
# 手机网站支付开发指南 PHP

文件版本：2.0.4



2012-04-18

# 前言

## 1. 面向读者

本文档主要面向需要接入支付宝手机网站支付的商户的开发人员。

## 2. 读者所需技能

读者需有 PHP 程序开发背景，掌握 PHP 与 Apache 服务器等相关技能。

## 3. 开发环境要求

本 Demo 在 php 5.2.6 下测试正常，如商户是其他版本，可以自行测试，只要相关扩展库支持即可。

# 目录

# 第一章手机网站支付服务简介

## 1.1 服务介绍

### 1.1.1 Wap 支付

步骤一 ：调用 mobile.merchant.paychannel 接口，查询最近使用支付方式和可用支付前置列表，在页面展现。

步骤二 ： 调用接口 alipay.wap.trade.create.direct， 提交订单信息，获取 token 串。

步骤三 ： 调用接口 alipay.wap.auth.authAndExecute，提交 token 串，跳转到支付宝收银台。

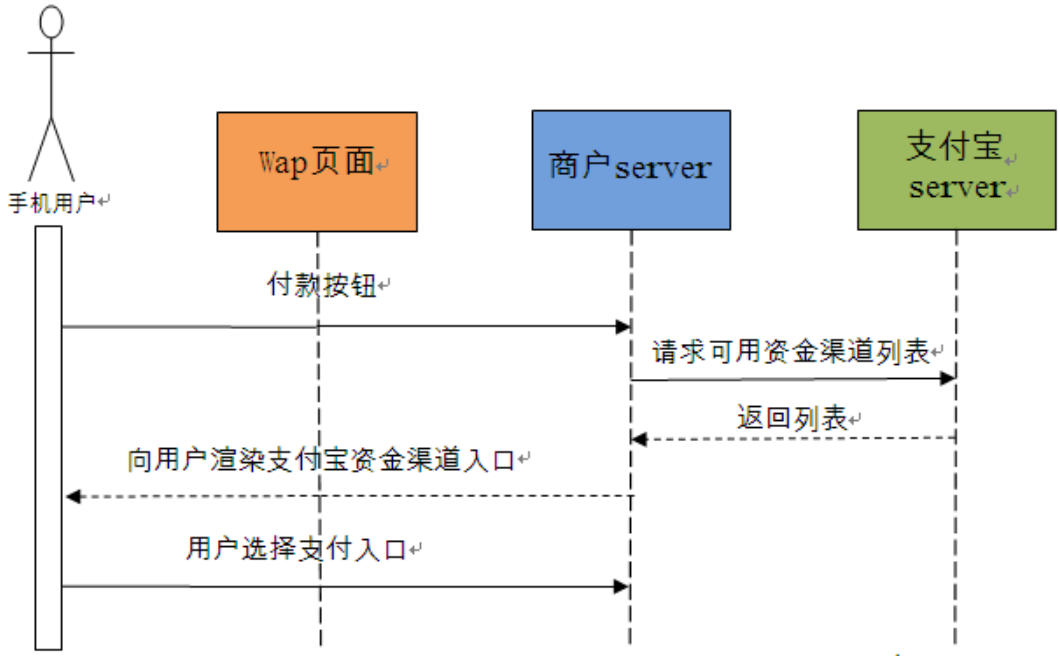
步骤四 ： 处理支付宝系统通知。

基于 http/https 的 请求/响应模式。建议使用 http 请求已适配更多机型。

http 请求地址:http://wappaygw.alipay.com/service/rest.htm

https 请求地址:https://wappaygw.alipay.com:443/service/rest.htm

## 1.2 流程图




图：1-1 流程图

# 第二章接入流程

## 2.1 接入前期准备

接入前期准备工作包括**商户签约**和**密钥配置**，已完成商户可略过。

### 2.1.1 商户签约

首先，商户需要在 https://ms.alipay.com 进行注册，并签约安全支付服务。签约成功后可获取支付宝分配的合作商户 ID(PartnerID)，账户 ID(SellerID)，如图：


图 2-1 商户 ID 获取示意图

签约过程中需要任何帮助请致电：0571-88158090（支付宝商户服务专线）

### 2.1.2 密钥配置

无线商户与支付宝交互加密一共有 2 种形式（MD5、RSA），RSA 的接入难度比 MD5 高，但是也比 MD5 更安全，防抵赖。

签约成功后，商户可登录 https://ms.alipay.com 后点击我的产品->密钥管理来获取商户账号对应的 MD5 Key（选择 MD5 加密的不用看支付宝公钥私钥等相关一切内容，请点击这里）或支付宝公钥（RSA 加密需要用到商户公钥、私钥，支付宝的公钥，具体如何得到，请点击这里）

至此，接入前期准备工作完成，下一节将使用 demo 测试准备工作是否正确。

## 2.2 使用 Demo 测试

### 2.2.1 Demo 配置运行

为了便于商户的接入，我们提供了安全支付 demo。通过本 demo，商户可测试 2.1 节的前期准备工作是否正确完成，同时还可参考 demo 的代码完成接入。以下步骤由 RSA 示范。

**步骤 1**

解压下载的开发资料压缩包 

，点击进去找到 WAP 支付 demo(php 版)文件夹，里面有 2 个加密类型的 Demo，商户一般在集成的时候只需要选择其一。

**步骤 2**

PHP 有很多运行环境，本人是用 WampServer 一站式工具做示范。

把支付宝的 demo 放进 wamp\www 路径

**步骤 3**

打开 Demo 里的 alipay_config.php 文件，在需要商户手动配置的地方填写商户自己的信息。

主要参数说明

*Partner*请参考接入前准备

*PrivateKey*商户私钥

*Alipaypublick*支付宝公钥

*Out_trade_no*外部交易号（每次交易不能重复）

*Seller_account_name*商户收款账号（买家在支付完成后即时到账至该账户）

*Call_back_url*同步跳转通知页面（买家支付完成后，15秒自动跳转，或点击返回跳转，一般只需要美化界面，告知用户交易状态）

*Notify_url*异步跳转通知页面（支付宝发送通知消息给商户服务器的地址，用于商户对该笔订单更新状态等操作，验签通过必须只返回success，不能包含其他文字和任何字符，否则均视为商户返回了fail，请在浏览器源代码中仔细检查）

*Merchant_url*取消支付跳转页面（支付过程中点取消返回的页面）

**步骤4**

本demo默认是支持信用卡快捷的，若商户签约了信用卡快捷支付的本Demo不用再修改，若商户没有签约信用卡快捷支付，请删除alipaychannel.php文件，重命名alipaychannel（储蓄卡）.php文件的文件名为alipaychannel.php

**步骤5**

在PC上模拟手机网页请使用火狐或者Opera，并安装支持wap网页显示的插件，具体不在本指南说明。

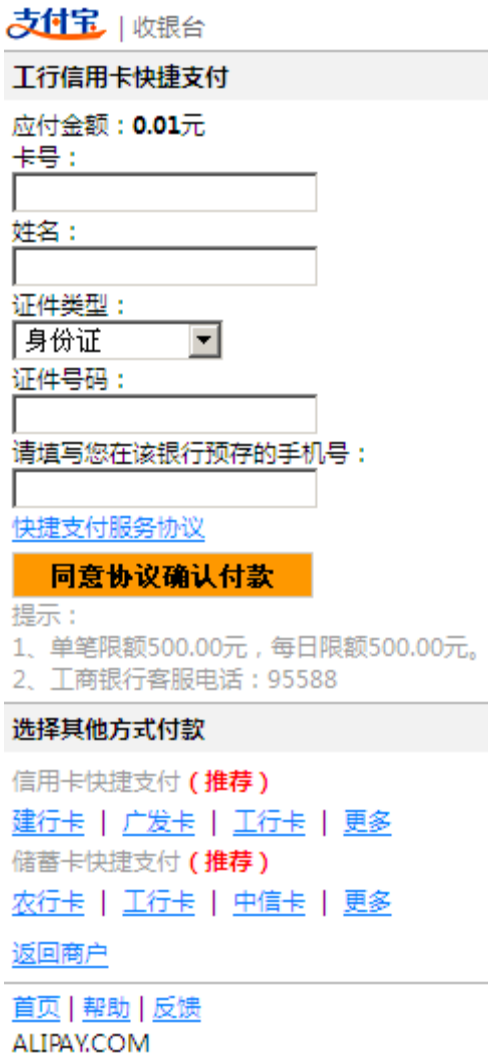
**步骤6**

开启响应的扩展服务，然后在浏览器（选中手机插件模式），界面如下：


图：2-9 模拟商城图：2-10 支付前置展现

有无外部交易号区别在于支付前置列表里是否显示最近支付方式，如图 2-10 是没有外部交易号或者该买家没有上一次支付记录，最后点击支付银行跳转至支付宝收银台，如图：2-11

有些商户由于签约行业是高风险行业，则不会有信用卡支付整块支付渠道，只显示储蓄卡支付渠道，具体事宜请与支付宝商务拓展联系。

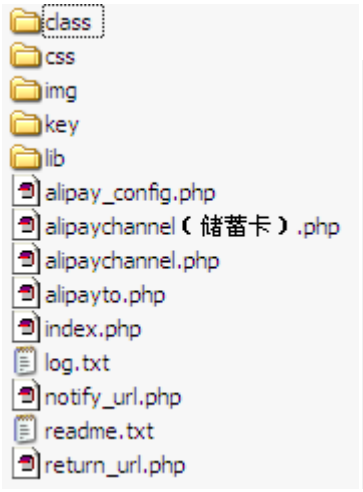



图：2-11 跳转快捷支付收银台

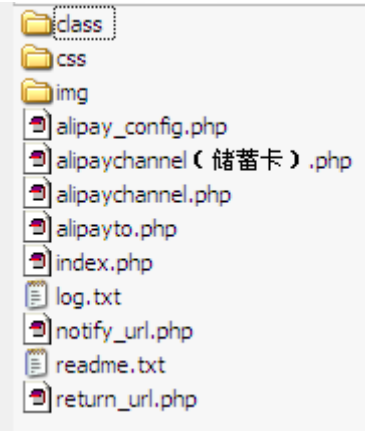
**总结：**

至此，参考本 Demo 的代码能够降低商户集成 wap 支付的难度，并且通过运行本 Demo 可以验证商户公钥私钥和 parent 等信息是否都填写正确。本 Demo 代码仅仅作为参考，只要商户能够实现最终功能，可以随意修改。

## 2.2.2 Demo 结构说明




图：2-12 RSA Demo




图：2-13 MD5 Demo

一些业务逻辑处理类都放在 class 文件夹里，RSA Demo 会多 key 文件夹（用于存放商户私钥、公钥和支付宝公钥）

类文件说明（RSADemo 结构和 MD5Demo 结构是类似的）：

*alipay_config.php*：该类是配置所有请求参数，支付宝网关、接口，商户的基本参数等

*alipay_function.php*：该类是请求、通知返回和 RSA 解密、签名、验签等方法调用类文件

*alipay_service.php*：构造支付宝各接口表单 HTML 文本，获取远程 HTTP 数据

*alipayto.php*：构造请求参数

*return_url.php*：同步返回通知处理页

*Index.php*：Demo 模拟首页

*Notify._url*：异步返回通知处理页面，验签成功后必须只能返回 success，失败则返回 fail

*alipaychannel.php*：支付前置显示页面，类似商户的购物页面

# 2.3 开发

## 2.3.1 获取支付前置列表

为了方便用户购买和增加商户交易成功率，不需要登录支付宝即可进行支付，具体效果请参考 2.2.1 步骤 6

**说明：**

支付前置列表可以放在商品展示的下面，或者在点击购买按钮触发的页面里。支付宝返回给商户是以 JSON 格式的数据，推荐商户使用与 Demo 中相同的文字描述（即返回某信用卡快捷）在 Demo 会处理成某信用卡支付，样式完全由商户自定义。

不支持信用卡快捷的商户，请看2.2.1 步骤4

**步骤1**

创建待签名字符串，格式例如：

如果商户是采用RSA签名，待签名数据不用加以下红色的key参数

```
input_charset=GBK&partner=2088301265823075&out_user=test@test.com&service=mobile.merchant.paychannelqi3ckotphdqr9vkc6az4fuqlh6nso4op
```

字符串以参数名=值表示，多个参数用&分隔，参数名必须按照首字母升序排列，最后加上红色部分的MD5的key参数

**步骤2**

将以上待签名字符串当做参数调用MD5加密方法，得到字符串（也就是MD5签名），例如：a998a9105c6e7a3e446dff05debc3454（如果商户是用RSA的，请调用RSA的签名方法，详情请见RSA签名方法，并加上参数sign_type=0001）

把签名当做sign参数拼装到待签名字符串末尾，如下：

```
input_charset=GBK&partner=2088301265823075&out_user=test@test.com&service=mobile.merchant.paychannel&sign=a998a9105c6e7a3e446dff05debc3454
```

**步骤3**

调用mobile_merchant_paychannel接口，POST或者GET方式请求给支付宝服务器

GET 方式请求样例：

```
https://mapi.alipay.com/cooperate/gateway.do?input_charset=GBK&partner=2088301265823075&out_user=test@test.com&service=mobile.merchant.paychannel&sign=a998a9105c6e7a3e446dff05debc3454
```

POST方式无法在文档中举例，请商户参考Demo实现

**步骤4**

成功返回样例（response）

<?xmlversion="1.0" encoding="GBK" ?>
- <alipay>
<is_success>T</is_success>
- <request>
<param name="service">mobile.merchant.paychannel</param>
<param name="partner">2088102114489547</param>
<param name="out_user">test@test.com</param>
<param name="sign_type">MD5</param>
<param name="sign">70e79c04e17fa1c18786ecf4533ce9a2</param>
</request>
- <response>
- <alipay>
<result>{"payChannleResult":{"lastestPayChannel":{"name":"中行信用卡快捷支付","cashierCode":"CREDITCARD_BOC"},"supportedPayChannelList":{"supportTopPayChannel":[{"name":"信用卡快捷支付","cashierCode":"CREDITCARD","supportSecPayChannelList":{"supportSecPayChannel":[{"name":"深发","cashierCode":"CREDITCARD_SDB"},{"name":"工行","cashierCode":"CREDITCARD_ICBC"},{"name":"建行","cashierCode":"CREDITCARD_CCB"},{"name":"上海农商","cashierCode":"CREDITCARD_SHRCB"},{"name":"宁夏","cashierCode":"CREDITCARD_NXBANK"},{"name":"更多","cashierCode":"CREDITCARD"}]}},{"name":"借记卡快捷支付","cashierCode":"DEBITCARD","supportSecPayChannelList":{"supportSecPayChannel":[{"name":"工行","cashierCode":"DEBITCARD_ICBC"},{"name":"农行","cashierCode":"DEBITCARD_ABC"},{"name":"中行","cashierCode":"DEBITCARD_BOC"},{"name":"上海农商行","cashierCode":"DEBITCARD_SHRCB"},{"name":"建行","cashierCode":"DEBITCARD_CCB"}]}}]}}}</result>
</alipay>
</response>
<sign>66da2223034957428fd5447937a41e23</sign>
<sign_type>MD5</sign_type>
</alipay>

**步骤5**

验签并解析该JSON字符串，通过调用一下方法返回json数据字符串：

```
$result = $alipay->mobile_merchant_paychannel($pms0, $key, $sec_id_MD5);
```

本demo用来实现JSON解析的代码，商户可以自行定义，demo仅作参考：

（alipay_serivce.php）

```
/**
 * 验签并反序列化Json数据
 */
function getJson($result)
```

最后返回：return json_decode($json);

$json是返回的Json部分字符串，在XML的result节点里。

用户在选择银行的时候，商户需要取得cashierCode值，该值用于创建交易并获取token步骤1里实现所选择的银行。

**注意事项：**

支付前置目前暂时只支持 MD5 签名和验签，今后会支持 RSA 签名和验签。

如果没有信用卡支付并不修改 Demo 中代码可能会导致错误，请参考本文档 2.2.1 的步骤 4，返回 JSON 数据如何在商户页面中展现，由商户自己决定样式，但是建议银行支付渠道文字按照 Demo 中例子来规范。

## 2.3.2 创建交易并获取 token

**步骤 1**

创建待签名字符串，格式例如：

```
format=xml&partner=2088301265823075&req_data=<direct_trade_create_req><subject>请输入商品名称
</subject><out_trade_no>030116396</out_trade_no><total_fee>0.01</total_fee><seller_account_name>xxx@sohu.com</seller_account_name><notify_url>http://www.xxx.com/Notify.aspx</notify_url><out_user>xxxyyyzzz</out_user><merchant_url>http://www.xxx.com/Merchant.aspx</merchant_url><cashier_code>CREDITCARD_GDB</cashier_code><call_back_url>http://www.xxx.com/callback.aspx</call_back_url></direct_trade_create_req>&req_id=030116396&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0
```

字符串以参数名=值表示，多个参数用&分隔，sec_id 是加密方式选择，参数名必须按照首字母升序排列如果是 MD5 就填 MD5，如果用 RSA 就填 0001，上面红色部分 cashierCode 参数由支付前置步骤 5 所获取，参数含义请看所有参数列表

**步骤 2**

将以上待签名字符串当做参数调用RSA签名方法，如下：

*function sign($data)*

参数详解：

*$data*：待签名数据，也就是上面方框内的字符串

返回值：签名字符串return $mysgin;就是RSA的签名，例如：

eAYqgDoK77/S2GzsHBalc3ezSPui5EO4uxSk1WgHc33Voc3J2DnUjOCCM7/SyIBJI8vuin/1cTZVC5bL9/+uAVBfnnhC+Zk2Ce0JwqHnSO0eB29/WZrpPml5lccb9u4cDzWx/fsX8nKwQJb3XYuQFOTdc2misnwIr7KRTKyafos=

签名的用到的是商户的私钥，编码是UTF-8，详情请见：class\alipay_function.php

生成好的签名先URLEncode转码，然后当做sign参数拼装到待签名字符串末尾，如下：

```
format=xml&partner=2088301265823075&req_data=<direct_trade_create_req><subject>请输入商品名称
</subject><out_trade_no>030116396</out_trade_no><total_fee>0.01</total_fee><seller_account_name>xxx@sohu.com</seller_account_name><notify_url>http://www.xxx.com/Notify.aspx</notify_url><out_user>xxxyyyzzz</out_user><merchant_url>http://www.xxx.com/Merchant.aspx</merchant_url><cashier_code>CREDITCARD_GDB</cashier_code><call_back_url>http://www.xxx.com/callback.aspx</call_back_url></direct_trade_create_req>&req_id=030116396&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0&sign=eAYqgDoK77%2fS2GzsHBalc3ezSPui5EO4uxSk1WgHc33Voc3J2DnUjOCCM7%2fSyIBJI8vuin%2f1cTZVC5bL9%2f%2buAVBfnnhC%2bZk2Ce0JwqHnSO0eB29%2fWZrpPml5lccb9u4cDzWx%2ffsX8nKwQJb3XYuQFOTdc2misnwIr7KRTKyafos%3d
```

**步骤3**

调用mobile_merchant_paychannel接口，POST或者GET方式请求给支付宝服务器

http 请求地址:http://wappaygw.alipay.com/service/rest.htm

https 请求地址:https://wappaygw.alipay.com:443/service/rest.htm

GET 方式请求样例：

```
http://wappaygw.alipay.com/service/rest.htm?call_back_url=http://www.xxx.com/Call_Back.aspx&format=xml&partner=2088301265823075&req_data=<direct_trade_create_req><subject>请输入商品名称
</subject><out_trade_no>030116396</out_trade_no><total_fee>0.01</total_fee><seller_account_name>xxx@sohu.com</seller_account_name><notify_url>http://www.xxx.com/Notify.aspx</notify_url><out_user>xxxyyyzzz</out_user><merchant_url>http://www.xxx.com/Merchant.aspx</merchant_url><cashier_code>CREDITCARD_GDB</cashier_code></direct_trade_create_req>&req_id=030116396&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0&sign=sign=eAYqgDoK77%2fS2GzsHBalc3ezSPui5EO4uxSk1WgHc33Voc3J2DnUjOCCM7%2fSyIBJI8vuin%2f1cTZVC5bL9%2f%2buAVBfnnhC%2bZk2Ce0JwqHnSO0eB29%2fWZrpPml5lccb9u4cDzWx%2ffsX8nKwQJb3XYuQFOTdc2misnwIr7KRTKyafos%3d
```

POST 方式无法在文档中举例，请商户参考 Demo 实现

**步骤 4**

说明：

商户通过上面步骤请求之后，支付宝会返回数据给商户，商户主要是得到 token 值，返回样例例举以下 3 种：（1、2 为成功，3 为失败）

以下验签规则：返回数据中除了 sign 之外的其他参数都要通过升序排列，然后调用验签方法

1、当商户使用 **RSA** 签名方式时，实际返回的内容如下（其中 res_data 参数值为加密内容（红色部分），商户必须先 URLDecode，然后用商户的 RSA 私钥解密，最后验签）：

```
res_data=Cl2Mm1Z2YILG8oYe8%2FngEAvYSM9YYmcqUqLtUCZ10habqYb6poowofjzVG3nsUJY6qlgnRrq%2FxFTtdLdwBDGltV8rwpf1AFB01ydCanpQoFgQg%2Brt79JRQ%2B9CC3E%2Fg148C4F95eJ1FNf0L6taXaMFwxaxrvTAdDHzzvSigy3%2BaKdFh8z2K1Zs4gm2bD39lR1CRXSipOyVFhCZZR9L9N8tQNZbDqnyBu%2FjLdLbvXvEuE4fImZPPbsALecVCvsHL4iKFrquPnhA4Zz%2FZEM%2FoJghXA6xlAO0a1d0h6Os%2Fd83mvDPfmhs3oVjPX3FsXCL18Dg4mdzj3gWllbqLnwammM94g%3D%3D&service=alipay.wap.trade.create.direct&sec_id=0001&partner=2088201747196380&req_id=1288337908547&sign=RiyyndPEei2QQc%2FHt1%2FIrmYyW6%2FFKNZFxPUiOcXndAOo3OifNRshRjaLlwEs3d2pBpbmyclfooF7tctFdXcrSM584wgsY%2Bj2o0Z6dXst9lmz%2F4OD%2BL2ubk1DXoLWau0f5NiteIuGqGDWUdXMKRLx1FJ0f%2FMN8GOCUZYN15%2FUE%2FE%3D&v=2.0
```

2、当商户使用**MD5**签名方式时，实际返回的内容如下（其中res_data参数值为明文内容，无需解密，直接验签）

```
partner=2088101000137799&req_id=1283133204160&res_data=<?xmlversion="1.0" encoding="utf-8"?><direct_trade_create_res><request_token>20100830e8085e3e0868a466b822350ede5886e8</request_token></direct_trade_create_res>&sec_id=MD5&service=alipay.wap.trade.create.direct&v=2.0&sign=72a64fb63f0b54f96b10cefb69319e8a
```

3、失败返回样例：

失败的 detail 里面会有各种信息，这里的示例是没有开通接口权限的错误，其他错误请商户检查以上步骤是否正确并修改重新提交。

失败返回无论哪种签名方式，内容都是明文无需解密。

```
partner=208810100013779&req_id=1283133132946&res_error=<?xml version="1.0" encoding="utf-8"?><err><code>0005</code><sub_code>0005</sub_code><msg>partner illegal</msg><detail>合作伙伴没有开通接口访问权限</detail></err>&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0
```

**步骤 5**

按照以上**步骤 4** 的 1 和 2 样例验签通过了之后，解析 XML 得到 token 字符串，并做为参数调用以下接口。

## 2.3.3 授权并执行

**步骤 1**

创建待签名字符串，格式例如：

```
call_back_url=http://10.2.46.218/Call_Back.aspx&format=xml&partner=2088301265823075&req_data=<auth_and_execute_req><request_token>201110259f7686ab763c20e630db9902166f0bfa</request_token></auth_and_execute_req>&sec_id=0001&service=alipay.wap.auth.authAndExecute&v=2.0
```

字符串以参数名=值表示，多个参数用&分隔，<request_token>里面填上面返回的 token 字符串。

**步骤 2**

将以上待签名字符串当做参数调用RSA签名方法（具体实现请参考2.3.2 步骤二），如下：

*function sign($data)*

生成好的签名先 URLEncode 转码，然后当做 sign 参数拼装到待签名字符串末尾，如下：

```
call_back_url=http://10.2.46.218/Call_Back.aspx&format=xml&partner=2088301265823075&req_data=<auth_and_execute_req><request_token>201110259f7686ab763c20e630db9902166f0bfa</request_token></auth_and_execute_req>&sec_id=0001&service=alipay.wap.auth.authAndExecute&v=2.0&sign=A1LhhVwoCHT9yVtKKdBLtcwFYbIlAlWO28stm8vuFYwZ%2bcYcT4XMSW5UMVOCbzBZQ76Go04AriB78LPbo%2fAhNO4nxYL%2fJs7rbymQtvVXRGaqtgrMu1JMWpDxUSyoqACPmyusG9OvXztXVjzbfquG2BVKfclYcEGOzF1WDiHOMjw%3d
```

**步骤 3**

以上地址加上请求网关，如下：

```
http://wappaygw.alipay.com/service/rest.htm?call_back_url=http://10.2.46.218/Call_Back.aspx&format=xml&partner=2088301265823075&req_data=<auth_and_execute_req><request_token>201110259f7686ab763c20e630db9902166f0bfa</request_token></auth_and_execute_req>&sec_id=0001&service=alipay.wap.auth.authAndExecute&v=2.0&sign=A1LhhVwoCHT9yVtKKdBLtcwFYbIlAlWO28stm8vuFYwZ%2bcYcT4XMSW5UMVOCbzBZQ76Go04AriB78LPbo%2fAhNO4nxYL%2fJs7rbymQtvVXRGaqtgrMu1JMWpDxUSyoqACPmyusG9OvXztXVjzbfquG2BVKfclYcEGOzF1WDiHOMjw%3d
```

Header跳转该地址（即：支付宝wap收银台地址）

## 2.4 处理支付宝系统通知

支付宝系统的通知包括同步和异步两种方式，同步是指在支付完成后支付宝直接调用商户指定的 call_back_url，并携带参数；异步是指支付宝在支付完成后发送通知到商户指定的 notify_url，以下为具体内容。

## 2.4.1 call_back_url

用户在支付宝收银台完成支付后，会以 GET 方式跳转到 call_back_url（用户直接点击或自动跳转），同时会携带交易参数。商户在收到这一参数后，要先进行验签。样例如下：

**样例**

http://10.14.42.49:8080/paychannel/servlet/CallBack?out_trade_no=1320742949342&request_token=requestToken&result=success&trade_no=2011110823389231&sign=49a330fee069465c64e561a25bf31c78

商户可根据“result”参数判断交易状态。具体参数的含义请查询参数表

## 2.4.2 notify_url

在交易完成后，支付宝通过访问商户提供的地址的形式，将交易状态信息发送给商户服务器。商户通过支付宝的通知判断交易是否成功，具体如下：

**商户地址：**提供一个 http 的 URL(例:http://www.partnertest.com/servlet/NotifyReceiver)，支付宝将以 POST 方式调用该地址。

**通知触发条件：**交易状态发生改变，如交易从“创建”到“成功”或“关闭”。

**商户返回信息：**商户服务器收到通知后需返回纯字符串“success”，不能包含其他任何 HTML 等语言的文本。

**通知重发：**若支付宝没有收到商户返回的“success”，将对同一笔订单的通知进行周期性重发 (间隔时间为：2 分钟,10 分钟,10 分钟,1 小时,2 小时,6 小时,15 小时共 7 次)。

**交易判断条件：**收到 trade_status=TRADE_FINISHED（如果签有高级即时到帐协议则 trade_status=TRADE_SUCCESS）的请求后才可判定交易成功（其它 trade_status 状态请求可以不作处理）

**以下为支付宝通知的样例：**

1） RSA 方式签名时

当商户使用 RSA 签名方式时，商户实际收到支付宝通知请求如下：

**样例：**

http://www.partnertest.com/servlet/NotifyReceiver?service=alipay.wap.trade.create.direct&sign=Rw/y4ROnNicXhaj287Fiw5pvP6viSyg53H3iNiJ61D3YVi7zGniG2680pZv6rakMCeXX++q9XRLw8Rj6I1//qHrwMAHS1hViNW6hQYsh2TqemuL/xjXRCY3vjm1HCoZOUa5zF2jU09yG23MsMIUx2FAWCL/rgbcQcOjLe5FugTc=&sec_id=0001&v=1.0&notify_data=g3ivqicRwI9rI5jgmSHSU2osBXV1jcxohapSAPjx4f6qiqsoAzstaRWuPuutE0gxQwzMOtwL3npZqWO3Z89J4w4dXIY/fvOLoTNn8FjExAf7OozoptUS6suBhdMyo/YJyS3lVALfCeT3s27pYWihHgQgna6cTfgi67H2MbX40xtexIpUnjgxBkmOLai8DPOUI58y4UrVwoXQgdcwnXsfn2OthhUFiFPfpINgEphUAq1nC/EPymP6ciHdTCWRI6l1BgWuCzdFy0MxJLliPSnuL

yZTou7f+Z5Mw24FgOacaISB+1/G+c4XIJVKJwshCDw9Emz+NAWsPvq34FEEQXVAeQRDOphJx8bDqLK75CGZX+6fx88
m5ztq4ykuRUcrmoxZLJ+PiABvYFzi5Yx2uBMP/PmknRmj1HUKEhuVWsXR0t6EWpJFXlyQA4uxbShzncWDigndD7wbf
NtkNLg5xMSFFIKay+4YzJK68H9deW4xqk4JYTKsv8eom9Eg9MrJZiIrFkFpVYPuaw0y/n61UEFYdzEQZz+garCmMYehE
AQCGibYUQXBlf1iwTOZdqJIxdgCpSX21MIa9N9jicmFu8OXWZJkdN+UrSyvIcpzRori+U6522ovMz5Z8EzVTfcUENu+d
WJRnhFlo6pvm0a3Fq2wBEyUV1/YYS3LaZiPj+wig5BCyJ92QXZnEUEtn87oX5kuzSRuLcVVi8OJIgyQWaWT9N0YFyH5
AfV+VDNxu4UYy6KkGtcaVjSvbzDuzThMXs2HDwX3qujq25A/hzJKlgR9EjcumJeF/TM6eS7JS+FKXE1kUXnMnGbokaN
emZn2yKlPC1VO4LU77G5v1nUs6MfYFq9HC4FYiQ6Y+hL8RgAMorty/RYT3cZ8SQCTO0bQ+qJuOnx79YEEmCUQc3iJB
p0zFKYXIU6viqJYghEs6F3LiK8TvJRo8+ST5hKtnuU5b8R6f9yD8Uek1BruWvlYaA7I3Cc90CDhTyOghL2oCMOoKlxqgXd
h3MGm128FOVyCjDLRw04b+kK83JGFMcjyVuhfhoVeETQicUCtFQ9ItlH3uFkB5su+r3399WGSXyGflrTbFhMq7mRzt
WotL2ATvf/enMBcGSCSCb47OzGxXhMDGZZu4Sq4pdF9fsZVBHgWsB/KS8bwxyvM068NoqnRmI72zgL7WFWumlm8
8j3K6KPxbB6soDSXRv6drbSv2t93llE5q4SP6GLztAw7UPWGTJLXOFyhyaszvhyZWxsX+C5PbXoCta1/cxt4Sp4WXDjaHn
6qHI/Vea28xx8fYV/xK5WTmvFwb0k9eRGCgB6/nzmGV1+lPJuK3pKy3L5LbUP0zJFh5gdPG7DecH+F0uBUC0QNMQ=
=

其中 notify_data 参数值为加密内容，商户需用商户 RSA 私钥先进行解密后再验签，验签具体请见这里 注意：支付宝系统通知待签名数据构造规则比较特殊，为固定顺序。

2） MD5 方式签名时

当商户使用 MD5 签名方式时，商户实际收到支付宝通知请求如下

**样例：**

http://www.partnertest.com/servlet/NotifyReceiver?service=alipay.wap.trade.create.direct&sign=Rw/y4ROnNicX
haj287Fiw5pvP6viSyg53H3iNiJ61D3YVi7zGniG2680pZv6rakMCeXX++q9XRLw8Rj6I1//qHrwMAHS1hViNW6hQYsh2
TqemuL/xjXRCY3vjm1HCoZOUa5zF2jU09yG23MsMIUx2FAWCL/rgbcQcOjLe5FugTc=&v=1.0&sec_id=MD5&notify_
data=<notify><payment_type>1</payment_type><subject>收银台【1283134629741】</subject><trade_no>20
10083000136835</trade_no><buyer_email>dinglang@a.com</buyer_email><gmt_create>2010-08-30
10:17:24</gmt_create><notify_type>trade_status_sync</notify_type><quantity>1</quantity><
out_trade_no>1283134629741</out_trade_no><notify_time>2010-08-30 10:18:15</notify_time><
seller_id>2088101000137799</seller_id><trade_status>TRADE_FINISHED</trade_status><is_total_fee_adjust>N
</is_total_fee_adjust><total_fee>1.00</total_fee><gmt_payment>2010-08-30 10:18:26</gmt_payment><seller
_email>chenf003@yahoo.cn</seller_email><gmt_close>2010-08-30 10:18:26</gmt_close><price>1.0
0</price><buyer_id>2088102001172352</buyer_id><notify_id>509ad84678759176212c247c46bec05303</notif
y_id><use_coupon>N</use_coupon></notify>

其中 notify_data 参数值为明文内容，无需解密。

通知中其他参数意义详见参数列表

# 第三章签名详解

## 3.1RSA 和 openssl 简介

### 3.1.1 什么是 RSA

RSA 是一种非对称的签名算法，即签名密钥（私钥）与验签密钥（公钥）是不一样的，私钥用于签名，公钥用于验签。

在与支付宝交易中，会有 2 对公私钥，即商户公私钥，支付宝公私钥。

### 3.1.2 为什么要用 RSA

使用这种算法可以起到防止数据被篡改的功能，保证支付订单和支付结果不可抵赖(商户私钥只有商户知道)。

### 3.1.3 什么是 OpenSSL

一句话概括：OpenSSL 是基于众多的密码算法、公钥基础设施标准以及 SSL 协议安全开发包。

### 3.1.4 为什么要用 OpenSSL

通过 OpenSSL 生成的签名和内置的算法可以做到跨平台，这样在不同的开发语言中均可以签名和验签。

## 3.2 RSA 密钥详解*

### 3.2.1 找到生成 RSA 密钥工具

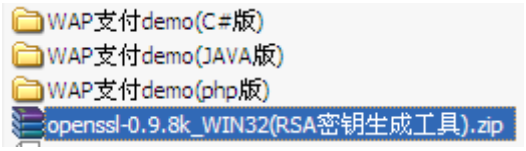
**步骤 1**

下载开发指南和集成资料，如下图，您能看到此文档说明开发指南和集成包已经下载了。


图：3-1 下载开发指南和集成资料

**步骤 2**

解压下载的压缩包(WS_WAP_PAYWAP)，找到并解压 openssl-0.9.8k_WIN32(RSA 密钥生成工具).zip 工具包




图：3-2 openssl 工具包

## 3.2.2 生成商户密钥并获取支付宝公钥

**(1) 生成原始 RSA 商户私钥文件**

假设解压后的目录为 c:\alipay，命令行进入目录 C:\alipay\bin，执行“*openssl genrsa -out rsa_private_key.pem 1024*”，在 C:\alipay\bin 下会生成文件 rsa_private_key.pem，其内容为原始的商户私钥（请妥善保存该文件），以下为命令正确执行截图：

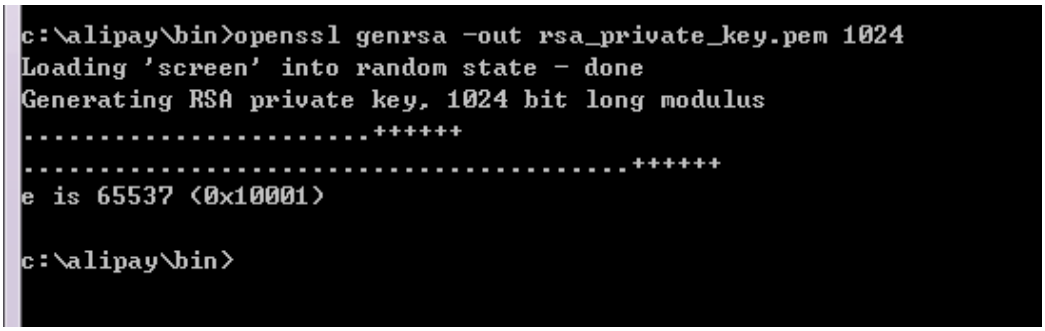



图 3-3 生成原始 RSA 商户私钥文件

**(2) 将原始 RSA 商户私钥转换为 pkcs8 格式**

命令行执行“ *openssl pkcs8 -topk8 -inform PEM -in rsa_private_key.pem -outform PEM*

–*nocrypt*”得到转换为 pkcs8 格式的私钥。复制下图红框内的内容至新建 txt 文档，去掉换行，最后另存为“private_key.txt”（请妥善保存，签名时使用）。

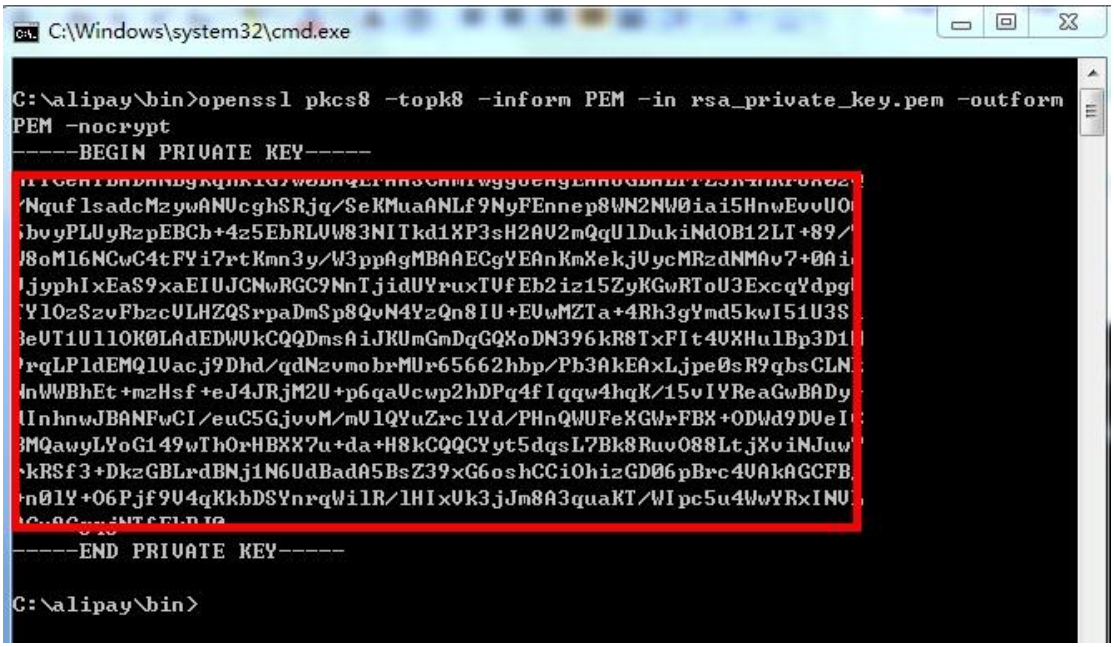



图 3-4 转换私钥格式

**(3) 生成 RSA 商户公钥**

命令行执行“ *openssl rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem*”，在 C:\alipay\bin 文件夹下生成文件 rsa_public_key.pem。接着用记事本打开 rsa_public_key.pem，复制全部内容至新建的 txt 文档，删除文件头“-----BEGIN PUBLIC KEY-----”与文件尾“-----END PUBLIC KEY-----”及空格、换行，如下图。最后得到一行字符串并保存该 txt 文件为“public_key.txt”。

```
1  -----BEGIN PUBLIC KEY-----
2  MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCuQBQMjMX+ossoDXoi5DlcDOsf
3  6hVT6twgwfuVbyouTSI/cHjH2xpu1S/RD4xXHBi/60GNmewAro2T70i1wxuMpgcD
4  S+3S/0z+4xyrW8ewXfeGmUVPKlyPbkmFeL/OuKWNdhpObOmCyByZPts0lkFKDPb9
5  B5lxZQzj6b+82L31kQIDAQAB
6  -----END PUBLIC KEY-----
```

去掉头尾注释、换行、空格

```
IGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCuQBQMjMX+ossoDXoi5DlcDOsf6hVT6twgwfuVbyouTSI/cHjH2xpu1S
```

图 3-5 生成公钥

**(4) 上传商户公钥至支付宝**

浏览器访问 https://ms.alipay.com/index.htm 并用签约帐号登录，点击菜单栏“我的产品”，右侧点击“密钥管理”，见下图红色框内


图 3-6 商户公钥上传

点击“上传”，选择步骤(3)生成的“public_key.txt”并完成上传。

**(5) 获取 RSA 支付宝公钥**

成功上传公钥至支付宝后，页面显示如下：


图 3- 7 支付宝公钥获取

其中红色框内部分即支付宝公钥，请复制至新建 txt 文档，去掉换行和空格，妥善保存（用于验签收到的支付宝通知）。

# 3.3 RSA 签名和验签 *

## 3.3.1 RSA 签名

**定义：**

RSA 是一种非对称的签名算法，即签名密钥（私钥）与验签名密钥（公钥）是不一样的，私钥用于签名，公钥用于验签名。使用这种算法签名在起到防数据篡改功能的同时，还可以起到防抵赖的作用，因为私钥只有签名者知道。

核心代码是调用 RSA.cs 文件中的 sign 方法

创建签名用的是商户的私钥

```
/**RSA 签名
 * $data 签名数据(需要先排序，然后拼接)
 *  签名用商户私钥，必须是没有经过 pkcs8 转换的私钥
 *  最后的签名，需要用 base64 编码
 * return Sign 签名
 */
function sign($data) {
    //读取私钥文件
    $priKey = file_get_contents('key/rsa_private_key.pem');

    //转换为 openssl 密钥，必须是没有经过 pkcs8 转换的私钥
    $res = openssl_get_privatekey($priKey);

    //调用 openssl 内置签名方法，生成签名$sign
openssl_sign($data, $sign, $res);

    //释放资源
openssl_free_key($res);

    //base64 编码
    $sign = base64_encode($sign);
    return $sign;
}
```

## 3.3.2RSA 验签

核心代码是调用 RSA.cs 文件中的 verify 方法

验签方法中用的是支付宝公钥，关于支付宝公钥哪里来，请点击这里

```
/**RSA 验签
 * $data 待签名数据(需要先排序，然后拼接)
 * $sign 需要验签的签名,需要 base64_decode 解码
 * 验签用支付宝公钥
 * return 验签是否通过 bool 值
 */
function verify($data, $sign)  {
    //读取支付宝公钥文件
    $pubKey = file_get_contents('key/alipay_public_key.pem');

    //转换为 openssl 格式密钥
    $res = openssl_get_publickey($pubKey);

    //调用 openssl 内置方法验签，返回 bool 值
    $result = (bool)openssl_verify($data, base64_decode($sign), $res);

    //释放资源
openssl_free_key($res);

    //返回资源是否成功
    return $result;
}
```

## 3.3.3 RSA 解密

核心代码是调用 RSA.cs 文件中 decryptData 方法

```
/**解密
 * $content为需要解密的内容
 * 解密用商户私钥
 * 解密前，需要用base64将内容还原成二进制
 * 将需要解密的内容，按128位拆开解密
 * return 解密后内容，明文
 */
function decrypt($content) {
```

```
    //读取商户私钥
    $priKey = file_get_contents('key/rsa_private_key.pem');

    //转换为openssl密钥，必须是没有经过pkcs8转换的私钥
    $res = openssl_get_privatekey($priKey);

    //密文经过base64解码
    $content = base64_decode($content);

    //声明明文字符串变量
    $result  = '';

    //循环按照128位解密
    for($i = 0; $i <strlen($content)/128; $i++  ) {
        $data = substr($content, $i * 128, 128);

        //拆分开长度为128的字符串片段通过私钥进行解密，返回$decrypt解析后的明文
openssl_private_decrypt($data, $decrypt, $res);

        //明文片段拼接
        $result .= $decrypt;
    }

    //释放资源
openssl_free_key($res);

    //返回明文
    return $result;
}
```

# 3.4 MD5

## 3.4.1 MD5 简介

**定义：**

MD5 是一种摘要生成算法，本来是不能用于签名的。但是，通过在待签名数据之后加上一串私密内容（指令发送、接收双方事先规定好的，这里我们称其为签名密钥），就可以用于签名了。使用这种算法签名只能起到防数据篡改的功能，不能起到签名防抵赖的功能，因为双方都知道签名密钥

## 3.4.2 MD5 Key

当商户使用 MD5 加密方式生成签名之前，需要将待签名参数加上 MD5 Key 参数。

获取 Key：登录 https://ms.alipay.com 我的产品->密钥管理，然后复制出来 MD5 密钥字符串


图：3-8 复制 MD5 密钥

## 3.4.3 MD5 签名和验签

签名：即调用 AlipayFunction.cs 文件中 MD5Sign 方法，切忌不要用.NET 自带的 MD5 加密，因为和其他语言平台加密大小写不同。

```
/**MD5 签名方法
 * $prestr 需要签名的字符串
 * $sign_type 签名类型，也就是 sec_id
 * return 签名结果
 */
function sign_MD5($prestr,$sign_type) {
    $sign='';
    if($sign_type == 'MD5') {
        $sign = md5($prestr);
    }elseif($sign_type =='DSA') {
        //DSA 签名方法待后续开发
        die("DSA 签名方法待后续开发，请先使用 MD5 签名方式");
    }else {
        die("支付宝暂不支持".$sign_type."类型的签名方式");
    }
    return $sign;
}
```

验签就是把支付宝返回的数据（除签名）进行签名，并对比返回的 sign 签名，如果相同代

表验签通过，否则验签没有通过，可能表单已经被篡改。

## 3.5 签名规范

为了确保数据传输过程中的数据真实性和完整性，支付宝和商户都需要对request/response 数据进行签名验证。目前本接口支持的签名算法为 MD5、RSA。

# 第四章常见问答

**1、支付完成之后 Notify_url 接收不到支付宝异步请求的通知？**

有以下几种可能：

a. Notify_url 不是一个能够公开访问的地址

b. 防火墙、白名单的问题（建议暂时关闭防火墙试试，或者配置下白名单）

**2、验签，报“订单信息被篡改”是什么问题？**

可能有以下 2 种情况

a) 有可能数据在传输过程中被黑客截取和篡改

b) 检查待签名字符串中的参数值是否有以下四个符号，如果参数当中包含了这四个字符也会报“订单信息被篡改”：

+加号

&连接符

“双引号

=等号

**3、 上传商户公钥报格式错误怎么办？**

首先确认上传的位置是否是RSA的下面，注意不要是DSA，无线目前不支持DSA加密；

另外请检查上传的文件中是否去除注释、空格、换行等，必须是一行的字符串

**4、当输入付款账号和支付密码后，支付宝收银台报“请求出错!”的提示？**

a) 请把demo中所有的参数都加上(Notify_url、Call_back_url、Out_user等都请填上)

b) seller_account_name请不要填2088开头的商户号，请填写支付宝账号(邮件或者手机号格式)

# 附录 A 错误代码列表

| 错误代码 | 说明 |
|---|---|
| 0000 | 系统异常 |
| 0001 | 缺少必要的参数，检查非空参数是否已经传递 |
| 0002 | 签名错误，检查签名的参数是否符合支付宝签名规范 |
| 0003 | 服务接口错误，检查 service 是否传递正确 |
| 0004 | req_data 格式不正确 |
| 0005 | 合作伙伴没有开通接口访问权限，合同是否有效 |
| 0006 | sec_id 不正确，支持 0001，MD5 |
| 0007 | 缺少了非空的业务参数 |
| ILLEGAL_SIGN | 签名错误，检查签名的数据是否符合支付宝签名规范 |
| ILLEGAL_SERVICE | 接口不存在，检查 service 是否传递正确 |
| ILLEGAL_PARTNER | 无效商户,检查传入的 PARTNER 值是否正确 |
| ILLEGAL_PARTNER_EXTERFACE | 商户接口不存在，该商户没有开通该接口 |
| HAS_NO_PRIVILEGE | 无权访问该接口 |
| SYSTEM_ERROR | 系统错误 |

# 附录 B 手机网站支付接口参数表

| 参数名 | 中文描述 | 类型(精度) | 说明 | 商户必传 | 参数值样例 |
|---|---|---|---|---|---|
| service | 接口名称 | String | 注意：交易创建、授权并执行两次请求的值不同。 | Y | alipay.wap.trade.create.direct/alipay.wap.auth.authAndExecute |
| partner | 合作伙伴id | String(16) | 合作伙伴在支付宝的用户 ID，与支付宝签约后自动生成 | Y | 2088002007015955 |
| sec_id | 签名算法 | String(4) | 签名算法。目前只支持 MD5 和 RSA(用 0001 表示) | Y | 0001 或 MD5 |
| req_id | 请求号 | String(32) | 请求号用于关联请求与响应，并且防止请求重播。支付宝 wap 限制来自一个 partner 的请求号必须唯一。 | Y | 1e925b9b4b115961660130f9281e3898 |

| | | | | | |
|---|---|---|---|---|---|
| sign | 签名 | String | 签名，对 request/response 中参数签名后的值 | Y | 72020eb70e0fdcfbf404edcbb83bfd81 |
| format | 请求参数格式 | String | 参数值必须和样例保持一致 | Y | xml |
| v | 接口版本号 | String | 参数值必须和样例保持一致 | Y | 2.0 |
| req_data | 请求业务参数 | String | 参数值内容为 xml 格式,包含内层标签参数 | Y | <?xml version="1.0" encoding="UTF-8"?><direct_trade_create_req> <subject>彩票</subject><out_trade_no>20080801-1</out_trade_no><total_fee>50</total_fee><seller_account_name>tbbusi003@126.com</seller_account_name><out_user>xxxxx</out_user><notify_url>http://www.yoursite.com/notifyurl.htm</notify_url></direct_trade_create_req> |
| direct_trade_create_req | 固定标签 | String | req_data 参数值 xml 内容中必须包含的固定标签。 | Y | <subject>彩票</subject><out_trade_no>20080801-1</out_trade_no><total_fee>50</total_fee><seller_account_name>tbbusi003@126.com</seller_account_name><out_user>xxxxx</out_user><notify_url>http://www.yoursite.com/notifyurl.htm</notify_url> |
| subject | 商品名称 | String(256) | 订单商品名称 | Y | 彩票 |
| out_trade_no | 外部交易号 | String(64) | 合作伙伴系统的交易号，传递给支付宝系统做外部交易号（不能重复） | Y | 2008080101 |
| total_fee | 订单价格 | String(15) | 用户购买的商品或服务的价格（必须是金额的格式,单位：元） | