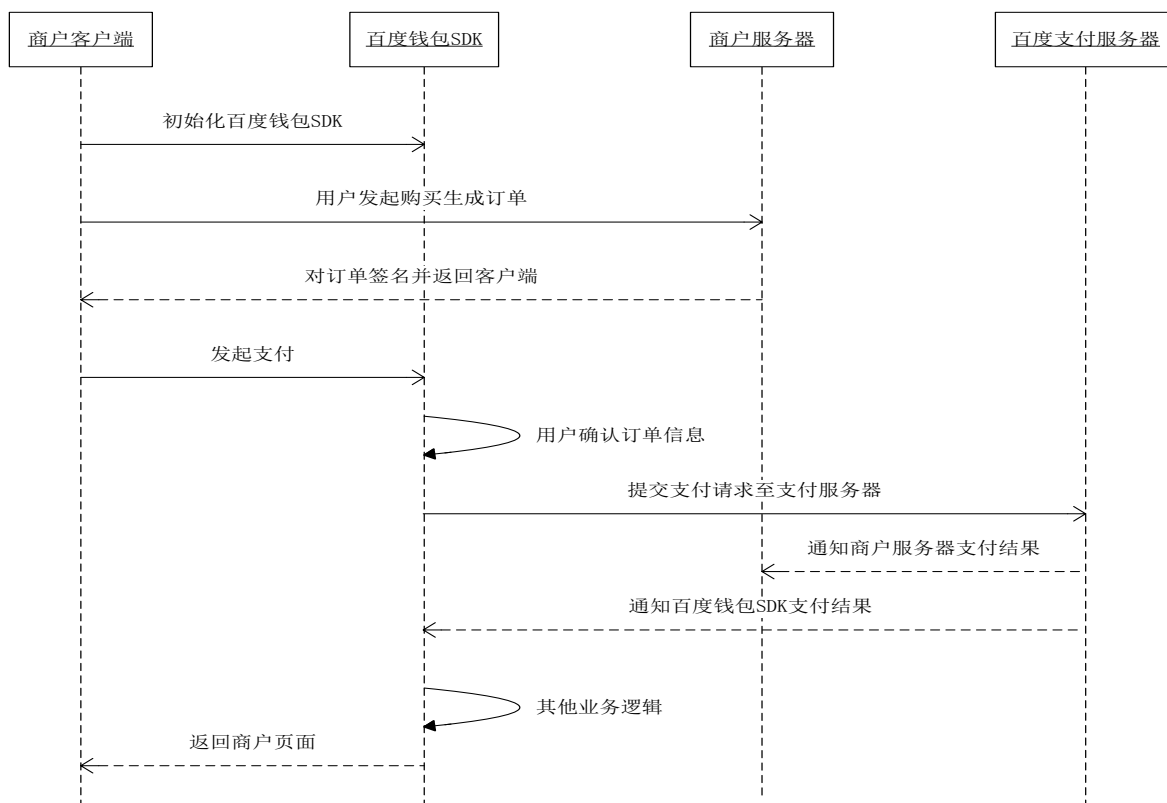
4.2 异步返回结果	17
4.3 主动绑卡返回结果	17
5 服务器通知支付结果	18
5.1 按订单号查询支付结果接口	18
5.2 支付结果通知接口	21
6 注意事项	23
6.1 重复的通知	23
6.2 查单接口	23
6.3 签名字符串和中文编码	23
6.4 百度钱包支付合作密钥的管理	24
7 附录	24
7.1 币种列表	24
7.2 摘要算法列表	24
7.3 支付结果列表	24
7.4 响应数据格式列表	25
7.5 字符编码列表	25
7.6 订单查询错误码列表	25

1 百度钱包简介

百度钱包 SDK 是提供给第三方的开发包，主要用来向其它的应用程序提供便捷、安全以及可靠的支付服务。同时提供一系列封装好的支付场景访问接口。

2 百度钱包支付集成

2.1 百度钱包支付业务流程



2.2 支付流程



2.3 百度钱包 SDKDemo 集成示例

为了便于商户的接入，我们提供了 demo。通过本 demo，商户可以提前了解百度钱包的功能及界面，同时还可参考 demo 的代码完成接入。

2.3.1 Demo 配置运行

步骤 1:

解压下载的百度钱包集成压缩包，进入目录，其中“BDWalletSdkDemo”即 demo 的项目文件，“BaiduWalletRefactorLib”是我们提供的百度钱包 sdk，将其导入 eclipse，步骤如图：

