



支付宝钱包支付接口开发包

2.0 标准版

服务名称：Alipay

版本号：1.0

目 录

1 文档说明	4
1.1 功能描述	4
1.2 阅读对象	4
1.3 业务术语	4
2 功能演示	5
3 数据交互	6
4 接口调用方式	7
4.1 iOS	7
4.1.1 接口描述	7
4.1.2 回调接口	8
4.2 Android	9
5 请求参数说明	11
5.1 含义	11
5.2 列表	11
5.3 样例	13
6 同步通知参数说明	14
6.1 含义	14
6.2 列表	14
6.3 样例	15

6.4 同步通知参数获取	15
6.4.1 iOS	15
6.4.2 Android	15
7 服务器异步通知参数说明	15
7.1 含义	15
7.2 列表	15
7.3 通知触发条件	18
7.4 服务器异步通知参数获取	18
7.5 样例	19
8 商户业务处理注意事项	19
9 签名机制	20
9.1 生成待签名的字符串	20
9.1.1 需要参与签名的参数	20
9.1.2 生成待签名字符串	20
9.2 RSA 签名	22
10 如何验证是否支付宝请求	22
11 附录	23
11.1 业务错误码	23
11.2 客户端错误码	23
11.3 交易状态	24
11.4 退款状态	24

1 文档说明

1.1 功能描述

支付宝钱包支付接口开发包 2.0 标准版主要用来向第三方应用程序提供便捷、安全以及可靠的支付服务。本文主要描述开发包支付接口的使用方法，供合作伙伴的开发者接入使用。

1.2 阅读对象

本文档面向具有一定 Android/iOS 客户端开发能力，了解 Android/iOS 客户端的开发和管理人员。

1.3 业务术语

表1-1 业务术语

术语	解释
请求	手机客户端以字符串形式把需要传输的数据发送给接收方的过程。
通知	服务器异步通知。支付宝根据得到的数据处理完成后，支付宝的服务器主动发起通知给商户的网站，同时携带处理完成的结果信息反馈给商户网站。
返回	支付宝以字符串形式直接把处理结果数据返回给手机客户端。

2 功能演示

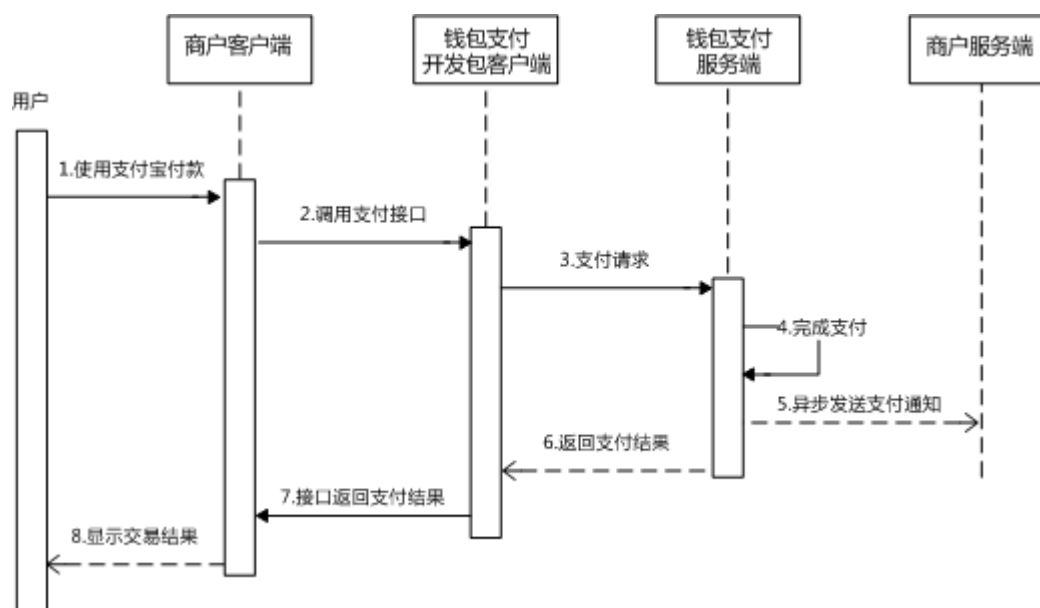


图2-1 钱包支付开发包业务流程图

流程说明（以 Android 平台为例）：

- (1) **第 2 步 调用支付接口**：此消息就是本接口所描述的开发包提供的支付对象 PayTask，将商户订单信息传进pay方法唤起支付宝收银台，订单格式具体参见“5 请求参数说明”。
- (2) **第 3 步**：钱包支付开发包将会按照商户 App 提供的参数发送支付请求。
- (3) **第 5 步 异步发送支付通知**：钱包支付服务器端发送异步通知消息给商户服务器端，参见“7 服务器异步通知参数说明”。
- (4) **第 7 步 接口返回支付结果**：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，参见“6 同步通知参数说明”。

3 数据交互

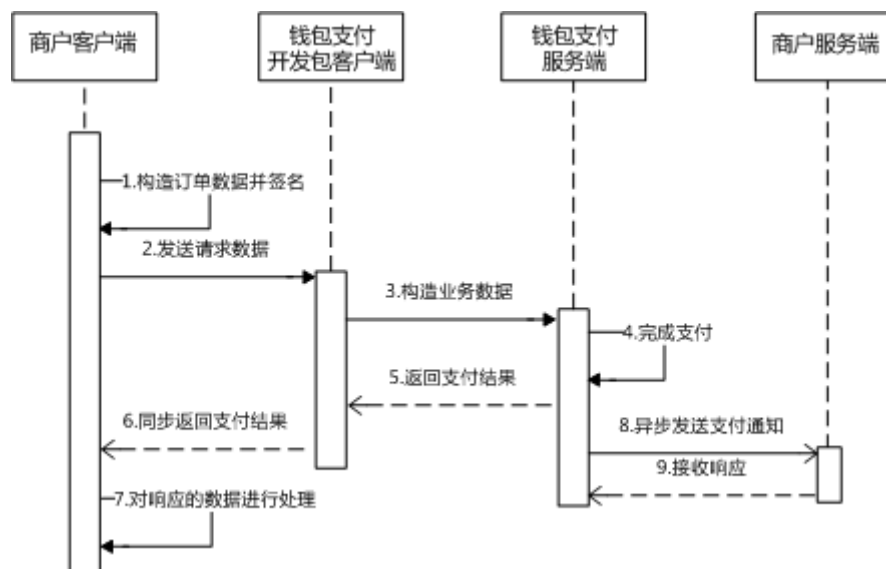


图3-1 商户客户端与支付宝钱包支付开发包交互模式

(1) 构造订单数据并签名

商户客户端根据支付宝钱包支付开发包的接口规则，通过程序生成得到签名结果及要传输给支付宝钱包支付开发包的数据集合。

(2) 发送请求数据

把构造完成的数据集合传递给支付宝钱包支付开发包。

(3) 支付宝钱包支付开发包对请求数据进行处理

支付宝钱包支付开发包将请求数据根据业务规则包装后传递给支付宝服务端，服务端得到这些集合后，会先进行安全校验等验证，一系列验证通过后便会处理完成这次发送过来的数据请求。

(4) 返回处理的结果数据

对于处理完成的交易，支付宝会以两种方式把数据分别反馈给商户应用和商户服务器。

- 在手机客户端上，开发包客户端直接把处理的数据结果反馈给商户客户端；
- 支付宝服务器主动发起通知，调用商户在请求时设定好的页面路径（参数 `notify_url`，如果商户没设定，则不会进行该操作）。

(5) 对获取的返回结果数据进行处理

商户在客户端同步通知接收模块或服务端异步通知接收模块获取支付宝返回的结果数据后，可以结合商户自身业务逻辑进行数据处理（如：订单更新、自动充值到会员账号中等）。同步通知结果仅用于结果展示，入库数据需以异步通知为准。

4 接口调用方式

4.1 iOS

4.1.1 接口描述

iOS 平台上的开发包接口如下表所示：

表4-1 iOS 开发包接口表

接口名称	接口描述
AlipaySDK	提供支付功能。

Alipay 接口主要为商户提供订单支付功能。接口所提供的方法，如下表所示：

表4-2 Alipay 接口方法表

方法名称	方法描述
+(Alipay *)defaultService;	获取服务实例。
- (BOOL)isLogined;	检测本地是否曾登录使用过。
- (void)payOrder:(NSString *)orderStr fromScheme:(NSString *)schemeStr callback:(PaymentCallbackBlock)callbackBlock;	支付并通过回调返回结果。

表4-3 pay 方法信息描述表

方法原型	- (void)payOrder:(NSString *)orderStr fromScheme:(NSString *)schemeStr callback:(PaymentCallbackBlock)callbackBlock;
方法功能	提供给商户快捷订单支付功能。
参数	<p>NSString* scheme 商户程序注册的 URL protocol，供支付完成后回调商户程序使用。</p> <p>PaymentCallbackBlock callbackBlock 快捷支付开发包回调函数，返回免登、支付结果。相应的结果参考“6 同步通知参数说明”。</p> <p>NSString* info 主要包含商户的订单信息，key="value"形式，以&连接。 支付参数示例如下： partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="测试"&body="测试测试" &total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay="30m"&sign="IBBK%2F0w5LOajrMrji7DUgEqNjlhQbidR13GovA5r3TglbNqv231yC1NksLdw%2Ba3JnfHXoXuet6XNNHtn7VE%2BeCoRO1O%2BR1KugLrQEZMtG5jmJle2pbjm%2F3kb%2FuGkpG%2BwYQYI51%2BhA3YBbvZHVQBvYveBqK%2Bh8mUyb7GM1HxWs9k4%3D"&sign_type="RSA" 请参考“5 请求参数说明”查看各个字段的含义。</p>

表4-4 处理客户端方法信息描述表

方法原型	-(void)processOrderWithPaymentResult:(NSURL *)resultUrl standbyCallback:(CompletionBlock)completionBlock;
方法功能	设备已安装支付宝客户端情况下，处理支付宝客户端返回的 url。
参数	<p>NSURL *resultUrl 支付宝客户端回传的 url</p> <p>CompletionBlock completionBlock 当支付宝客户端在操作时，商户 app 进程在后台被结束，只能通过这个 block 输出支付结果。</p> <p>备注： 请在 AppDelegate 的 - (BOOL)application:(UIApplication *)application openURL:(NSURL *)url sourceApplication:(NSString *)sourceApplication annotation:(id)annotation 中调用该方法，具体可参见 Demo。</p>

4.1.2 回调接口

在支付过程结束后，会通过 callbackBlock 同步返回支付结果。

返回结果需要通过 `resultStatus` 以及 `result` 字段的值来综合判断并确定支付结果。在 `resultStatus=9000`，并且 `success="true"`以及 `sign="xxx"`校验通过的情况下，证明支付成功。其它情况归为失败。较低安全级别的场合，也可以只通过检查 `resultStatus` 以及 `success="true"`来判定支付结果。以下为订单支付成功的完成信息示例：

```
resultStatus={9000};memo={};result={partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject=" 测试 "&body=" 测试 "&total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay="30m"&success="true"&sign_type="RSA"&sign="hkFZr+zE9499nuqDNLZEF7W75RFFPsly876QuRSeN8WMaUgcdR00IKy5ZyBJ4eldhoJ/2zghqrD4E2G2mNjs3aE+HCLiBXrPDNdLKCZgSOIqmv46TfPTEqopYfhs+o5fZzXxt34fwdrzN4mX6S13cr3UwmEV4L3Ffir/02RBVtU="}
```

4.2 Android

Android 平台上的快捷支付开发包接口如下表所示：

表4-5 Android 开发包接口表

接口名称	接口描述
PayTask	开发包提供支付，查询的对象接口。

`PayTask` 对象主要为商户提供订单支付功能，查询该设备终端是否存在认证过的支付宝账户，及获取当前开发包版本号。

接口所提供的方法，如下表所示：

表4-6 payTask.pay 描述表

方法原型	PayTask payTask = new PayTask(activity); payTask.pay(orderInfo);
方法功能	提供给商户订单支付功能。
方法参数	<p>实例化 PayTask，传入参数 activity 的实例。</p> <p>String orderInfo</p> <p>主要包含商户的订单信息，key="value"形式，以&连接。</p> <p>支付参数示例如下：</p> <pre>partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="测试"&body="测试测试"&total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay="30m"&sign="IBBK%2F0w5LOajrMrji7DUgEqNjlhQbidR13GovA5r3TglbNqv231yC1NksLdw%2Ba3JnfHXoXuet6XNNHtn7VE%2BeCoRO1O%2BR1KugLrQEZMtG5jmJle2pbjm%2F3kb%2FuGkpG%2BwYQYI51%2BhA3YBbvZHVQBvYveBqK%2Bh8mUyb7GM1HxWs9k4%3D"&sign_type="RSA"</pre> <p>请参考“5 请求参数说明”查看各个字段的含义。</p>
返回值	<p>本方法调用的结果。字符串格式，形式一般如下：</p> <pre>resultStatus={9000};memo={};result={partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="测试"&body="测试测试"&total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay="30m"&success="true"&sign_type="RSA"&sign="hkFZr+zE9499nuqDNLZEF7W75RFFPsly876QuRSeN8WMaUgcdR00IKy5ZyBJ4eldhoJ/2zghqrD4E2G2mNjs3aE+HCLiBXrPDNdLKCZgSOlqmv46TfPTEqopYfhs+o5fZzXxt34fwdrzN4mX6S13cr3UwmEV4L3Ffir/02RBVtU="}</pre> <p>请参考“6 同步通知参数说明”查看各个字段的含义。</p>

表4-7 payTask.checkAccountIfExist 描述表

方法原型	PayTask payTask = new PayTask(activity); payTask.checkAccountIfExist ();
方法功能	查询该终端是否存在支付宝认证账户。
方法参数	实例化 PayTask，传入参数 activity 的实例。
返回值	Boolean 类型值



注意：

payTask.checkAccountIfExist 方法不必在支付接口 PayTask pay 方法前调用，商户应根据自己的需要使用。



表4-8 payTask.getVersion 描述表

方法原型	PayTask payTask = new PayTask(activity); payTask.getVersion();
方法功能	获取当前开发包版本号。
方法参数	实例化 PayTask，传入参数 activity 的实例。
返回值	String 类型值，如 “2.0.0”。

5 请求参数说明

5.1 含义

请求参数是商户在与支付宝进行数据交互时，提供给支付宝的请求数据，以便支付宝根据这些数据进一步处理。

5.2 列表

表5-1 请求参数说明

参数	参数名称	类型（字节长度）	参数说明	是否可为空	样例
基本参数					
service	接口名称	String	接口名称。固定值。	不可空	mobile.securitypay.pay
partner	合作者身份 ID	String(16)	签约的支付宝账号对应的支付宝唯一用户号。 以 2088 开头的 16 位纯数字组成。	不可空	2088101568358171
_input_charset	参数编码字符集	String	商户网站使用的编码格式，固定为 utf-8。	不可空	utf-8
sign_type	签名方式	String	签名类型，目前仅支持 RSA。	不可空	RSA

参数	参数名称	类型（字节长度）	参数说明	是否可为空	样例
sign	签名	String	请参见“9 签名机制”。	不可空	IBBK%2F0w5LOajrMrji7DUgEqNjIhQbidR13GovA5r3TglbNqv231yC1NksLdw%2Ba3JnfHXoXuet6XNNHtn7VE%2BeCoRO1O%2BR1KugLrQEZMtG5jmJle2pbjm%2F3kb%2FuGkpG%2BwYQYI51%2BhA3YBbvZHVQBYveBqK%2Bh8mUyb7GM1HxWs9k4%3D
notify_url	服务器异步通知页面路径	String(200)	支付宝服务器主动通知商户网站里指定的页面 http 路径。	不可空	http://notify.msp.hk/notify.htm
app_id	客户端号	String	标识客户端。	可空	external
appenv	客户端来源	String	标识客户端来源。参数值内容约定如下： appenv="system=客户端平台名^version=业务系统版本”，例如： appenv="system=iphone^version=3.0.1.2" appenv="system=ipad^version=4.0.1.1"	可空	appenv="system=android^version=3.0.1.2"
业务参数					
out_trade_no	商户网站唯一订单号	String(64)	支付宝合作商户网站唯一订单号。	不可空	0819145412-6177
subject	商品名称	String(128)	商品的标题/交易标题/订单标题/订单关键字等。 该参数最长为 128 个汉字。	不可空	测试
payment_type	支付类型	String(4)	支付类型。默认值为：1（商品购买）。	不可空	1
seller_id	卖家支付宝账号	String(16)	卖家支付宝账号（邮箱或手机号码格式）或其对应的支付宝唯一用户号（以 2088 开头的纯 16 位数字）。	不可空	xxx@alipay.com

参数	参数名称	类型（字节长度）	参数说明	是否可为空	样例
total_fee	总金额	Number	该笔订单的资金总额，单位为RMB-Yuan。取值范围为[0.01, 100000000.00]，精确到小数点后两位。	不可空	0.01
body	商品详情	String(512)	对一笔交易的具体描述信息。如果是多种商品，请将商品描述字符串累加传给 body。	不可空	测试测试
it_b_pay	未付款交易的超时时间	String	<p>设置未付款交易的超时时间，一旦超时，该笔交易就会自动被关闭。</p> <p>当用户输入支付密码、点击确认付款后（即创建支付宝交易后）开始计时。</p> <p>取值范围：1m~15d，或者使用绝对时间（示例格式：2014-06-13 16:00:00）。</p> <p>m-分钟，h-小时，d-天，1c-当天（无论交易何时创建，都在0点关闭）。</p> <p>该参数数值不接受小数点，如1.5h，可转换为90m。</p>	可空	30m
extern_token	授权令牌	String(32)	<p>开放平台返回的包含账户信息的 token（授权令牌，商户在一定时间内对支付宝某些服务的访问权限）。通过授权登录后获取的 alipay_open_id，作为该参数的 value，登录授权账户即为支付账户。</p>	可空	1b258b84ed2faf3e88b4d979ed9fd4db

说明：

部分参数类型为 String，未指明长度范围，表明系统不校验该参数的长度。

5.3 样例

```
partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="      测      试      "&body="      测      试      测      试      "&total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay=
```

```
"30m"&sign="lBBK%2F0w5LOajrMrji7DUgEqNjIhQbidR13GovA5r3TgIbNqv23lyC1NksLd
w%2Ba3JnfHXoXu6t6XNNHtn7VE%2BeCoRO10%2BR1KugLrQEZMtG5jmJIe2pbjm%2F3kb%2Fu
GkpG%2BwYQYI51%2BhA3YBbvZHVQBYveBqK%2Bh8mUyb7GM1HxWs9k4%3D"&sign_type="RS
A"
```

6 同步通知参数说明

6.1 含义

支付宝对商户的请求数据处理完成后，会将处理的结果数据直接通知给商户。这些处理结果数据就是同步通知参数。

6.2 列表

表6-1 同步通知参数说明

参数	参数名称	类型（长度范围）	参数说明	是否可为空	样例
resultStatus	状态代码	String	本次操作的状态返回值，标识本次调用的结果，参见“11.2 客户端错误码”。	不可空	9000
result	本次操作返回的结果数据	String	本次操作返回的结果数据。 其中： &success="true"&sign_type="RSA"&sign="xxx"之前的部分为商户的原始数据。 success 用来标识本次支付结果。 sign="xxx"为支付宝对本次支付结果的签名，商户可以使用签约时支付宝提供的公钥进行验证。	不可空	partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="测试"&body="测试测试" &total_fee="0.01"¬ify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&input_charset="utf-8"&it_b_pay="30m"&success="true"&sign_type="RSA"&sign="hkFZr+zE9499nuqDNLZEF7W75RFFPsly876QuRSeN8WMAUgCdR00IKy5ZyBJ4eldhoJ/2zghqrD4E2G2mNjs3aE+HCLiBXrPDNdLKCZgSOlqmv46TfPTEqopYfhs+o5fZzXxt34fwdrzN4mX6S13cr3UwmEV4L3Ffir/02RBVtU="
memo	提示信息	String	保留参数，一般无内容。	可空	

6.3 样例

```
resultStatus={9000};memo={};result={partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="测试"&body="测试测试"&total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay="30m"&success="true"&sign_type="RSA"&sign="hkFZr+zE9499nuqDNLZEF7W75RFFPsly876QuRSeN8WMaUgcdR00IKy5ZyBJ4eldhoJ/2zghqrD4E2G2mNjs3aE+HCLiBXrPDNdLKCZgSOIqmv46TfPTEqopYfhs+o5fZzXxt34fwdrzN4mX6S13cr3UwmEV4L3Ffir/02RBVtU="}
```

6.4 同步通知参数获取

6.4.1 iOS

请参考“4.1.2 回调接口”。

6.4.2 Android

在新线程实例化 PayTask 对象，调用 pay 方法，通过 Handler 对象通知主线程获取支付结果。可参考 alipay_sdk_demo 工程实现。

7 服务器异步通知参数说明

7.1 含义

支付宝对商户的请求数据处理完成后，会将处理的结果数据通过服务器主动通知的方式通知给商户网站。这些处理结果数据就是服务器异步通知参数。

7.2 列表

表7-1 支付异步通知参数说明

参数	参数名称	类型（长度范围）	参数说明	是否可为空	样例
基本参数					
notify_time	通知时间	Date	通知的发送时间。 格式为 yyyy-MM-dd HH:mm:ss。	不可空	2013-08-22 14:45:24

参数	参数名称	类型（长度范围）	参数说明	是否可为空	样例
notify_type	通知类型	String	通知的类型。	不可空	trade_status_sync
notify_id	通知校验 ID	String	通知校验 ID。	不可空	64ce1b6ab92d00ede0ee56ade98df2f4c
sign_type	签名方式	String	固定取值为 RSA。	不可空	RSA
sign	签名	String	请参见“9 签名机制”。	不可空	1glihU9DPWee+UJ82u3+mw3Bdnr9u01at0M/xJnPsGuHh+JA5bk3zbWaoWhU6GmLab3dIM4JNdkTcEUl9/FBGhgfLO39BKX/eBCFQ3bXAmIZn4l26fiwoO613BptT44GTEtnPiQ6+tnLsGIVSrFZaLB9FVhrGfipH2SWJcnwYs=
业务参数					
out_trade_no	商户网站唯一订单号	String(64)	对应商户网站的订单系统中的唯一订单号，非支付宝交易号。需保证在商户网站中的唯一性。是请求时对应的参数，原样返回。	可空	082215222612710
subject	商品名称	String(128)	商品的标题/交易标题/订单标题/订单关键字等。 它在支付宝的交易明细中排在第一列，对于财务对账尤为重要。是请求时对应的参数，原样通知回来。	可空	测试
payment_type	支付类型	String(4)	支付类型。默认值为：1（商品购买）。	可空	1
trade_no	支付宝交易号	String(64)	该交易在支付宝系统中的交易流水号。 最短 16 位，最长 64 位。	可空	2013082244524842
trade_status	交易状态	String	交易状态，取值范围请参见“11.3 交易状态”。	可空	TRADE_SUCCESS
seller_id	卖家支付宝用户号	String(30)	卖家支付宝账号对应的支付宝唯一用户号。 以 2088 开头的纯 16 位数字。	可空	2088501624816263
seller_email	卖家支付宝账号	String(100)	卖家支付宝账号，可以是 email 和手机号码。	可空	xxx@alipay.com
buyer_id	买家支付宝用户号	String(30)	买家支付宝账号对应的支付宝唯一用户号。 以 2088 开头的纯 16 位数字。	可空	2088602315385429

参数	参数名称	类型(长度范围)	参数说明	是否可为空	样例
buyer_email	买家支付宝账号	String(100)	买家支付宝账号，可以是 Email 或手机号码。	可空	dlwdgl@gmail.com
total_fee	交易金额	Number	该笔订单的总金额。 请求时对应的参数，原样通知回来。	可空	1.00
quantity	购买数量	Number	购买数量，固定取值为 1（请求时使用的是 total_fee）。	可空	1
price	商品单价	Number	price 等于 total_fee（请求时使用的是 total_fee）。	可空	1.00
body	商品描述	String(512)	该笔订单的备注、描述、明细等。 对应请求时的 body 参数，原样通知回来。	可空	测试测试
gmt_create	交易创建时间	Date	该笔交易创建的时间。 格式为 yyyy-MM-dd HH:mm:ss。	可空	2013-08-22 14:45:23
gmt_payment	交易付款时间	Date	该笔交易的买家付款时间。 格式为 yyyy-MM-dd HH:mm:ss。	可空	2013-08-22 14:45:24
is_total_fee_adjust	是否调整总价	String(1)	该交易是否调整过价格。	可空	N
use_coupon	是否使用红包买家	String(1)	是否在交易过程中使用了红包。	可空	N
discount	折扣	String	支付宝系统会把 discount 的值加到交易金额上，如果有折扣，本参数为负数，单位为元。	可空	0.00
refund_status	退款状态	String	取值范围请参见“11.4 退款状态”。	可空	REFUND_SUCCESS
gmt_refund	退款时间	Date	卖家退款的时间，退款通知时会发送。 格式为 yyyy-MM-dd HH:mm:ss。	可空	2008-10-29 19:38:25

7.3 通知触发条件

表7-2 通知触发条件

触发条件名	触发条件描述	备注
TRADE_FINISHED	交易成功	true（触发通知）
TRADE_SUCCESS	支付成功	true（触发通知）
WAIT_BUYER_PAY	交易创建	true（触发通知）
TRADE_CLOSED	交易关闭	false（不触发通知）

说明：

true（触发通知）/false（不触发通知）具体值和签约配置时保持同步。

7.4 服务器异步通知参数获取

- (1) 必须保证服务器异步通知页面（notify_url）上无任何字符，如空格、HTML 标签、开发系统自带抛出的异常提示信息等；
- (2) 支付宝是用 POST 方式发送通知信息，因此该页面中获取参数的方式，如：
`request.Form("out_trade_no")`、`$_POST['out_trade_no']`；
- (3) 支付宝主动发起通知，该方式才会被启用；
- (4) 服务器间的交互，不像页面跳转同步通知可以在页面上显示出来，这种交互方式是不可见的；
- (5) 程序执行完后必须打印输出“**success**”（不包含引号）。如果商户反馈给支付宝的字符不是 **success** 这 7 个字符，支付宝服务器会不断重发通知，直到超过 24 小时 22 分钟。
一般情况下，25 小时以内完成 8 次通知（通知的间隔频率一般是：
2m,10m,10m,1h,2h,6h,15h）；
- (6) 程序执行完成后，该页面不能执行页面跳转。如果执行页面跳转，支付宝会收不到 **success** 字符，会被支付宝服务器判定为该页面程序运行出现异常，而重发处理结果通知；
- (7) **cookies**、**session** 等在此页面会失效，即无法获取这些数据；
- (8) 该方式的调试与运行必须在服务器上，即互联网上能访问；
- (9) 该方式的作用主要防止订单丢失，即页面跳转同步通知没有处理订单更新，它则去处理；
- (10) 当商户收到服务器异步通知并打印出 **success** 时，服务器异步通知参数 **notify_id** 才会失效。也就是说在支付宝发送同一条异步通知时（包含商户并

未成功打印出 success 导致支付宝重发数次通知），服务器异步通知参数 notify_id 是不变的。

7.5 样例

```
http://notify.java.jpxx.org/index.jsp?discount=0.00&payment_type=1&subject=测
&trade_no=2013082244524842&buyer_email=dlwdgl@gmail.com&gmt_create=2013-08-22
14:45:23&notify_type=trade_status_sync&quantity=1&out_trade_no=0822152226
12710&seller_id=2088501624816263&notify_time=2013-08-22 14:45:24&body=测试
&trade_status=TRADE_SUCCESS&is_total_fee_adjust=N&total_fee=1.00&gmt_payment=2013-08-22
14:45:24&seller_email=xxx@alipay.com&price=1.00&buyer_id=2088602315385429
&notify_id=64celb6ab92d0ede0ee56ade98fdf2f4c&use_coupon=N&sign_type=RSA&
sign=lglihU9DPWee+UJ82u3+mw3Bdnr9u01at0M/xJnPsGuHh+JA5bk3zbWaoWhU6GmLab3d
IM4JNdktTcEUI9/FBGhgflO39BkX/eBCFQ3bXAmIZn4l26fiwoO6l3BptT44GTetnPiQ6+tnL
sGlVSrFZaLB9FVhrGfipH2SWJcnwYs=
```

8 商户业务处理注意事项

当支付宝处理完成后会把数据结果反馈给商户。商户获得这些数据时，**必须根据支付宝不同类型的业务通知，正确的进行不同的业务处理，并且过滤重复的通知结果数据。在支付宝的业务通知中，只有交易通知状态为 TRADE_SUCCESS 或 TRADE_FINISHED 时，支付宝才会认定为买家付款成功。**如果商户需要对同步返回的数据做验签，必须通过服务端的签名验签代码逻辑来实现。如果商户未正确处理业务通知，存在潜在的风险，商户自行承担因此而产生的所有损失。



注意：

- 交易状态 TRADE_SUCCESS 的通知触发条件是商户签约的产品支持退款功能的前提下，买家付款成功；
- 交易状态 TRADE_FINISHED 的通知触发条件是商户签约的产品不支持退款功能的前提下，买家付款成功；或者，商户签约的产品支持退款功能的前提下，交易已经成功并且已经超过可退款期限；
- 交易成功之后，商户（高级即时到账或机票平台商）可调用批量退款接口，系统会发送退款通知给商户，具体内容请参见批量退款接口文档；
- 当商户使用站内退款时，系统会发送包含 refund_status 和 gmt_refund 字段的 **通知给商户。**

9 签名机制

9.1 生成待签名的字符串

9.1.1 需要参与签名的参数

- 在请求参数列表中，除去 `sign`、`sign_type` 两个参数外，其他需要使用到的参数皆是要签名的参数。
- 在同步通知参数列表中，除去 `sign`、`sign_type` 两个参数外，凡是通知返回回来的参数皆是要验签的参数。
- 在服务器异步通知参数列表中，除去 `sign`、`sign_type` 两个参数外，凡是通知返回回来的参数皆是要验签的参数。
- `sign` 值要做 utf-8 URLencode。

9.1.2 生成待签名字符串

1. 支付接口待签名字符串生成

对于如下的参数数组：

```
string[] parameters={
    "service=\"mobile.securitypay.pay\"",
    "partner=\"2088101568338364\"",
    "_input_charset=\"utf-8\"",
    "notify_url=\"http://notify.msp.hk/notify.htm\"",
    "out_trade_no=\"0819145412-6177\"",
    "subject=\"《暗黑破坏神 3:凯恩之书》\"",
    "payment_type=\"1\"",
    "seller_id=\"alipay-test01@alipay.com\"",
    "total_fee=\"0.01\""
};
```

把所有数组值以 `key= "value"` 进行组合，之后用 “&” 字符连接起来，支持无序，如：

```
service="mobile.securitypay.pay"&partner="2088101568338364"&_input_charse
t="utf-8"&notify_url="http://notify.msp.hk/notify.htm"&out_trade_no="0819
145412-6177"&subject="                                测                    试
"&payment_type="1"&seller_id="xxx@alipay.com"&total_fee="0.01"
```

这串字符串便是待签名字符串。

2. 同步通知待签名字符串生成

对于同步通知的内容:

```
resultStatus={9000};memo={};result={partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="测试"&body="测试测试"&total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay="30m"&success="true"&sign_type="RSA"&sign="hkFZr+zE9499nuqDNLZEF7W75RFFPsly876QuRSeN8WMaUgcdR00IKy5ZyBJ4eldhoJ/2zghqrD4E2G2mNjs3aE+HCLiBXrPDNdLKCZgSOIqmv46TfPTEqopYfhs+o5fZzXxt34fwdrzN4mX6S13cr3UwmEV4L3Ffir/02RBvtU="}
```

取出 **result** 中的部分, 去掉 **sign** 和 **sign_type** 两个参数:

```
partner="2088101568358171"&seller_id="xxx@alipay.com"&out_trade_no="0819145412-6177"&subject="测试"&body="测试测试测试测试"&total_fee="0.01"&notify_url="http://notify.msp.hk/notify.htm"&service="mobile.securitypay.pay"&payment_type="1"&_input_charset="utf-8"&it_b_pay="30m"&success="true"
```

即为待签名字符串。

3. 异步通知待签名字符串生成

对于异步通知的内容:

```
http://notify.java.jpaxx.org/index.jsp?discount=0.00&payment_type=8&subject=测试&trade_no=2013082244524842&buyer_email=dlwdgl@gmail.com&gmt_create=2013-08-22 14:45:23&notify_type=trade_status_sync&quantity=1&out_trade_no=082215222612710&seller_id=2088501624816263&notify_time=2013-08-22 14:45:24&body=测试测试&trade_status=TRADE_SUCCESS&is_total_fee_adjust=N&total_fee=1.00&gmt_payment=2013-08-22 14:45:24&seller_email=xxx@alipay.com&price=1.00&buyer_id=2088602315385429&notify_id=64celb6ab92d00ede0ee56ade98fdf2f4c&use_coupon=N&sign_type=RSA&sign=lglihU9DPWee+UJ82u3+mw3Bdnr9u01at0M/xJnPsGuHh+JA5bk3zbWaoWhU6GmLab3dIM4JNdktTcEUI9/FBGhgflO39BKX/eBCFQ3bXAmIZn4l26fiwoO613BptT44GTetnPiQ6+tnLsGlVSrFZaLB9FVhrGfipH2SWJcnwYs=
```

去掉 **sign** 和 **sign_type** 两个参数, 将其他参数按照字母顺序升序排列, 再把所有数组值以 “&” 字符连接起来:

```
body=测试测试测试测试&buyer_email=dlwdgl@gmail.com&buyer_id=2088602315385429&discount=0.00&gmt_create=2013-08-22 14:45:23&gmt_payment=2013-08-22 14:45:24&is_total_fee_adjust=N&notify_time=2013-08-22
```

```
14:45:24&notify_type=trade_status_sync&out_trade_no=082215222612710&payme  
nt_type=8&price=1.00&quantity=1&seller_email=alipayrisk18@alipay.com&sell  
er_id=2088501624816263&subject 测 试  
&total_fee=1.00&trade_no=2013082244524842&trade_status=TRADE_SUCCESS&use_  
coupon=N
```

即为待签名字符串。



注意：

- 没有值的参数无需传递，也无需包含到待签名数据中；
- 签名时将字符转化成字节流时指定的字符集与_input_charset 保持一致；
- 如果传递了_input_charset 参数，这个参数也应该包含在待签名数据中。

9.2 RSA签名

在 RSA 的签名时，需要私钥和公钥一起参与签名。私钥与公钥皆是客户通过 OPENSSL 来生成得出的。客户把生成出的公钥与支付宝技术人员配置好的支付宝公钥做交换。因此，在签名时，客户要用到的是客户的私钥及支付宝的公钥。

• 请求时签名

当拿到请求时的待签名字符串后，把待签名字符串与客户的私钥一同放入 RSA 的签名函数中进行签名运算，从而得到签名结果字符串。

• 通知返回时验证签名

当获得到通知返回时的待签名字符串后，把待签名字符串、支付宝提供的公钥、支付宝通知返回参数中的参数 sign 的值三者一同放入 RSA 的签名函数中进行非对称的签名运算，来判断签名是否验证通过。

10 如何验证是否支付宝请求

• 目的

验证此次通知信息是否是支付宝服务器发来的信息，以帮助校验反馈回来的数据真假性。

• 工作原理

获取支付宝返回数据之一的通知校验 ID (notify_id)，按照支付宝要求的格式拼接成要请求的链接，如：

https://mapi.alipay.com/gateway.do?service=notify_verify&partner=2088002396712354¬ify_id=RqPnCoPT3K9%252Fvwbh3I%252BFioE227%252BPfNMI8jwyZqMliXQWxhOCmQ5MQO%252FWd93rvCB%252BaiGg

通过访问这个请求链接,利用编程方法来模拟 http 请求与支付宝服务器进行交互,获得支付宝服务器上处理的结果。

如果获得的信息是 `true`, 则校验成功; 如果获得的信息是其他, 则校验失败。

11 附录

11.1 业务错误码

表11-1 业务错误码

错误代码 (error_code)	含义
9000	订单支付成功
8000	正在处理中
4000	订单支付失败

11.2 客户端错误码

表11-2 客户端错误码

错误代码 (error_code)	含义
9000	订单支付成功
8000	正在处理中
4000	订单支付失败
6001	用户中途取消
6002	网络连接出错

11.3 交易状态

表11-3 交易状态列表

枚举名称	枚举说明
WAIT_BUYER_PAY	交易创建，等待买家付款。
TRADE_CLOSED	<ul style="list-style-type: none">在指定时间段内未支付时关闭的交易；在交易完成全额退款成功时关闭的交易。
TRADE_SUCCESS	交易成功，且可对该交易做操作，如：多级分润、退款等。
TRADE_FINISHED	交易成功且结束，即不可再做任何操作。

11.4 退款状态

表11-4 退款状态

枚举名称	枚举说明
REFUND_SUCCESS	退款成功： <ul style="list-style-type: none">全额退款情况：trade_status= TRADE_CLOSED，而 refund_status=REFUND_SUCCESS非全额退款情况：trade_status= TRADE_SUCCESS，而 refund_status=REFUND_SUCCESS
REFUND_CLOSED	退款关闭