

# 中国银行

## 支付宝 APP 支付接口

版本号：1.0.0

版本记录：

编号	时间	版本号	修改说明
1	2017.03.22	V1.0.0	✧ 接口文档的创建

# 目 录

<b>1. 扫码支付简介.....</b>	<b>1</b>
1.1. 应用场景.....	1
1.2. 业务流程.....	1
1.3. 使用步骤.....	1
1.4. 接口调用时序图.....	2
1.5. 生成二维码规则.....	错误！未定义书签。
<b>2. 接口说明.....</b>	<b>6</b>
2.1. 接口调用规则.....	6
2.2. SIGN 签名生成方法.....	7
2.3. 代码实例.....	8
<b>3. API 列表.....</b>	<b>9</b>
3.1. 交易预创建接口.....	9
3.1.1 应用说明: .....	9
3.1.2 接口地址: .....	9
3.1.3 输入参数: .....	9
3.1.4 返回结果: .....	11
3.1.5 错误码: .....	12
3.2. 支付结果通知接口.....	14
3.2.1 应用说明: .....	14
3.2.2 接口地址: .....	15
3.2.3 通知参数: .....	15
3.2.4 返回结果: .....	17
3.3 查询订单接口.....	18
3.3.1 应用说明: .....	18
3.3.2 接口地址: .....	18
3.3.3 输入参数: .....	18
3.3.4 返回结果: .....	19
3.3.5 错误码: .....	21
3.4. 撤销订单接口.....	21

3.4.1 应用说明:	21
3.4.2 接口地址:	22
3.4.3 输入参数:	22
3.4.4 返回结果:	22
3.4.5 错误码:	23
3.5 申请退款接口	24
3.5.1 应用说明:	24
3.5.2 接口地址:	25
3.5.3 输入参数:	25
3.5.4 返回结果:	26
3.5.5 错误码:	27
3.6 查询退款接口	28
3.6.1 应用说明:	28
3.6.2 接口地址:	28
3.6.3 输入参数:	29
3.6.4 返回结果:	29
3.6.5 错误码:	31
3.7 对账单接口	31
3.7.1 应用说明:	31
3.7.2 接口地址:	32
3.7.3 输入参数:	32
3.7.4 返回结果:	32
3.7.5 错误码:	33
4. 公共错误码	34

# 1. APP 支付简介

## 1.1. 应用场景

适用于商家在 App 应用中集成支付宝支付功能。

商家 APP 调用支付宝提供的 SDK 调用支付宝客户端内的支付模块，商家 APP 会跳转到支付宝中完成支付，支付完后跳回到商家 APP 内，最后展示支付结果。

目前支持手机系统有：iOS（苹果）、Android（安卓）。

## 1.2. 业务流程

步骤 1：用户在商家 App 中选择商品下单、确认购买，进入支付环节，选择支付宝，用户点击确认支付；

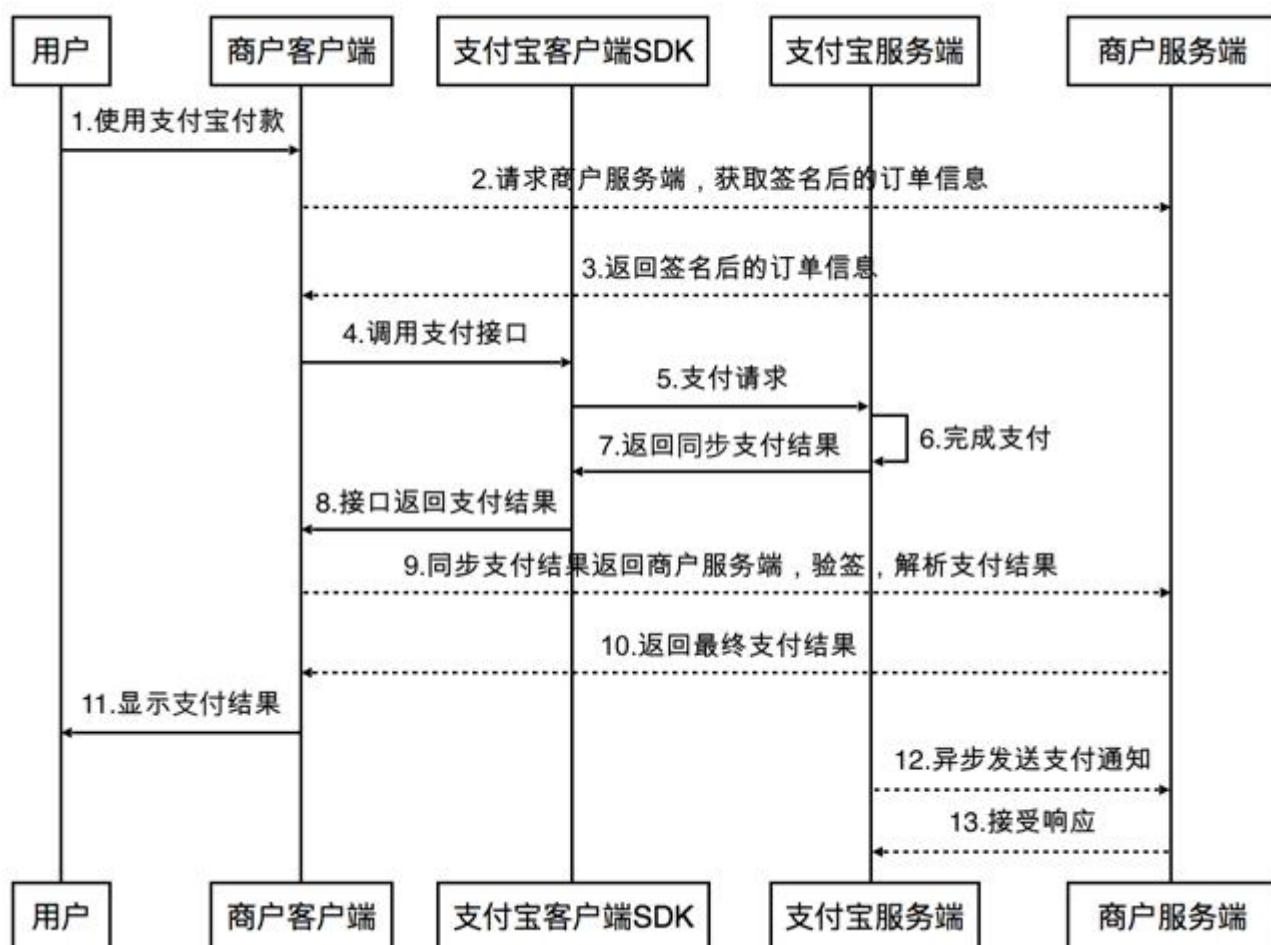
步骤 2：进入到支付宝页面，调起支付宝支付，出现确认支付界面；

步骤 3：用户确认收款方和金额，点击立即支付后出现输入密码界面。

## 1.3. 使用步骤

- (1) 用户完成订单下单；
- (2) 登录支付宝钱包，输入付款，完成密码支付；
- (3) 用户付款后商家收银系统会拿到支付成功或者失败的结果。

## 1.4. 接口调用时序图



如图：以 Android 为例

第 4 步：调用支付接口：此消息就是本接口所描述的支付宝客户端 SDK 提供的支付对象 PayTask, 将商户签名后的订单信息传进 payv2 方法唤起支付宝收银台。

第 5 步：支付请求：支付宝客户端 SDK 将会按照商户客户端提供的请求参数发送。

第 8 步：接口返回支付结果：商户客户端在第 4 步中调用的支付接口，会返回最终的支付结果（即同步通知），参见客户端同步返回。

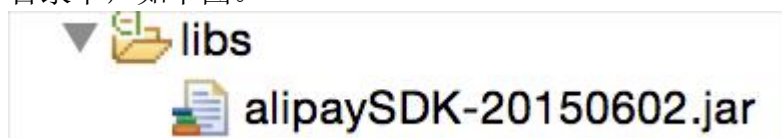
第 13 步：用户在支付宝 APP 或 H5 收银台完成支付后，会根据商户在手机网站支付 API 中传入的前台回跳地址 return\_url 自动跳转回商户页面，同时在 URL 请求中附带支付结果参数。同时，支付宝还会根据原始支付 API 中传入的异步

通知地址 `notify_url`，通过 POST 请求的形式将支付结果作为参数通知到商户系统，详情见支付结果异步通知。

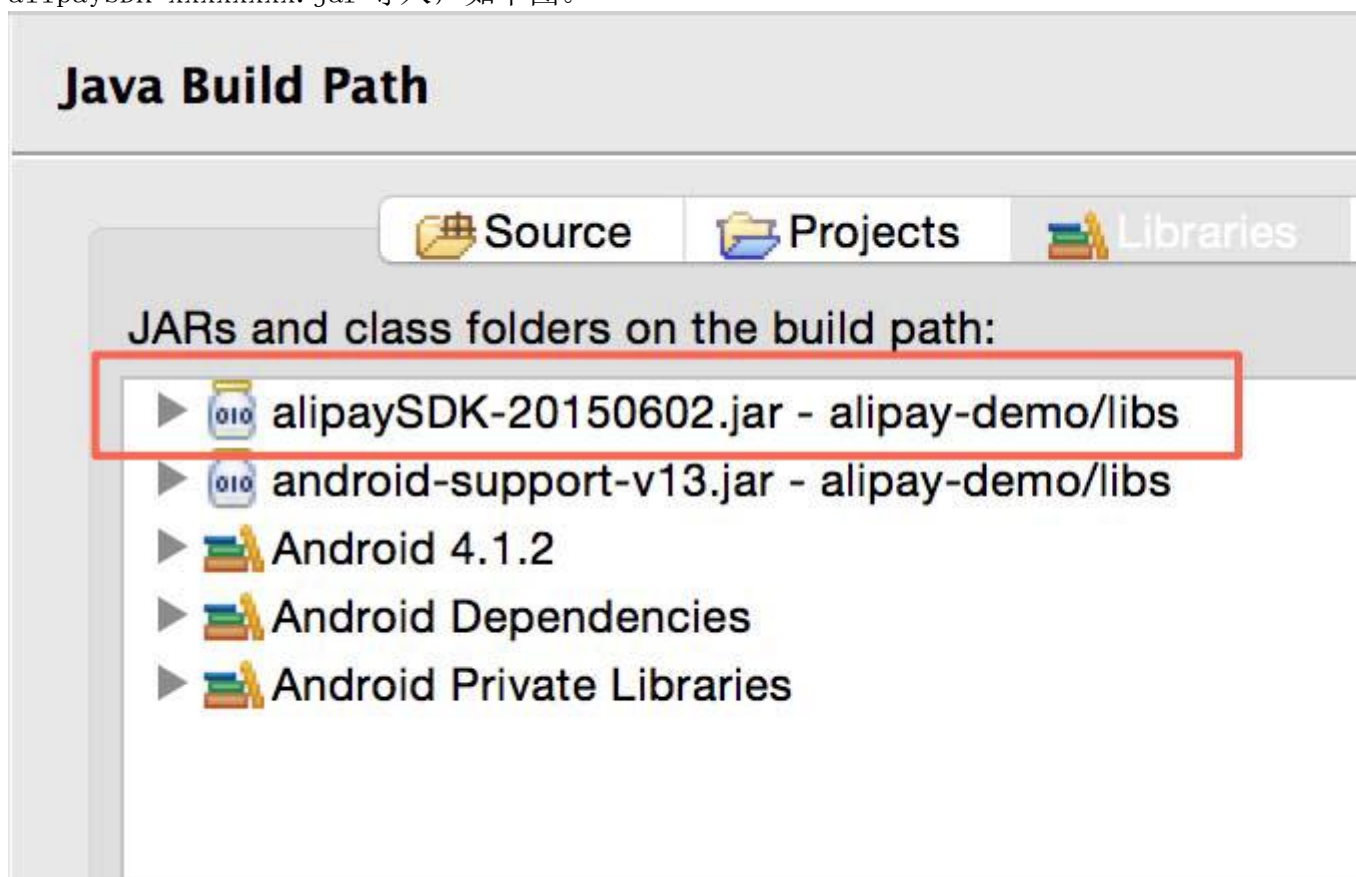
## 1.5. Android 集成流程

### 导入开发资源

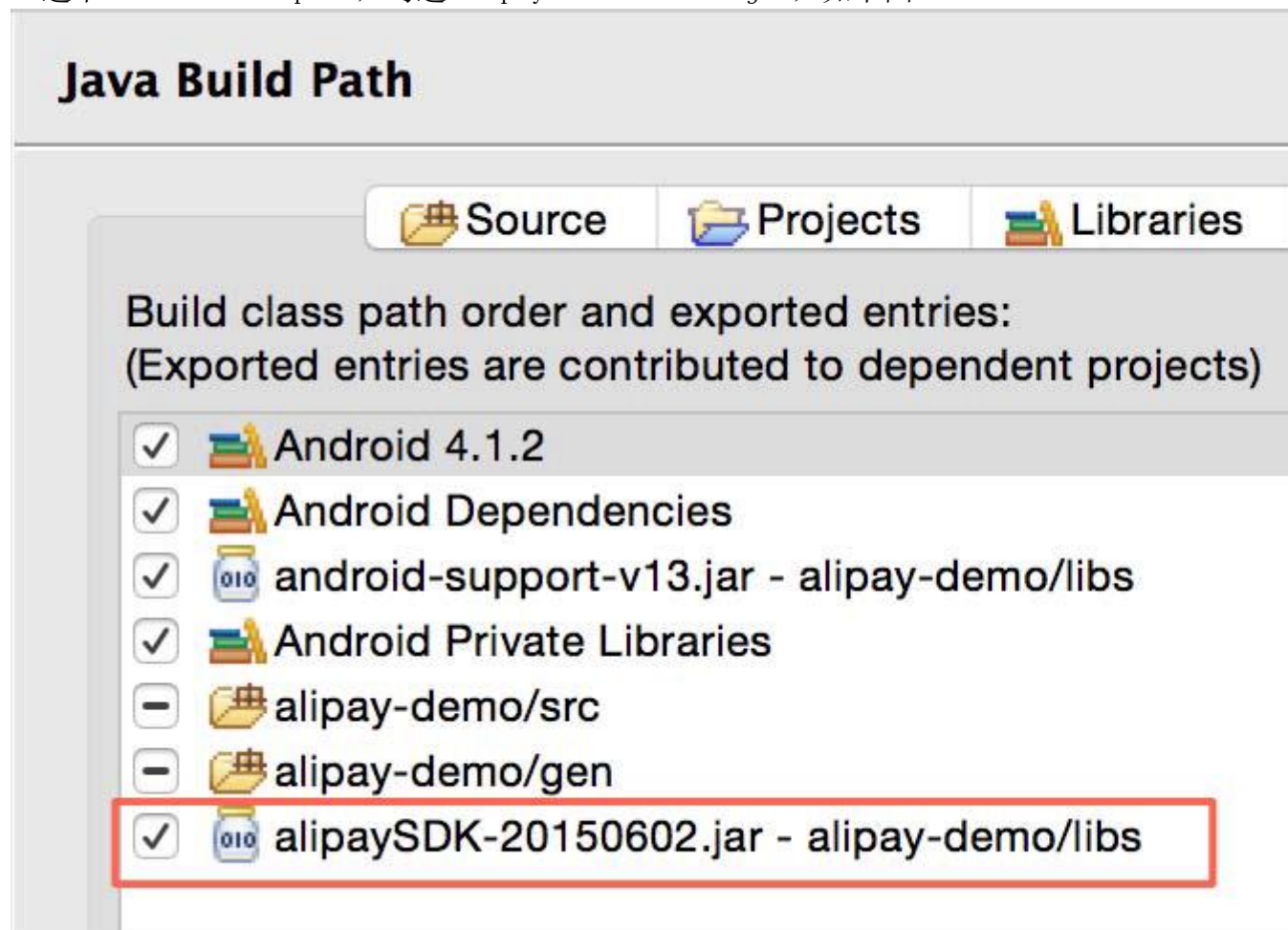
1. 将 `alipay-sdk-common/alipaySdk-xxxxxxx.jar` 包放入商户应用工程的 `libs` 目录下，如下图。



2. 进入商户应用工程的 Java Build Path，将 `libs` 目录下的 `alipaySDK-xxxxxxx.jar` 导入，如下图。



3. 选中 Order and Export, 勾选 alipaySDK-xxxxxxx.jar, 如下图。



## 修改 Manifest

在商户应用工程的 AndroidManifest.xml 文件里面添加声明：

```

1
2
3 <activity
4     android:name="com.alipay.sdk.app.H5PayActivity"
5     android:configChanges="orientation|keyboardHidden|navigation"
6     android:exported="false"
7     android:screenOrientation="behind" >
8 </activity>
9 <activity
10    android:name="com.alipay.sdk.auth.AuthActivity"
11    android:configChanges="orientation|keyboardHidden|navigation"
12    android:exported="false"
13    android:screenOrientation="behind" >
14 </activity>

```

和权限声明：

```

1 <uses-permission android:name="android.permission.INTERNET" />
2 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
3 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

```

```
3<uses-permission android:name="android.permission.READ_PHONE_STATE" />
4<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
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：

```
1
2-libraryjars libs/alipaySDK-20150602.jar
3
4-keep class com.alipay.android.app.IAlipay{*};
4-keep class com.alipay.android.app.IAlipay$Stub{*};
5-keep class com.alipay.android.app.IRemoteServiceCallback{*};
6-keep class com.alipay.android.app.IRemoteServiceCallback$Stub{*};
7-keep class com.alipay.sdk.app.PayTask{ public *};
7-keep class com.alipay.sdk.app.AuthTask{ public *};
8
```

至此，开发包开发资源导入完成。

## 支付接口调用

需要在新线程中调用支付接口。（可参考 `alipay_demo` 实现）

获取 `PayTask` 支付对象调用支付（支付行为需要在独立的非 `ui` 线程中执行），代码示例：

```
1
2
3 final String orderInfo = info; // 订单信息
4
5 Runnable payRunnable = new Runnable() {
6
7     @Override
8     public void run() {
9         PayTask alipay = new PayTask(DemoActivity.this);
9         String result = alipay.payV2(orderInfo,true);
10
11         Message msg = new Message();
11         msg.what = SDK_PAY_FLAG;
12         msg.obj = result;
12         mHandler.sendMessage(msg);
13     }
14 };
14 // 必须异步调用
15 Thread payThread = new Thread(payRunnable);
16 payThread.start();
17
18
```

## 支付结果获取和处理

调用 `pay` 方法支付后，将通过 2 种途径获得支付结果：

## 同步返回

商户应用客户端通过当前调用支付的 **Activity** 的 **Handler** 对象，通过它的回调函数获取支付结果。（可参考 **alipay\_demo** 实现）

代码示例：

```

1
2 private Handler mHandler = new Handler() {
3     public void handleMessage(Message msg) {
4         Result result = new Result((String) msg.obj);
5         Toast.makeText(DemoActivity.this, result.getResult(),
6             Toast.LENGTH_LONG).show();
7     };
8 };

```

## 异步通知

商户需要提供一个 **http** 协议的接口，包含在请求支付的入参中，其 **key** 对应 **notify\_url**。支付宝服务器在支付完成后，会以 **POST** 方式调用 **notify\_url** 传输数据。

## 获取当前开发包版本号

调用 **PayTask** 对象的 **getVersion()** 方法查询。

代码示例：

```

1 PayTask payTask = new PayTask(activity);
2 String version = payTask.getVersion();

```

## 2. 接口说明

### 2.1. 接口调用规则

- ◆ 采用 **HTTPS** 方式访问
- ◆ 请求采用 **POST** 方式
- ◆ 提交和返回结果采用 **XML** 格式
- ◆ 字符集默认使用 **UTF-8**，请勿使用其它字符集
- ◆ 商户与中国银行支付宝扫码支付平台之间的交互(特别是支付通知等的回调)，都需要验证签名。
- ◆ 处理返回时先判断协议返回错误码，再判断业务返回错误码，最后判断交易状态

## 2.2. Sign 签名生成方法

支付宝支付中将对数据里面的内容进行鉴权,确定携带的信息是真实、有效、合理的。因此,这里将定义生成 sign 字符串的方法。

a. 对所有传入参数按照字段名的 ASCII 码从小到大排序(字典序)后,使用 URL 键值对的格式(即 key1=value1&key2=value2...) 拼接成字符串 string1, 注意: 值为空的参数不参与签名;

b. 在 string1 最后拼接上 key=Key(商户支付密钥)得到 stringSignTemp 字符串,并对 stringSignTemp 进行 md5 运算,再将得到的字符串所有字符转换为大写,得到 sign 值 signValue。

下面定义了一段生成 sign 字符串的示范过程:

假设以下为 package 传入参数:

```
appid=wx930ea5d5a258f4f
terminal_id=123
mch_id=1900000109
store_id=s123456
subject=test
body=test
nonce_str=960f228109051b9969f76c82bde183ac
out_trade_no=1400755861
total_amount=1
timeout_express=1h
notify_url=http://wxpay.weixin.qq.com/pub_v2/pay/notify.php
key=8934e7d15453e97507ef794cf7b0519d
```

i: 经过 a 过程 URL 键值对字典序排序后的字符串 string1 为:

```
appid=wx930ea5d5a258f4f&body=test&mch_id=1900000109&nonce_str=960f228109051b9969f76c82bde183ac&notify_url=http://xxx.xxx.com/pub_v2/pay/notify.php&out_trade_no=1400755861&store_id=s123456&subject=test&terminal_id=123&timeout_express=1h&total_amount=1
```

ii: 经过 b 过程后得到 sign 为:

Sign

```

=md5(string1&key=8934e7d15453e97507ef794cf7b0519d).toUpperCase
=md5(appid=wx930ea5d5a258f4f&body=test&mch_id=1900000109&nonce_str=960f228109051b9969f76c82bde183ac&notify_url=http://xxx.xxx.com/pub_v2/pay/notify.php&out_trade_no=1400755861&store_id=s123456&subject=test&terminal_id=123&timeout_express=1h&total_amount=1&key=8934e7d15453e97507ef794cf7b0519d).toUpperCase()
="88f66d378212b9a28073f81699e43582".toUpperCase()
="88F66D378212B9A28073F81699E43582"

```

## 2.3. 代码实例

接口调用实例，以开发为例说明数据提交和返回的具体格式：

**提交数据：**

```

<xml>
<appid> wx930ea5d5a258f4f </appid>
<terminal_id> terminal_id </terminal_id>
<subject><![CDATA[test]]></subject>
<body><![CDATA[test]]></body>
<mch_id>1900000109</mch_id>
<store_id><![CDATA[s123456]]></store_id>
<nonce_str><![CDATA[960f228109051b9969f76c82bde183ac]]></nonce_str>
<notify_url><![CDATA[http://xxx.xxx.com/pub_v2/pay/notify.php]]>
</notify_url>
<out_trade_no>1400755861</out_trade_no>
<total_amount>1</total_amount>
<timeout_express><![CDATA[1h]]></timeout_express>
<sign><![CDATA[88F66D378212B9A28073F81699E43582]]></sign>
</xml>

```

**响应示例：**

```

<xml>
<code>10000</code>
<msg><![CDATA[Success]]></msg>
<out_trade_no>1400755861</out_trade_no>

```

```
<qr_code><![CDATA[https://qr.alipay.com/bax06004nrs5o5kk58dt20b2]]></qr_code>
<nonce_str><![CDATA[FvYsNPuFFPkAr77M]]></nonce_str>
<sign><![CDATA[63238039D6E43634297CF2A6EB5F3B72]]></sign>
</xml>
```

异常示例:

```
<xml>
<code>20000</code>
<msg><![CDATA[Service Currently Unavailable]]></msg>
<sub_code><![CDATA[isp.unknown-error]]></sub_code>
<sub_msg><![CDATA[系统繁忙]]></sub_msg>
</xml>
```

## 3. API 列表

### 3.1. 交易创建接口

#### 3.1.1 应用说明:

收银员通过收银台或商户后台调用交易创建接口,生成订单信息后,由 appsdk 调起支付。

#### 3.1.2 接口地址:

**<https://api.boc.dcorepay.com/alipay/appcreate>**

#### 3.1.3 输入参数:

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写,则默认为 1.0.0
应用 ID	appid	是	String(32)	中国银行分配的 APPID

中国银行支付宝扫码支付接口

商户号	mch_id	是	String(32)	中国银行分配的商户号
设备号	terminal_id	否	String(32)	终端设备号(收银设备 ID)
随机字符串	nonce_str	是	String(32)	随机字符串, 不长于 32 位。 推荐随机数生成算法
签名	sign	是	String(32)	签名, 详细签名方法见 2.2 节
订单标题	subject	是	String(256)	订单标题
商品描述	body	否	String(128)	对交易或商品的描述
商品详情	goods_detail	否	String(4000)	商品描述列表信息, JSON 字符串 <pre>{   "goods_detail":[     {       "goods_id":"iphone6s_16G",       "alipay_goods_id":"1001",       "goods_name":"iPhone6s 16G",       "quantity":1,       "price":5288,       "goods_category":"123456",       "body":"苹果手机",       "show_url":""     }   ] }</pre> 参数说明: goods_id 商品编号, 必填; alipay_goods_id 支付宝定义的统一商品编号, 选填; goods_name 商品名称, 必填; quantity 商品数量, 必填;

				price 商品单价, 必填, 单位元; goods_category 商品类目, 选填 body 商品描述信息, 选填 show_url 商品展示地址, 选填
门店 ID	store_id	是	String(32)	中国银行分配门店 APPID
操作员编号	operator_id	否	String(28)	收银操作员编号
商户订单号	out_trade_no	是	String(64)	商 户 系 统 内 部 的 订 单 号, 64 个 字 符 内、可 包 含 字 母。 <b>注意：请务必确保该订单号在商户系统的唯一性。</b>
货币类型	fee_type	否	String(16)	货币类型。默认为人民币：CNY
总金额	total_amount	是	Int	订单总金额, 单位为分, 不能带小数点
最晚付款时间	timeout_express	是	String(6)	该笔订单允许的最晚付款时间, 逾期将关闭交易。取值范围: 1m~15d。m-分钟, h-小时, d-天, 1c-当天 (1c-当天的情况下, 无论交易何时创建, 都在 0 点关闭)。该参数数值不接受小数点, 如 1.5h, 可转换为 90m
通知地址	notify_url	否	String(256)	接收支付宝支付结果通知

### 3.1.4 返回结果:

字段名	变量名	必填	类型	说明
-----	-----	----	----	----

版本信息	version	否	String(8)	若不填写，则默认为 1.0.0
状态码	code	是	String(6)	10000 表示成功 其它表示错误
状态信息	msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误
以下字段在 code 为 10000 的时候有返回				
商户订单信息	order_str	是	String(1000)	商户订单信息
随机字符串	nonce_str	是	String(32)	中国银行返回的随机字符串，不长于 32 位
签名	sign	是	String(32)	中国银行返回的签名, 详细签名方法见 2.2 节
以下字段在 code 不为 10000 的时候有返回				
错误码	sub_code	是	String(32)	详见错误码定义
错误描述	sub_msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误

### 3.1.5 错误码：

错误码	错误描述	解决方案
ACQ. SYSTEM_ERROR	接口返回错误	请立即调用查询订单 API，查询当前订单的状态，并根据订单状态决定下一步的操作
ACQ. XML_ERROR	XML 格式错误	请检查请求的 XML 格式是否正确
ACQ. INVALID_SIGN	无效签名	请检查是否符合 2.2 签名规则

ACQ. INVALID_APPID	商户 appid 和 mch_id 不存在或已停用或不匹配	
ACQ. ORDER_REPEAT	订单重复	
ACQ. INVALID_PARAMETER	参数无效	检查请求参数,修改后重新发起请求
ACQ. ACCESS_FORBIDDEN	无权限使用接口	
ACQ. EXIST_FORBIDDEN_WORD	订单信息中包含违禁词	修改订单信息后,重新发起请求
ACQ. PARTNER_ERROR	应用 APP_ID 填写错误	
ACQ. TOTAL_FEE_EXCEED	订单总金额超过限额	修改订单金额再发起请求
ACQ. CONTEXT_INCONSISTENT	交易信息被篡改	更换商家订单号后,重新发起请求
ACQ. TRADE_HAS_SUCCESS	交易已被支付	确认该笔交易信息是否为当前买家的,如果是则认为交易付款成功,如果不是则更换商家订单号后,重新发起请求
ACQ. TRADE_HAS_CLOSE	交易已经关闭	更换商家订单号后,重新发起请求
ACQ. BUYER_SELLER_EQUAL	买卖家不能相同	更换买家重新付款
ACQ. TRADE_BUYER_NOT_MATCH	交易买家不匹配	更换商家订单号后,重新发起请求

ACQ. BUYER_ENABLE_STATUS_FORBID	买家状态非法	确认买家状态为什么非法
ACQ. BUYER_PAYMENT_AMOUNT_DAY_LIMIT_ERROR	买家付款日限额超限	更换买家进行支付
ACQ. BEYOND_PAY_RESTRICTION	商户收款额度超限	
ACQ. BEYOND_PER_RECEIPT_RESTRICTION	商户收款金额超过月限额	
ACQ. BUYER_PAYMENT_AMOUNT_MONTH_LIMIT_ERROR	买家付款月额度超限	让买家更换账号后,重新付款或者更换其它付款方式
ACQ. SELLER_BEEN_BLOCKED	商家账号被冻结	
ACQ. ERROR_BUYER_CERTIFY_LEVEL_LIMIT	买家未通过人行认证	让用户更换其它付款方式

## 3.2. 支付结果通知接口

### 3.2.1 应用说明:

✧ **通知触发条件为支付成功。**

**支付成功**后, 中国银行把支付宝相关支付和用户信息发送到该 URL, 商户需要接收处理信息, **并使用订单查询接口查询详细订单信息。**

对后台通知交互时, 如果中国银行收到商户的应答不是成功或超时, 中国银行认为通知失败, 中国银行会通过一定的策略定期重新发起通知, 尽可能提高通知的成功率, 但中国银行不保证通知最终能成功。

由于存在重新发送后台通知的情况, 因此同样的通知可能会多次发送给商户系统。**商户系统必须能够正确处理重复的通知。**

推荐的做法是: 当收到通知进行处理时, 首先检查对应业务数据的状态, 判断该通知是否已经处理过, 如果没有处理过再进行处理, 如果处理过直接返回结

果成功。在对业务数据进行状态检查和处理之前，要采用数据锁进行并发控制，以避免函数重入造成的数据混乱。

### 3.2.2 接口地址：

该链接是通过【交易预创建接口】中提交的参数 `notify_url` 设置，如果链接无法访问，商户将无法接收到通知。

### 3.2.3 通知参数：

字段名	变量名	必填	类型	说明
版本信息	<code>version</code>	否	String(8)	若不填写，则默认为 1.0.0
支付通道类型	<code>pay_type</code>	是	String(6)	固定参数 ALIPAY，表示是支付宝支付。商户在同时开通微信和支付宝的情况下，支付通知 url 可能填写的是一样的，此时可以通过判断通知参数 <code>pay_type</code> 是否为 ALIPAY 来判断是支付宝还是微信的支付通知。
应用 ID	<code>appid</code>	是	String(32)	中国银行分配的 APPID
商户号	<code>mch_id</code>	是	String(32)	中国银行分配的商户号
随机字符串	<code>nonce_str</code>	是	String(32)	随机字符串，不长于 32 位
签名	<code>sign</code>	是	String(32)	签名, 详细签名方法见 2.2 节
订单金额	<code>total_amount</code>	是	Int	本次交易支付的订单金额，单位为人民币（分）
实收金额	<code>receipt_amount</code>	否	Int	商家在交易中实际收到的款项，单位为人民币（分）
开票金额	<code>invoice_amount</code>	否	Int	用户在交易中支付的可开发票的金额，单位为人民币

中国银行支付宝扫码支付接口

				(分)
付款金额	buyer_pay_amount	否	Int	用户在交易中支付的金额,单位为人民币(分)
集分宝支付金额	point_amount	否	Int	使用集分宝支付的金额,单位为人民币(分)
交易状态	trade_status	是	String(32)	交易状态:  WAIT_BUYER_PAY (交易创建, 等待买家付款)  TRADE_CLOSED(未付款交易超时关闭,或支付完成后全额退款)  TRADE_SUCCESS (交易支付成功)  TRADE_FINISHED (交易结束, 不可退款)
买家支付宝用户号	buyer_id	否	String(16)	买家支付宝用户号, 例: 2088102122524333
买家支付宝账号	buyer_logon_id	否	String(100)	买家支付宝账号
支付宝交易号	trade_no	是	String(64)	支付宝交易号
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号
商户订单号	out_trade_no	是	String(64)	商户系统的订单号, 与请求一致。
交易付款时间	gmt_payment	是	String(14)	支付完成时间, 格式为 yyyyMMddhhmmss, 如 2009 年 12 月 27 日 9 点 10 分 10 秒 表示为 20091227091010。时区为

				GMT+8 beijing。
交易创建时间	gmt_create	否	String(14)	交易创建时间，格式为 yyyyMMddhhmmss，如 2009 年 12 月 27 日 9 点 10 分 10 秒表示为 20091227091010。时区为 GMT+8 beijing。
支付金额信息	fund_bill_list	否	String(512)	支付成功的各个渠道金额信息，详见资金明细信息说明 [{“amount”：“15.00”，“fundChannel”：“ALIPAYACCOUNT”}]

### 3.2.4 返回结果：

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写，则默认为 1.0.0
状态码	code	是	String(16)	10000 表示商户接收通知成功并校验成功 其它表示失败
状态信息	msg	否	String(128)	返回状态信息 SUCCESS/FAIL SUCCESS 表示商户接收通知成功并校验成功

### 3.3 查询订单接口

#### 3.3.1 应用说明：

该接口提供所有中国支付宝支付订单的查询，商户可以通过该接口主动查询订单状态，完成下一步的业务逻辑。

需要调用查询接口的情况：

- ✧ 当商户后台、网络、服务器等出现异常，商户系统最终未接收到支付通知；调用支付接口后，返回系统错误或未知交易状态情况；
- ✧ 调用刷卡支付交易支付接口，返回 INPROCESS 的状态；
- ✧ 调用交易撤销接口之前，需确认支付状态；

#### 3.3.2 接口地址：

**<https://api.boc.dcorepay.com/alipay/orderquery>**

#### 3.3.3 输入参数：

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写，则默认为 1.0.0
应用 ID	appid	是	String(32)	中国银行分配的 APPID
商户号	mch_id	是	String(32)	中国银行分配的商户号
支付宝交易号	trade_no	否	String(64)	支付宝交易号，优先于通道订单号和商户订单号。
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号，优先于商户订单号。
商户订单号	out_trade_no	否	String(64)	商户订单号。商户订单号、通道订单号和微信订单号必须传入其中一个订单号，如果三个订单号同时传入，优先顺序为：微信订单号、通道订单号、商户订单号。 <b>三个订单号不能同</b>

				时为空
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
签名	sign	是	String(32)	签名

### 3.3.4 返回结果：

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写，则默认为 1.0.0
状态码	code	是	String(6)	10000 表示成功 其它表示错误
状态信息	msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误
以下字段在 code 为 10000 的时候有返回				
支付宝交易号	trade_no	是	String(64)	支付宝交易号
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号
商户订单号	out_trade_no	是	String(64)	商户订单号
买家支付宝账号	buyer_logon_id	是	String(100)	买家支付宝账号
买家支付宝账的用户 ID	buyer_user_id	是	String(16)	买家支付宝账的用户 ID
交易状态	trade_status	是	String(32)	交易状态： WAIT_BUYER_PAY (交易创建，等待买家付款)  TRADE_CLOSED (未付款交易超时关闭，或支付完成后全额退款)

中国银行支付宝扫码支付接口

				TRADE_SUCCESS(交易支付成功)  TRADE_FINISHED(交易结束, 不可退款)
交易订单金额	total_amount	是	Int	交易订单金额, 单位为分 与请求中的订单金额一致
实收金额	receipt_amount	是	Int	商户实收金额, 单位为分 商户实际入账金额(扣手续费之前)
买家实付金额	buyer_pay_amount	否	Int	买家实付金额, 单位为分 建议打印在小票上避免退款是出现纠纷
积分支付金额	point_amount	否	Int	积分支付金额, 单位为分
可开具发票金额	invoice_amount	否	Int	用户支付的可开具发票金额, 单位为分 快速告知商户应该给用户开多少钱发票
商户门店 ID	store_id	否	String(32)	中国分配给门店的 APPID
设备号	terminal_id	否	String(32)	设备号
支付金额信息	fund_bill_list	是	String(512)	支付成功的各个渠道金额信息, 详见资金明细信息说明 [{"amount": "15.00", "fund_channel": "ALIPAYACCOUNT"}]
随机字符串	nonce_str	是	String(32)	中国银行返回的随机字符串, 不长于 32 位
签名	sign	是	String(32)	中国银行返回的签名, 详细签名方法见 2.2 节
以下字段在 code 不为 10000 的时候有返回				
错误码	sub_code	是	String(32)	详见错误码定义

错误描述	sub_msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误
------	---------	---	-------------	------------------------------------

### 3.3.5 错误码：

错误码	错误描述	解决方案
ACQ. SYSTEM_ERROR	系统错误	重新发起请求
ACQ. XML_ERROR	XML 格式错误	请检查请求的 XML 格式是否正确
ACQ. INVALID_SIGN	无效签名	请检查是否符合 2.2 签名规则
ACQ. INVALID_PARAMETER	参数无效	检查请求参数，修改后重新发起请求
ACQ. TRADE_NOT_EXIST	查询的交易不存在	检查传入的交易号是否正确，修改后重新发起请求

## 3.4. 撤销订单接口

### 3.4.1 应用说明：

支付交易返回失败或支付系统超时，调用该接口撤销交易。

- ✧ 如果此订单用户支付失败，支付宝系统会将此订单关闭；
- ✧ 如果用户支付成功，支付宝系统会将此订单资金退还给用户；
- ✧ 注意：只有发生支付系统超时或者支付结果未知时可调用撤销，其他正常支付的单如需实现相同功能请调用申请退款 API。提交支付交易后调用【查询订单 API】，没有明确的支付结果再调用【撤销订单 API】；

### 3.4.2 接口地址:

**<https://api.boc.dcorepay.com/alipay/cancelorder>**

### 3.4.3 输入参数:

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写, 则默认为 1.0.0
应用 ID	appid	是	String(32)	中国银行分配的 APPID
商户号	mch_id	是	String(32)	中国银行分配的商户号
支付宝交易号	trade_no	否	String(64)	支付宝交易号, 优先于通道订单号和商户订单号。
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号, 优先于商户订单号。
商户订单号	out_trade_no	否	String(64)	商户订单号。商户订单号、通道订单号和微信订单号必须传入其中一个订单号, 如果三个订单号同时传入, 优先顺序为: 微信订单号、通道订单号、商户订单号。 <b>三个订单号不能同时为空</b>
随机字符串	nonce_str	是	String(32)	随机字符串, 不长于 32 位
签名	sign	是	String(32)	签名

### 3.4.4 返回结果:

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写, 则默认为 1.0.0
状态码	code	是	String(6)	10000 表示成功 其它表示错误

状态信息	msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误
以下字段在 code 为 10000 的时候有返回				
支付宝交易号	trade_no	是	String(64)	支付宝交易号
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号
商户订单号	out_trade_no	是	String(64)	商户订单号
是否需要重试标记	retry_flag	是	String(1)	是否需要重试标记 示例：N
撤销触发的交易动作	action	是	String(10)	本次撤销触发的交易动作  close: 关闭交易，无退款 refund: 产生了退款
随机字符串	nonce_str	是	String(32)	中国银行返回的随机字符串，不长于 32 位
签名	sign	是	String(32)	中国银行返回的签名, 详细签名方法见 2.2 节
以下字段在 code 不为 10000 的时候有返回				
错误码	sub_code	是	String(32)	详见错误码定义
错误描述	sub_msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误

### 3.4.5 错误码:

错误码	错误描述	解决方案
AQC.SYSTEM_ERROR	系统错误	请使用相同的参数再次调用

ACQ. XML_ERROR	XML 格式错误	请检查请求的 XML 格式是否正确
ACQ. INVALID_SIGN	无效签名	请检查是否符合 2.2 签名规则
ACQ. TRADE_SUCCESS_NO T_CANCEL	交易成功，请使用退款接口撤销订单	
ACQ. TRADE_CANCEL_REPEAT	订单撤销重复提交，请调用订单查询接口查询订单状态	
ACQ. INVALID_PARAMETER	参数无效	请求参数有错，重新检查请求后，再调用撤销
ACQ. TRADE_NOT_EXIST	交易不存在	检查请求中的交易号和商户订单号是否正确，确认后重新发起
ACQ. SELLER_BALANCE_NOT_ENOUGH	商户的支付宝账户中无足够的资金进行撤销	商户支付宝账户充值后重新发起撤销即可
ACQ. REASON_TRADE_BEEN_FREEZEN	当前交易被冻结，不允许进行撤销	

## 3.5 申请退款接口

### 3.5.1 应用说明：

当交易发生之后一段时间内，由于买家或者卖家的原因需要退款时，卖家可以通过退款接口将支付款退还给买家，支付宝将在收到退款请求并且验证成功之后，按照退款规则将支付款按原路退到买家帐号上。

✧ 交易超过约定时间（签约时设置的可退款时间）的订单无法进行退款；

- ✧ 支付宝退款支持单笔交易分多次退款，多次退款需要提交原支付订单的商户订单号和设置不同的退款单号；
- ✧ 一笔退款失败后重新提交，要采用原来的退款单号；
- ✧ 总退款金额不能超过用户实际支付金额；

### 3.5.2 接口地址：

**<https://api.boc.dcorepay.com/alipay/refund>**

### 3.5.3 输入参数：

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写，则默认为 1.0.0
应用 ID	appid	是	String(32)	中国银行分配的 APPID
商户号	mch_id	是	String(32)	中国银行分配的商户号
支付宝交易号	trade_no	否	String(64)	支付宝交易号，优先于通道订单号和商户订单号。
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号，优先于商户订单号。
商户订单号	out_trade_no	否	String(64)	商户订单号。商户订单号、通道订单号和微信订单号必须传入其中一个订单号，如果三个订单号同时传入，优先顺序为：微信订单号、通道订单号、商户订单号。 <b>三个订单号不能同时为空</b>
退款的金额	refund_amount	是	Int	需要退款的金额，该金额不能大于订单金额，单位为分
商户退款单号	out_refund_no	是	String(64)	标识一次退款请求，同一笔交易多次退款需要保证唯一。
操作员	op_user_id	是	String(32)	操作员 ID，默认为商户号
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位

签名	sign	是	String(32)	签名
----	------	---	------------	----

### 3.5.4 返回结果:

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写, 则默认为 1.0.0
状态码	code	是	String(6)	10000 表示退款成功 10003 表示退款申请中 其它表示错误
状态信息	msg	是	String(128)	返回信息, 如非空, 为错误原因 签名失败 参数格式校验错误
以下字段在 code 为 10000, 10003 的时候有返回				
支付宝交易号	trade_no	是	String(64)	支付宝交易号
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号
商户订单号	out_trade_no	是	String(64)	商户订单号
通道退款单号	pass_refund_no	是		通道退款单号
随机字符串	nonce_str	是	String(32)	中国银行返回的随机字符串, 不长于 32 位
签名	sign	是	String(32)	中国银行返回的签名, 详细签名方法见 2.2 节
以下字段在 code 为 10000 的时候有返回				
买家支付宝账号	buyer_logon_id	是	String(100)	买家支付宝账号
资金变化标识	fund_change	是	String(1)	本次退款是否发生了资金变化 示例: Y
退款总金额	refund_fee	是	Int	退款总金额, 单位分
本次退款金额	send_back_fee	否	Int	本次实际退款金额, 单位分

退款支付时间	gmt_refund_pay	是	String(32)	退款支付时间 格式: 2014-11-27 15:45:57
买家支付宝用户 ID	buyer_user_id	是	String(28)	买家支付宝用户 ID
退款使用的资金渠道	refund_detail_item_list	是	String(28)	退款使用的资金渠道 [{"amount": "15.00", "fund_channel": "ALIPAYACOUNT"}]
以下字段在 code 不为 10000、10003 的时候有返回				
错误码	sub_code	是	String(32)	详见错误码定义
错误描述	sub_msg	是	String(128)	返回信息, 如非空, 为错误原因 签名失败 参数格式校验错误

### 3.5.5 错误码:

错误码	错误描述	解决方案
ACQ. SYSTEM_ERROR	系统错误	请使用相同的参数再次调用
ACQ. XML_ERROR	XML 格式错误	请检查请求的 XML 格式是否正确
ACQ. INVALID_SIGN	无效签名	请检查是否符合 2.2 签名规则
ACQ. REFUND_ACCEPT	退款请求已处理	请调用订单退款查询接口查询订单退款状态
ACQ. INVALID_PARAMETER	参数无效	请求参数有错, 重新检查请求后, 再调用退款
ACQ. SELLER_BALANCE_NOT_ENOUGH	卖家余额不足	商户支付宝账户充值后重新发起退款即可

ACQ. REFUND_AMT_NOT_EQUAL_TOTAL	退款金额超限	检查退款金额是否正确,重新修改请求后,重新发起退款
ACQ. REASON_TRADE_BEEN_FREEZEN	请求退款的交易被冻结	
ACQ. TRADE_NOT_EXIST	交易不存在	检查请求中的交易号和商户订单号是否正确,确认后重新发起
ACQ. TRADE_HAS_FINISHED	交易已完结	该交易已完结,不允许进行退款,确认请求的退款的交易信息是否正确
ACQ. TRADE_STATUS_ERROR	交易状态非法	查询交易,确认交易是否已经付款
ACQ. DISCORDANT_REPEAT_REQUEST	不一致的请求	检查该退款号是否已退过款或更换退款号重新发起请求
ACQ. REASON_TRADE_REFUND_FEE_ERROR	退款金额无效	检查退款请求的金额是否正确
ACQ. TRADE_NOT_ALLOW_REFUND	当前交易不允许退款	检查当前交易的状态是否为交易成功状态以及签约的退款属性是否允许退款,确认后,重新发起请求

## 3.6 查询退款接口

### 3.6.1 应用说明:

商户可使用该接口查询自己通过申请退款接口提交的退款请求是否执行成功。

### 3.6.2 接口地址:

**<https://api.boc.dcorepay.com/alipay/refundquery>**

### 3.6.3 输入参数:

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写, 则默认为 1.0.0
应用 ID	appid	是	String(32)	中国银行分配的 APPID
商户号	mch_id	是	String(32)	中国银行分配的商户号
支付宝交易号	trade_no	否	String(64)	支付宝交易号, 优先于通道订单号和商户订单号。
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号, 优先于商户订单号。
商户订单号	out_trade_no	否	String(64)	商户订单号。商户订单号、通道订单号和微信订单号必须传入其中一个订单号, 如果三个订单号同时传入, 优先顺序为: 微信订单号、通道订单号、商户订单号。 <b>三个订单号不能同时为空</b>
商户退款单号	out_refund_no	是	String(64)	商户退款单号
通道退款单号	pass_refund_no	否	String(64)	通道退款单号 <b>两个退款单号不能同时为空</b>
随机字符串	nonce_str	是	String(32)	随机字符串, 不长于 32 位
签名	sign	是	String(32)	签名

### 3.6.4 返回结果:

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写, 则默认为 1.0.0
状态码	code	是	String(6)	10000 表示请求成功 其它表示错误

中国银行支付宝扫码支付接口

状态信息	msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误
以下字段在 code 为 10000 的时候有返回				
支付宝交易号	trade_no	否	String(64)	支付宝交易号
通道订单号	pass_trade_no	否	String(64)	通道的统一订单号
商户订单号	out_trade_no	否	String(64)	商户订单号
商户退款单号	out_refund_no	否	String(64)	商户退款单号
通道退款单号	pass_refund_no	否	String(64)	通道退款单号
退款状态	refund_status	是	String(16)	退款状态： SUCCESS--退款成功 FAIL--退款失败 PROCESSING--退款处理中
订单金额	total_amount	是	Int	该笔退款所对应的交易订单金额, 单位分
退款请求金额	refund_amount	是	Int	本次退款请求对应的退款金额, 单位分
实际退款金额	send_back_fee	否	Int	本次实际退款金额, 单位分
退款支付时间	gmt_refund_pay	否	String(32)	退款支付时间 格式：2014-11-27 15:45:57
退款使用的资金渠道	refund_detail_item_list	否	String(28)	退款使用的资金渠道 [{"amount": "15.00", "fund_channel": "ALIPAYACOUNT"}]
随机字符串	nonce_str	是	String(32)	中国银行返回的随机字符串, 不长于 32 位
签名	sign	是	String(32)	中国银行返回的签名, 详细签名方法见 2.2 节
以下字段在 code 不为 10000 的时候有返回				

错误码	sub_code	是	String(32)	详见错误码定义
错误描述	sub_msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误

### 3.6.5 错误码：

错误码	错误描述	解决方案
ACQ. SYSTEM_ERROR	系统错误	重新发起请求
ACQ. XML_ERROR	XML 格式错误	请检查请求的 XML 格式是否正确
ACQ. INVALID_SIGN	无效签名	请检查是否符合 2.2 签名规则
ACQ. INVALID_PARAMETER	参数无效	检查请求参数，修改后重新发起请求
ACQ. TRADE_NOT_EXIST	查询退款的交易不存在	确认交易号是否为正确的支付宝交易号，修改后重新查询

## 3.7 对账单接口

### 3.7.1 应用说明：

商户可以通过该接口下载历史交易清单。比如掉单、系统错误等导致商户侧和支付宝侧数据不一致，通过对账单核对后可校正支付状态。

注意：

- 1、支付宝侧未成功下单的交易不会出现在对账单中；
- 2、对账单中涉及金额的字段单位为“元”。

### 3.7.2 接口地址：

**<https://api.boc.dcorepay.com/alipay/downloadbill>**

### 3.7.3 输入参数：

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写，则默认为 1.0.0
应用 ID	appid	是	String(32)	中国银行分配的 APPID
商户号	mch_id	是	String(32)	中国银行分配的商户号
对账单日期		否	String(15)	对账单日期 格式：yyyy-MM-dd
随机字符串	nonce_str	是	String(32)	随机字符串，不长于 32 位
签名	sign	是	String(32)	签名

### 3.7.4 返回结果：

字段名	变量名	必填	类型	说明
版本信息	version	否	String(8)	若不填写，则默认为 1.0.0
状态码	code	是	String(6)	失败，不为 10000 的状态码
状态信息	msg	是	String(128)	状态信息
错误码	sub_code	是	String(32)	详见错误码定义
错误描述	sub_msg	是	String(128)	返回信息，如非空，为错误原因 签名失败 参数格式校验错误

成功时，数据以文本表格的方式返回，第一行为表头，后面各行为对应的字段内容，字段内容跟查询订单或退款结果一致，具体字段说明可查阅相应接口。

### 第一行为表头

商户 ID,支付宝交易号,商户订单号,业务类型,商品名称,创建时间,完成时间,门店编号,门店名称,操作员,终端号,对方账户,订单金额（元）,商家实收（元）,支付宝红包（元）,集分宝（元）,支付宝优惠（元）,商家优惠（元）,券核销金额（元）,券名称,商家红包消费金额（元）,卡消费金额（元）,退款批次号,手续费（元）,费率,实收净额（元）,交易方式,备注

从第二行起，为数据记录，各参数以逗号分隔，参数前增加`符号，为标准键盘 1 左边键的字符，字段顺序与表头一致。其中，商户 ID 指中国银行分配给商户的商户号。

### 倒数第二行为订单统计标题，最后一行为统计数据：

总交易单数,总交易实收额,交易手续费总金额,总退款笔数,总退款金额,退款手续费总金额

### 3.7.5 错误码：

错误码	错误描述	解决方案
INVAILED_ARGUMENTS	入参不合法	确认参数后重新查询
ACQ. XML_ERROR	XML 格式错误	请检查请求的 XML 格式是否正确
ACQ. INVALID_SIGN	无效签名	请检查是否符合 2.2 签名规则
BILL_NOT_EXIST	账单不存在	确认参数后重新查询

UNKNOWN_ERROR	未知错误	
---------------	------	--

## 4. 公共错误码

			解决方案
20000	isp. unknow-error	服务暂不可用（业务系统不可用）	稍后重试
	aop. unknow-error	服务暂不可用（网关自身的未知错误）	稍后重试
20001	aop. invalid-auth-token	无效的访问令牌	请刷新授权令牌或重新授权获取新的令牌
	aop. auth-token-time-out	访问令牌已过期	请刷新授权令牌或重新授权获取新的令牌
	aop. invalid-app-auth-token	无效的应用授权令牌	请刷新应用授权令牌或重新授权获取新的令牌
	aop. invalid-app-auth-token-no-api	商户未授权当前接口	请重新授权获取新的应用授权令牌
	aop. app-auth-token-time-out	应用授权令牌已过期	请刷新应用授权令牌或重新授权获取新的令牌
	aop. no-product-reg-by-partner	商户未签约任何产品	ISV 代理调用的场景，请上线商户的服务窗

40001	isv.missing-method	缺少方法名参数	请求参数里面必须要有 method 参数
	isv.missing-signature	缺少签名参数	检查请求参数, 缺少 sign 参数
	isv.missing-signature-type	缺少签名类型参数	检查请求参数, 缺少 sign_type 参数
	isv.missing-signature-key	缺少签名配置	未上传公钥配置
	isv.missing-app-id	缺少 appId 参数	检查请求参数, 缺少 app_id 参数
	isv.missing-timestamp	缺少时间戳参数	检查请求参数, 缺少 timestamp 参数
	isv.missing-version	缺少版本参数	检查请求参数, 缺少 version 参数
	isv.decryption-error-missing-encrypt-type	解密出错, 未指定加密算法	检查参数, 缺少 encrypt_type 参数
40002	isv.invalid-parameter	参数无效	检查参数, 格式不对、非法值、越界等
	isv.upload-fail	文件上传失败	文件写入失败, 重试
	isv.invalid-file-extension	文件扩展名无效	检查传入的文件扩展名称, 目前支持格式: csv, txt, zip, rar, gz, doc, docx, xls, xlsx, pdf, bmp, gif, jpg, jpeg, png

	isv.invalid-file-size	文件大小无效	检查文件大小，目前支持最大为：50MB
	isv.invalid-method	不存在的方法名	检查入参 method 是否正确
	isv.invalid-format	无效的数据格式	检查入参 format，目前只支持 json 和 xml 2 种格式
	isv.invalid-signature-type	无效的签名类型	检查入参 sign_type，目前只支持 RSA, RSA2, HMAC_SHA1
	isv.invalid-signature	无效签名	1. 公私钥是否是一对 2. 检查公钥上传是否与私钥匹配 3. 存在中文需要做 urlencode 4. 签名算法是否无误
	isv.invalid-encrypt-type	无效的加密类型	检查入参 encrypt_type，目前只支持 AES
	isv.invalid-encrypt	解密异常	重试
	isv.invalid-app-id	无效的 appId 参数	检查入参 app_id，app_id 不存在，或者未上线
	isv.invalid-timestamp	非法的时间戳参数	时间戳参数 timestamp 非法，请检查格式需要为 "yyyy-MM-dd HH:mm:ss"
	isv.invalid-charset	字符集错误	请求参数 charset 错误，目前支持格式：GBK, UTF-8

	isv.invalid-digest	摘要错误	检查请求参数，文件摘要参数必填
	isv.decryption-error-not-valid-encrypt-type	解密出错，不支持的加密算法	检查入参 encrypt_type，目前只支持 AES
	isv.decryption-error-not-valid-encrypt-key	解密出错，未配置加密密钥或加密密钥格式错误	没有配置加密密钥
	isv.decryption-error-unknown	解密出错，未知异常	重试
	isv.missing-signature-config	验签出错，未配置对应签名算法的公钥或者证书	没有配置应用公钥
	isv.not-support-app-auth	本接口不支持第三方代理调用	本接口不支持第三方代理调用
40004	对应业务错误码，明细错误码和解决方案请参见具体的 API 接口文档		
40006	isv.insufficient-isv-permissions	ISV 权限不足	请检查配置的账户是否有当前接口权限
	isv.insufficient-user-permissions	用户权限不足	代理的商户没有当前接口权限