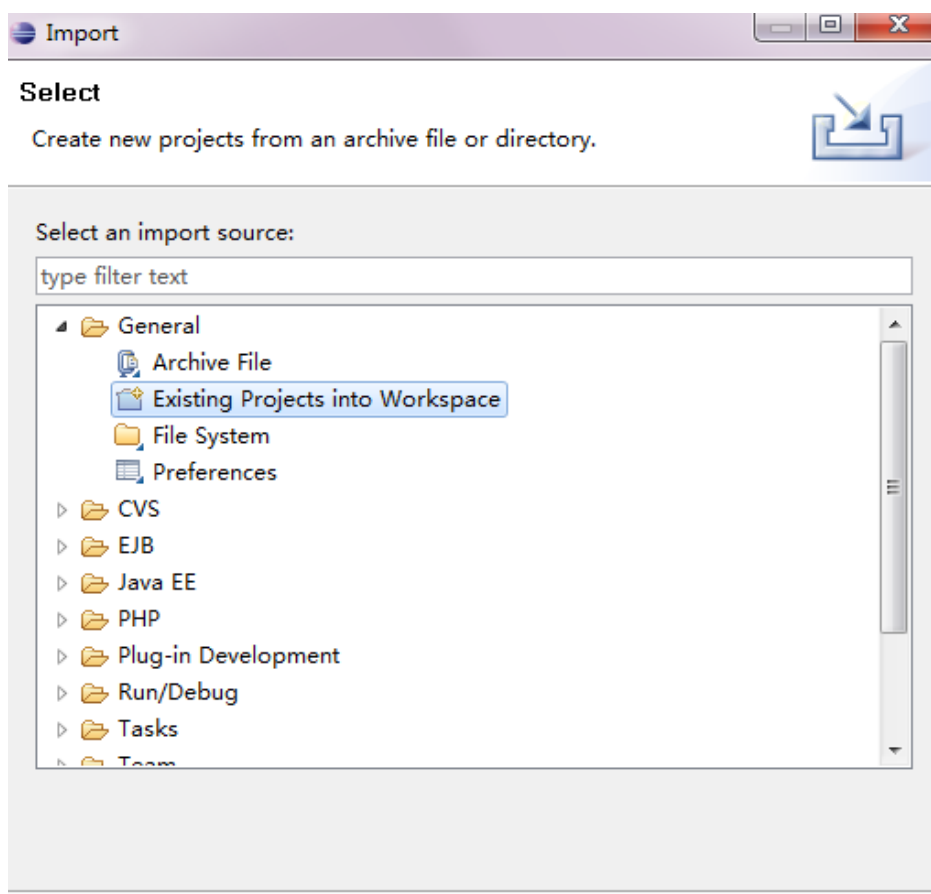


图 2- 1 Demo 导入示意图 1

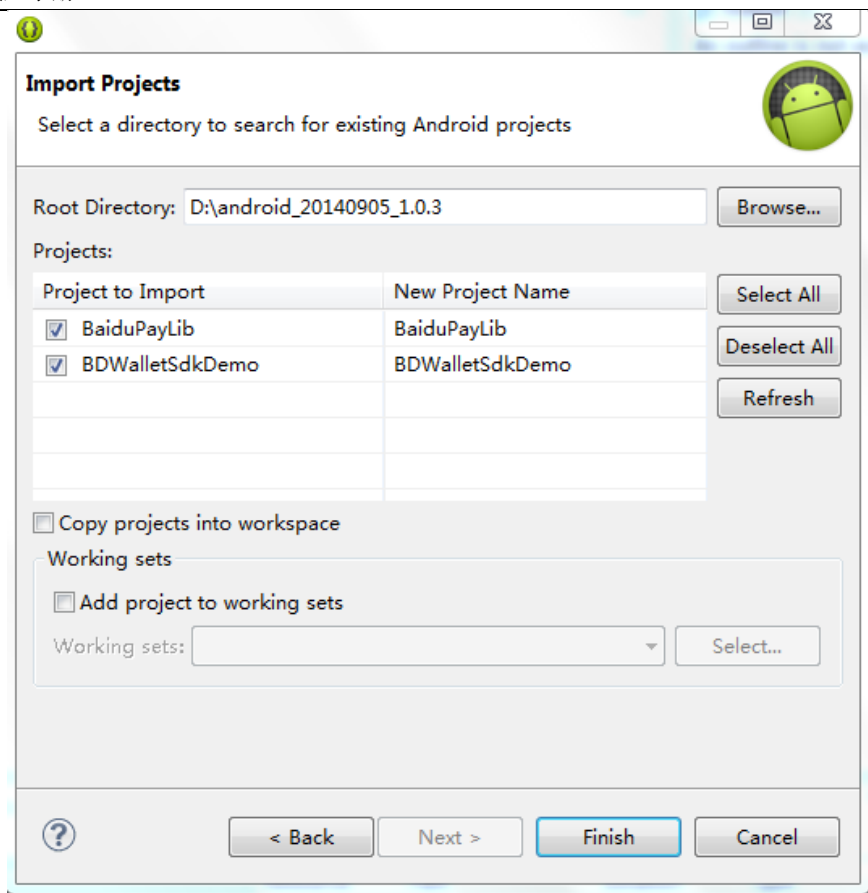


图 2- 2 Demo 导入示意图 2

BDWalletSdkDemo 引用 BaiduWalletRefactorLib 工程

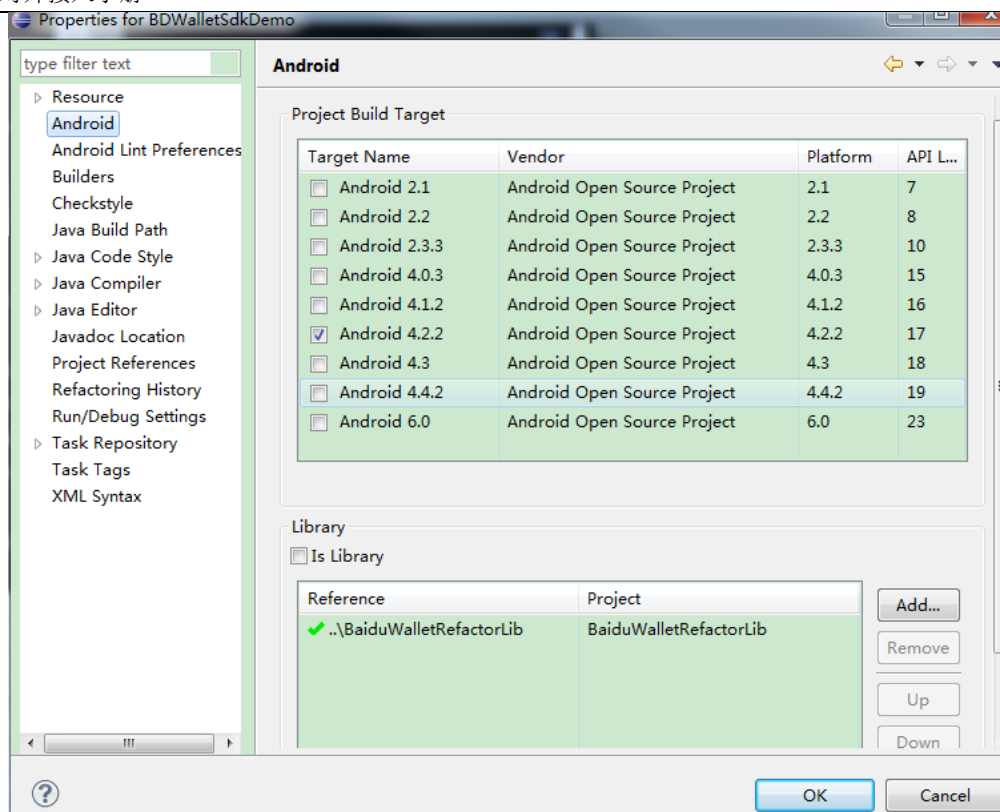


图 2- 3 Demo 导入示意图 3

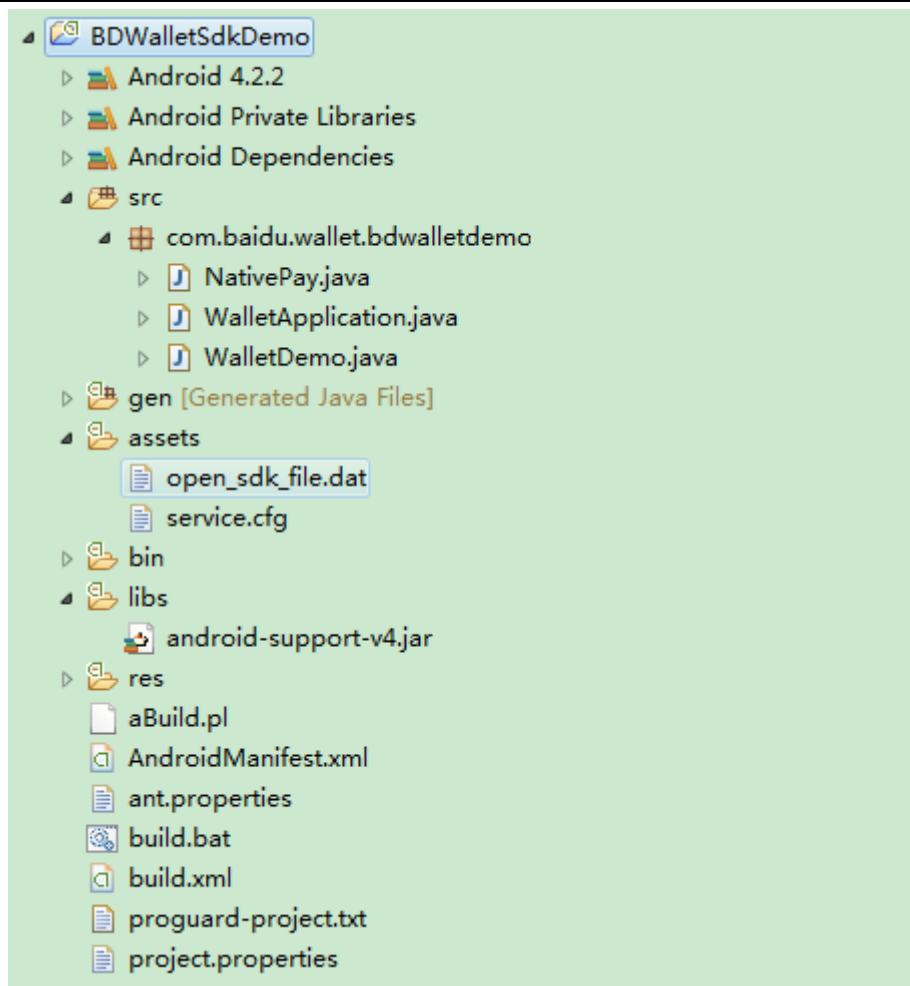
步骤 3：

在真机上运行项目将出现以下页面：



2.3.2 Demo 结构说明

项目结构如图：



注：需要 android-support-v4.jar

类名或文件名	说明
NativePay.java	提供调用百度钱包支付的实例
WalletApplication.java	应用程序
WalletDemo.java	主Activity
service.cfg	钱包首页配置文件
open_sdk_file.dat	Lib工程中so安全文件，防止so被替换

2.4 百度钱包 SDK 集成

2.4.1 接口说明

类名	描述
----	----

BaiduWallet	提供了和第三方交互的入口，是一个单例
--------------------	---------------------------

BaiduWallet 的方法

方法原形	initWallet (Context context)
方法功能	初始化钱包,这个接口是使用钱包 SDK 所有接口的先决条件，建议此接口在应用的 application.onCreate 中调用
方法参数	Context 上下文，传当前的 application 即可
返回值	无

方法原形	doPay (final Context context, final String orderInfo, final PayCallBack callback)
方法功能	支付接口
方法参数	Context 上下文对象 orderInfo 提供给支付的参数，长字符串，是由支付参数加上签名后拼接而成，详见 支付参数列表 章节 callback 支付结果回调接口，详见下表
返回值	无

接口名	描述
PayCallBack	支付后的回调接口，用来告诉宿主支付是否成功

PayCallBack 的接口：提供支付结果的回调

方法原形	onPayResult (int statusCode, String payDesc)
方法参数	int statusCode 状态码 参见 同步返回结果章节 String payDesc 参见 同步返回结果章节
返回值	无

方法原形	<code>boolean isHideLoadingDialog()</code>
方法参数	无
返回值	是否隐藏启动 sdk 时显示的 loading dialog ,该方法已过时 ,宿主返回 true 即可

方法原形	<code>doBind(final Context context, final BindBack bindback, final Map<String, String> params)</code>
方法功能	主动绑卡方法
方法参数	Context 上下文对象 bindback 绑卡接口回调接口, 详见下表 params 提供给绑卡的参数, 详见 主动绑卡参数 章节
返回值	无

接口名	描述
BindBack	绑卡认证回调的接口

BindBack 的接口: 提供绑卡结果的回调

方法原形	<code>onBindResult(int statusCode, String params)</code>
方法参数	int statusCode 状态码 0 表示成功 String params 参见 主动绑卡返回结果 章节
返回值	无

2.4.2 接入步骤

商户将自己的工程当做 demo 来做即可。

本章指导在商户项目中集成百度钱包步骤, 关键代码以 Demo 为例:

步骤 1: 引用BaiduPayLib工程

商户项目要引用 “BaiduPayLib” 工程, 如图 2- 3

步骤 2: 添加open_sdk_file.dat

copy assets目录下相关文件

- 添加open_sdk_file.dat

添加demo中的assets目录中open_sdk_file.dat，添加到商户项目的assets目录

- 银行卡识别相关

Copy demo中的assets目录中的bankcard文件夹到商户项目的assets目录中

步骤 3：修改商户项目的manifest.xml文件，详见demo的manifest.xml

添加权限：

```
<uses-permission android:name="android.permission.INTERNET" />
<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" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission
    android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.hardware.camera" />
<uses-permission android:name="android.permission.FLASHLIGHT" />
<uses-permission android:name="org.fidoalliance.uaf.permissions.FIDO_CLIENT" />
<uses-permission android:name="android.permission.USE_FINGERPRINT"/>
<uses-permission
    android:name="com.fingerprints.service.ACCESS_FINGERPRINT_MANAGER"/>
<uses-permission android:name="com.sec.feature.fingerprint_manager_service"/>

<!-- 设备特性声明 -->
<uses-feature android:name="android.hardware.camera" android:required="false"
/>
<uses-feature android:name="android.hardware.camera.front"
    android:required="false" />
<uses-feature android:name="android.hardware.camera.autofocus"
    android:required="false" />
<uses-feature android:name="android.hardware.camera.flash"
    android:required="false" />

添加Activity：
<!-- 以下需要商户添加activity start-->

<activity
    android:name="com.baidu.wallet.passport.PassLoginActivity"
    android:configChanges="keyboardHidden/navigation/orientation/screenSize"
```

```

        android:exported="@bool/bd_wallet_switch_global_debug"
        android:hardwareAccelerated="false"
        android:theme="@style/EbpayThemeActivitTranslucent"
        android:windowSoftInputMode="stateHidden" >
<activity
    android:name="com.baidu.paysdk.ui.WelcomeActivity"
    android:configChanges="keyboardHidden/navigation/orientation/screenSize"
    android:excludeFromRecents="true"
    android:exported="@bool/bd_wallet_switch_global_debug"
    android:theme="@style/EbpayThemeActivityWelcome"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="stateHidden" />
<!-- 银行卡检测 -->
<activity
    android:name="com.baidu.wallet.bankdetection.BankCardDetectionActivity"
    android:label="@string/app_name"
    android:launchMode="singleTop"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen" />
    <activity
        android:name="com.baidu.wallet.bankdetection.BankCardResultActivity"
        android:label="@string/app_name"
        android:screenOrientation="portrait"
        android:theme="@style/CameraMist" />
<!-- 指纹支付 -->
    <activity android:name="com.Lenovo.fido.framework.IntentHelperActivity"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"/>

```

由于篇幅，此处省略... 具体请参照demo中的manifest文件

```

<activity
    android:name="com.baidu.scancode.ui.ScanCodePwdPayActivity"
    android:configChanges="keyboardHidden/navigation/orientation/screenSize"
    android:excludeFromRecents="true"
    android:exported="@bool/bd_wallet_switch_global_debug"
    android:theme="@style/EbpayThemeActivitTranslucent"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="stateVisible/adjustResize" />

<!-- 商户添加activity end -->

```

步骤4：修改商户项目的proguard文件，请加入以下内容

以下是推荐的配置，供参考使用

```
# baiduwallet start
```

```
-keep interface com.baidu.wallet.core.NoProguard
-keep public class * implements com.baidu.wallet.core.NoProguard {
    public protected *;
}
-keep class com.baidu.android.lbspay.** { *; }
-keep class com.baidu.android.pay.** { *; }
-keep class com.baidu.balance.** { *; }
-keep class com.baidu.fastpay.** { *; }
#如果接入nfc需要增加nfc的混淆配置 nfc start
-keep class com.baidu.nfc.** { *; }
#nfc end
-keep class com.baidu.traffic.** { *; }
-keep class com.baidu.scancode.** { *; }
-keep class com.baidu.apollon.**{*;}
-keep class com.baidu.wallet.remotepay.**{*;}
-keep class com.baidu.home.** { *; }
-keep class com.baidu.paysdk.** { *; }
-keep class com.baidu.personal.** { *; }
-keep class com.baidu.seclab.sps.** { *; }
-keep class com.baidu.transfer.** { *; }
-keep class com.baidu.wallet.** { *; }
-keep class com.baidu.BankCardProcessing
-keep class com.baidu.BCResult
-dontwarn com.baidu.searchbox.plugin.api.**
# passsdk start
-keep class com.baidu.sapi2.** { *; }
-keepattributes JavascriptInterface
-keepattributes *Annotation*
#passsdk end

#bankcard start
-keep class com.baidu.bankdetection.** { *; }
#bankcard end

#fingerprint start
-keep class com.lenovo.appsdk.** { *; }
-keep class com.lenovo.fido.** { *; }
-keep class com.samsung.android.sdk.** { *; }
```

#fingerprint end

baiduwallet end

步骤 5：初始化百度钱包

在调用钱包任何功能之前，必须初始化钱包，而且初始化必须放在UI线程里面。**强烈建议放到应用的Application.onCreate里面**，请参照 demo 的WalletApplication.java

```
BaiduWallet.getInstance().initWallet(this);
```

2.5 支付参数列表

下表的参数是供 doPay 调用时所需的参数，需要经过拼接签名后传递给 [BaiduWallet.doPay](#) 接口中的 orderInfo 参数。

参数名	参数含义	格式说明	长度	是否必须	是否参与签名
service_code	服务编号	整数，目前必须为 1		是	是
sp_no	百度钱包商户号	10 位数字组成的字符串		是	是
order_create_time	创建订单的时间	YYYYMMDDHHMMSS		是	是
order_no	订单号，商户须保证订单号在商户系统内部唯一。	不超过 20 个字符	Max(20)	是	是
goods_category	商品分类号。与优惠券相关。在创建优惠券推广计划的时候，可以指定优惠券使用的商品类型。若订单的类型与优惠券的类型一致时，就会出相应的优惠券。没有指定时，出用户可用的全部优惠券	取值由钱包系统分配		否	是
goods_channel_sp	数字商品开发商	10 位数字组成的字符串	10	否	否

goods_channel	数字商品渠道	商户与渠道商提前约定好，字符串，字母和数字的组合，不能包含其他特殊字符，不超过 20 为字符串	Max(20)	否	否
goods_name	商品的名称	允许包含中文 ；不超过 128 个字符或 64 个汉字	Max(128)	是	是
goods_desc	商品的描述信息	允许包含中文 ；不超过 255 个字符或 127 个汉字	Max(255)	是	是
goods_url	商品在商户网站上的 URL。	仅支持 http(s)的 URL	Max(255)	否	是
unit_amount	商品单价，以分为单位	非负整数		否	是
unit_count	商品数量	非负整数		否	是
transport_amount	运费	非负整数		否	是
total_amount	总金额，以分为单位	非负整数		是	是
currency	币种，默认人民币	取值范围参见附录		是	是
buyer_sp_username	买家在商户网站的用户名	允许包含中文 ；不超过 64 字符或 32 个汉字	Max(64)	否	是
return_url	百度钱包主动通知商户支付结果的 URL	仅支持 http(s)的 URL	Max(255)	是	是
pay_type	支付方式	默认取值 2		是	是
sp_pass_through	商户定制服务字段	商户定制服务字段 如果接返现则为固定值：%7B%22offline_pay%22%3A1%7D	Max(255)	否	是
expire_time	交易的超时时间	YYYYMMDDHHMMSS，不得早于交易创建的时间。		否	是
input_charset	请求参数的字符编码	取值范围参见附录		是	是
version	接口的版本号	必须为 2		是	是
sign	签名结果	取决于签名方法，详见 签名机制		是	是
sign_method	签名方法	取值范围参见附录		是	是
extra	商户自定义数据	不超过 255 个字符	Max(255)	否	是

说明：

1. 商户通过上述参数拼成订单信息。如果只有 1 个商品，那么 **goods_desc** 和 **goods_url** 可以使该商品的名称和展现 URL；如果包括多个商品，那么 **goods_desc** 和 **goods_url** 不

可能与每个商品一一对应，具体内容商户定义。

2. 该接口“可能包含中文”的参数包括：**goods_name**、**goods_desc**、**buyer_sp_username**，因此指定了 **input_charset** 参数。这三个参数值必须按 **input_charset** 编码后，才能参与签名。
3. **goods_name** 和 **goods_desc** 同样都是商品的相关信息。区别在于，**goods_name** 是商品名称，比较短；**goods_desc** 是商品的描述，比较长。如果商户觉得 **goods_name** 不足以说明商品的信息，则可以使用 **goods_desc** 字段。
4. **total_amount** 的值是必须提供的；**unit_amount**、**unit_count** 和 **transport_amount** 的值或者都提供，或者都不提供。如果这 4 个参数都提供了，那么必须满足下面等式：
total_amount = unit_amount*unit_count+transport_amount
否则，百度钱包支付将拒绝该支付请求。
5. **buyer_sp_username** 是买家在商户网站的用户名。如果商户需要在百度钱包支付的交易记录中保存该信息，则可以使用这个参数。
6. **return_url** 用于后台通知。**return_url** 的内容包括必要的文本信息即可。这个 URL 不应该带有参数。**需要注意的是，return_url 必须返回 200，不能要求登录或重定向。**
7. **extra** 完全由商户自己使用，对百度钱包支付是透明的。如果提供了该参数，那么查单接口和通知接口将原样返回该参数。

每个参数以 **key=value** 的形式拼接，参数之间用**&**相连，请注意需要签名的参数，签名规则请

查阅[签名机制](#)章节，参数经过签名拼接后再加上签名 **sign** 值的示例如下：

orderInfo示例：

```
service_code=1&sp_no=1500000001&order_create_time=20151209213048&order_no=20151209213048905379&goods_name=%CE%D2%B5%C4%C9%CC%C6%B7&goods_desc=&goods_url=&unit_amount=&unit_count=&transport_amount=&total_amount=1&currency=1&buyer_sp_username=&return_url=http%3A%2F%2Fdb-testing-eb07.db01.baidu.com%3A8666%2Fsuccess.html&page_url=&pay_type=2&expire_time=20151211213048&input_charset=1&version=2&sign_method=1&extra=&sign=2b8256ebee3fb051ea2168d07bdf9006
```

以上字符串对应 [BaiduWallet.doPay](#) 接口中的**orderInfo**参数，请宿主按照要求对相应的参数进行拼接签名后传入**doPay**接口中。

2.6 主动绑卡参数列表

下表的参数是供 **doBind** 调用时所需的参数，需要经过拼接签名后传递给 [BaiduWallet.doBind](#)

接口中的 **orderInfo** 参数。

参数名	参数含义	格式说明	是否必须
sp_no	商户号	10位数字，由百度钱包分配，同支付商户号相同	是
version	版本	1	是
activity_no	活动id	字符串，最大20个字符	是
order_no	流水号	3-20位，商户保证不能重复	是
return_url	主动通知绑卡结果的URL	仅支持http和https的URL	是
sign	签名结果	取决于签名方法，具体参考签名机制	是
sign_method	签名方法	取值范围参见附录	是

3 签名机制

由于商户和百度钱包支付之间的通信涉及到订单信息和支付信息，必须保证通信数据不被篡改和伪造。否则，将给商户和百度安全支付造成资金损失。

百度钱包支付采用**签名机制**来保证通信安全。本文中的每个接口规范都包括两个参数：sign_method 和 sign。sign_method 是摘要算法，sign 是签名结果。商户或百度钱包支付向对方发送数据时，必须指定 sign_method 并生成相应的 sign；对方接收到数据后，必须使用相同的签名算法对 sign 的值进行验证。如果验证不通过，则说明通信数据已经被篡改或伪造。

3.1 签名原理

百度钱包支付没有使用 RSA、DSA 等基于非对称密钥的签名算法，而是使用了 MD5、SHA-1 等摘要算法。这些摘要算法本身并不能用作签名，但是结合百度钱包支付合作密钥，也可以起到签名的作用，进而达到**防篡改和伪造**的目的。

商户与百度钱包支付签约时，百度钱包会分配一个百度钱包支付合作密钥。**每个商户的密钥是唯一的，一定不能让第三方知道。如果密钥泄漏，必须及时通知百度钱包更换密钥。**

需要签名的参数请查看支付参数列表。

3.2 签名步骤

签名步骤包括拼接待签名数据和对待签名数据进行摘要两个步骤：

步骤一：待签名数据由除 **sign** 和**不需要参与签名的参数**之外的所有请求参数和百度钱包支付合作密钥按以下规则拼接而成：

- 请求参数都按照名称**字符升序**排列（参数名称不允许相同）
- 某些请求参数的值是允许包含中文的，为了避免中文的编码问题，我们规定**所有参数**的值必须按照 input_charset 进行编码

- c) 对于可选参数（接口规范中的非“必须”参数），如果没有使用，则无需参与拼接。这样可以增强接口参数的可扩展性。
- d) 将百度钱包支付合作密钥作为最后一个参数，参数名为 key，参数值就是百度钱包支付合作密钥本身
- e) 将请求参数按上述顺序用&拼接起来

需要注意的是：在签名时，如果参数的值包括&、@等特殊字符或中文，这些字符需要保持原样，不要做 URL 编码。

步骤二：用指定的摘要算法对待签名数据进行摘要，生成签名数据（不区分大小写），作为附加的请求参数 sign 的值。

下面是一个使用 MD5 算法进行签名的示例：

```
sign =  
MD5(currency=1&extra=&goods_category=1&goods_channel=baidu&goods_channel_  
sp=0001&goods_desc= 使用百度钱包支付支付的订单 &goods_name= 商品  
1&goods_url=http://bj.nuomi.com/364&input_charset=1&order_create_time=201305  
08131702&order_no=1372323335119&pay_type=2&return_url=http://item.jd.com/73  
6610.html&service_code=1&sign_method=1&sp_no=1210010002&total_amount=1&tr  
ansport_amount=0&unit_amount=1&unit_count=1&key=XXXXXXXXXXXXXXXXXX)  
= 39375042FD5F801C2B0A128B145589C3
```

其中,goods_name 和 goods_desc 的值(加粗部分)应该是按照 input_charset 编码后的结果。

下面解释一下为什么签名之前，需要对中文进行特定的编码。

相同的字符串(包括中文)，如果内部编码格式不同，那么对应的字节流可能也不相同。MD5 等摘要算法是对字节流进行操作的。因此，相同字符串的摘要结果未必相同，取决于内部编码。

商户和百度钱包必须协商好中文的编码，才能成功验签。如果商户的请求参数中可能包含中文，那么必须指定 input_charset 参数，并按照 input_charset 进行编码后，才能签名。百度钱包收到请求数据后，使用相同的编码进行验签。

4 通知结果

4.1 同步返回结果

百度钱包对商户的请求数据处理完成后，会在 [onPayResult](#) 回调接口中将处理的结果数据直接通知给商户客户端（对应 payDesc 参数）。结果格式如下：

statecode={状态码};order_no={商户传入的订单号};notify={订单签名}；

结果信息详细描述如下：

字段名称	描述	属性	备注
stateCode	状态码	整数	0：成功 1：支付中 2：取消
order_no	订单号	字符串	商户传入的订单号
notify	结果数据	字符串	订单签名

payDesc 的示例如下：

```
statecode={0};order_no={1372852640712};notify="
currency=1&extra=&order_no=1372852640712&pay_result=1&pay_time=20130703200113&pay
_type=2&sign_method=1&sp_no=1210010002&total_amount=1&transport_amount=0&unit_a
mount=1&unit_count=1&sign=b3e35d180b747d5302d5ccbab6410c53";
```

4.2 异步返回结果

在支付完成之后，支付结果会通过服务器主动通知给商户服务器，这就是异步通知，通知地址是商户传入的 **return_url**。商户最终支付结果以商户服务器的异步通知为准。

4.3 主动绑卡返回结果

当绑卡完成之后，会在 onBindResult 回调接口中将处理的结果数据直接通知给商户客户端（对应 params 参数）。结果格式如下：

statecode={状态码};order_no={商户传入的订单号};notify={订单签名}；

认证信息返回参数结果，详细描述如下：

参数名	参数含义	格式说明	是否必须
statusCode	状态码	整数	0：绑卡成功 -10：参数错误 2：绑卡取消
params	结果数据	字符串，绑卡状态详细数据，包含状态码，订单号，绑卡结果等信息的字符串例如： statecode={x};order_no={xxx}	包含 (activity_code:0、可以参加活动

		};notify={http://www.baidu.com?activity_code=1&activity_no=1001&order_no=1446793174&page_url=http://www.baidu.com&return_url=http://www.baidu.com&sign_method=1&sp_no=1500000001&version=1&sign=393ab9bbca8c60f673d761e0d86d15d6};) 等信息, 详细说明参见后端订单查询文档
--	--	--	-----------------------

5 服务器通知支付结果

如果由于种种原因（网络问题、商户网站临时维护等）百度钱包支付的通知不能达到商户，那么商户就不能及时得到订单的支付状态。这在一定程度上会影响商户网站的用户体验。例如，用户支付成功后，去商户网站查看订单的支付状态，却发现该订单“尚未支付”。

在这种情况下，商户可以主动向百度钱包支付查询订单的支付结果。目前，百度钱包支付只支持“按订单号”的查询条件。

5.1 按订单号查询支付结果接口

URL : https://www.baifubao.com/api/0/query/0/pay_result_by_order_no

请求方式：GET

请求参数：

参数名	参数含义	格式说明	长度	是否必须
service_code	服务编号	整数，目前必须为 11		是
sp_no	百度钱包支付商户号	10 位数字组成的字符串		是
order_no	订单号	不超过 20 个字符	Max(20)	是
output_type	响应数据的格式，默认 XML	取值范围参见附录		是
output_charset	响应数据的字符编码，默认 GBK	取值范围参见附录		是
version	接口的版本号	必须为 2		是
sign	签名结果	取决于签名方法		是

sign_method	签名方法，默认 MD5	取值范围参见附录		是
-------------	-------------	----------	--	---

响应数据可以支持多种格式和编码，由 output_type 和 output_charset 参数指定。

响应数据：

参数名	参数含义	格式说明	长度	是否必须
query_status	订单查询的错误码	取值范围参见附录		是
sp_no	百度钱包支付商户号	10 位数字组成的字符串		是
order_no	订单号	不超过 20 个字符	Max(20)	是
bfb_order_no	百度钱包支付交易号	不超过 30 个字符	Max(30)	否；交易存在时才会有
bfb_order_create_time	百度钱包支付交易创建时间	YYYYMMDDHHMMSS		否；交易存在时才会有
pay_time	支付时间	YYYYMMDDHHMMSS		否；用户成功付过款才会有
pay_type	支付方式	默认取值 2		否；用户成功付过款才会有
goods_name	商品的名称	允许包含中文；不超过 128 个字符或 64 个汉字	Max(128)	是
unit_amount	商品单价，以分为单位	非负整数		否
unit_count	商品数量	非负整数		否
transport_amount	运费，以分为单位	非负整数		否
total_amount	总金额，以分为单位	非负整数		是
cash_amount	交易现金金额，以分为单位	非负整数		是
fee_amount	手续费，以分为单位	非负整数		是
buyer_sp_username	买家在商户网站的用户名	允许包含中文；不超过 64 字符或 32 个汉字	Max(64)	否
currency	币种	取值范围参见附录		是
pay_result	支付结果代码	取值范围参见附录		是
sign	签名结果	取决于签名方法		是
sign_method	签名方法	取值范围参见附录		是
extra	商户自定义数据	不超过 255 个字符	Max(255)	否

示例：

商户的请求参数：

参数名	参数值	参数值的说明
service_code	11	表示按订单号查询支付结果

sp_no	1234567890	百度钱包支付商户号
order_no	20080808123456123456	待查询的订单号
output_type	1	响应数据的格式为 XML
output_charset	1	响应数据的编码是 GBK
version	2	版本号是 2
sign	5C7E1DBAC2C40764D9D00678D42B45C0	签名结果
sign_method	1	签名算法为 MD5

签名：

```
sign =
MD5(order_no=20080808123456123456&output_charset=1&output_type=1&service_c
ode=11
&sign_method=1&sp_no=1234567890&version=2&key=XXXXXXXXXXXXXXXXXX)
=
5C7E1DBAC2C40764D9D00678D42B45C0
```

发送的 URL:

```
https://www.baifubao.com/api/0/query/0/pay_result_by_order_no?order_no=12345678
902008080808080812345671234567&output_type=1&output_charset=1&service_cod
e=11&sign_method=1&sp_no=1234567890&version=2&sign=5C7E1DBAC2C40764D
9D00678D42B45C0
```

响应数据(XML 格式)：

```
<?xml version="1.0" encoding="GBK" ?>
<response>
  <query_status>0</query_status>
  <sign>7D6C167DCE83AD23AD22851792D77F65</sign>
  <sign_method>1</sign_method>
  <sp_no>1234567890</sp_no>
  <order_no>20080808123456123456</order_no>
  <bfb_order_no>20080808BFB20080808123456123456</bfb_order_no>
  <bfb_order_creator_time>20080808080808</bfb_order_creator_time>
  <cash_amount>2500</cash_amount>
  <pay_time>20080808090909</pay_time>
  <pay_type>2</pay_type>
  <bank_no>201</bank_no>
  <goods_name>使用百度钱包支付支付的商品</goods_name>
  <unit_amount>1000</unit_amount>
  <unit_count>2</unit_count>
  <transport_amount>500</transport_amount>
  <total_amount>2500</total_amount>
  <fee_amount>0</fee_amount>
  <currency>1</currency>
  <buyer_sp_username>jarfield</buyer_sp_username>
  <pay_result>1</pay_result>
</response>
```

其中，

```
sign =
MD5(bank_no=201&bfb_order_create_time=20080808080808&bfb_order_no=2008
0808BFB20080808123456123456&buyer_sp_username=jarfield&currency=1&fee_amo
```



```
unt=o&goods_name= 使用百度钱包支付支付的商
品
&order_no=20080808123456123456&pay_result=3&pay_time=20080808090909&pay
_type=2&query_status=0&sign_method=1&sp_no=1234567890&total_amount=2500&
transport_amount=500&unit_amount=1000&unit_count=2&key=XXXXXXXXXXXXX
XXX)
=
7D6C167DCE83AD23AD22851792D77F65
query_status 表示查询的错误码，参见附录。
```

5.2 支付结果通知接口

URL：商户提供的 **return_url**

请求方式：GET

请求参数：

参数名	参数含义	格式说明	长度	是否必须
sp_no	百度钱包支付商户号	10 位数字组成的字符串		是
order_no	商户订单号	不超过 20 个字符	Max(20)	是
bfb_order_no	百度钱包支付交易号	不超过 32 个字符	Max(32)	是
bfb_order_create_time	百度钱包支付交易创建时间	YYYYMMDDHHMMSS		是
pay_time	支付时间	YYYYMMDDHHMMSS		是
pay_type	支付方式	默认取值 2		是
unit_amount	商品单价，以分为单位	非负整数		否
unit_count	商品数量	非负整数		否
transport_amount	运费，以分为单位	非负整数		否
total_amount	总金额，以分为单位	非负整数		是
fee_amount	百度钱包支付收取商户的手续费，以分为单位	非负整数		是
currency	币种，目前仅支持人民币	取值范围参见附录		是
buyer_sp_username	买家在商户网站的用户名	允许包含中文；不超过 64 字符或 32 个汉字	Max(64)	否
pay_result	支付结果代码	取值范围参见附录		是
input_charset	请求参数的字符编码	取值范围参见附录		是
version	版本号	与支付请求中的 version 保持一致		是
sign	签名结果	取决于签名方法		是
sign_method	签名方法	取值范围参见附录		是
extra	商户自定义数据	不超过 255 个字符	Max(255)	否

说明:

1. 由于 **buyer_sp_username** 可能包含中文，因此必须传 **input_charset** 参数。
Input_charset 是必须参数，即使没有传 **buyer_sp_username**，也必须传 **input_charset**。

响应数据：

接收到百度钱包支付的后台通知后，商户须返回特定的 HTML 页面。该页面应该满足以下要求：

HTML 头部须包括 `<meta name="VIP_BFB_PAYMENT" content="BAIFUBAO">`

百度钱包支付只有检测到该字符串，才会认为商户已经成功接收到通知、已经验证过并且认可通知的内容。

示例：**通知参数：**

参数名	参数值	参数值的说明
sp_no	1234567890	10 位数字组成的字符串
order_no	20080808123456123456	订单号
bfb_order_no	20080808BFB20080808123456123456	百度钱包支付交易号
bfb_order_create_time	20080808080808	百度钱包支付交易创建时间
pay_time	20080808090909	支付时间
pay_type	2	默认支付方式
unit_amount	1000	商品单价为 10 元
unit_count	2	订单包含 2 件相同的商品
transport_amount	500	运费为 5 元
total_amount	2500	买家需要支付 25 元
fee_amount	0	手续费 0 元
currency	1	支付的币种是人民币
buyer_sp_username	jarfield	卖家在商户网站的用户名
pay_result	1	支付成功
input_charset	1	中文编码是 GBK
version	2	版本号为 2
sign	B219D1A2784C1F12868FEE887374AFB2	签名结果
sign_method	1	签名算法为 MD5

签名：

```
sign =
MD5(bank_no=201&bfb_order_create_time=20080808080808&bfb_order_no=20080808BFB20080808123456123456&buyer_sp_username=jarfield&currency=1&fee_amount=0&input_charset=1&order_no=20080808123456123456&pay_result=1&pay_time=20080808090909&pay_type=2&sign_method=1&sp_no=1234567890&total_amount=2500&trans
```

```
port_amount=500&unit_amount=1000&unit_count=2&version=2&key=XXXXXXXXXXXXX
XXXX)
=
B219D1A2784C1F12868FEE887374AFB2
```

6 注意事项

在商户接入的过程中，有一些细节需要额外注意。如果忽略了这些细节，可能会延迟联调进度，更有可能导致资金问题。这些细节在功能概述、交互流程和接口规范这三节中都有重点描述，这里再汇总和强调一次。

6.1 重复的通知

对于支付系统，后台通知有可能被重复发送，甚至同时发送。如果由于网络问题或商户系统维护，百度钱包支付没有收到商户对通知的正确响应，那么百度钱包支付会尝试重发通知。

总之，通知到达商户系统的次数和时间都是不可预知。商户系统不能依赖于任何假设来保证系统的正确性。商户的系统必须对通知进行除重处理，对于同一笔订单的通知，必须只处理第一次通知，忽略后续的通知。特别是涉及到资金的操作，须额外谨慎。

百度钱包支付系统在通知成功的情况下，尽量不重复发送后台通知。但是为了避免掉单，重复发送后台通知的情况是不可避免的。

6.2 查单接口

发现很多商户都不使用查单接口。其实这个接口的作用还是比较大的，建议大家使用。目前，有两个场景很适合使用查单接口：

1. 处理支付结果通知。为了防止第三方伪造通知，商户系统在收到通知后立刻通过查单接口反查交易的支付结果。
2. 处理掉单情况。由于网络、系统升级等不可避免的因素，支付结果通知是有可能丢失的。百度钱包支付会加强通知机制，直到通知成功为止。但是，如果商户系统可以主动查单，那么基本可以杜绝掉单情况，并且在最短的时间内修复交易的状态，从而提高用户体验。

6.3 签名字符串和中文编码

在商户接入的过程中，出现最多的问题就是“验签失败”。这个问题的原因往往有两个：

1. 签名字符串拼接方式的不对
2. 签名字符串中，中文参数没有使用正确的中文编码

签名字符串的拼接方式在 **签名机制** 中已经详细描述。但是，关于可选参数的拼接，还有有可能混淆。下面重点介绍一下。

假设参数 A 是必选的，其值为 value_a；参数 B 是可选的，而且没有值；没有其他参数。那么，签名字符串有两种可能：

$A=value_a\&B=$ 或 $A=value_a$

在签名时，对于可选参数，到底采用哪种拼接方式，取决与 HTTP 请求的参数。如果请求参数中包含 B，但是没有值，那么就采用 $A=value_a\&B=$ 的方式。如果请求参数中不包含 B，那么就采用 $A=value_a$ 的方式。

在验签时，对于可选参数，到底采用哪种拼接方式，取决与 HTTP 响应的参数。如果响应参数中包含 B，但是没有值，那么就采用 $A=value_a\&B=$ 的方式。如果响应参数中不包含 B，那么就采用 $A=value_a$ 的方式。

特别需要注意的是，考虑到接口的兼容性，商户系统需要特别注意签名和验签逻辑的编码方式。建议根据 HTTP 请求或响应中存在的参数来拼接签名字符串，**而不要将包含在签名字符串中的参数写死**。因为百度钱包支付以后可能会增加可选参数，如果硬编码，那么可能导致验签失败。

6.4 百度钱包支付合作密钥的管理

百度钱包支付合作密钥是百度钱包支付与商户系统交互安全的基础。如果密钥丢失，则可能发生伪造交易、通知等行为，进而引发资金风险。因此，**商户须严格保护百度钱包支付合作密钥，不能让第三方知道**。

在商户公司内部，知道百度钱包支付合作密钥的员工越少越好。如果相关的员工离职，最好联系百度钱包支付更换密钥。

7 附录

7.1 币种列表

取值	含义
1	人民币

7.2 摘要算法列表

取值	含义
1	MD5
2	SHA-1

7.3 支付结果列表

取值	含义
1	支付成功

2	等待支付
3	退款成功

7.4 响应数据格式列表

取值	含义
1	XML

7.5 字符编码列表

取值	含义
1	GBK

7.6 订单查询错误码列表

取值	含义
0	查询到结果
1002	查询结果为空
5801	缺少请求参数
5802	请求参数非法
5803	不支持的签名算法
5804	验签失败
5806	服务器内部错误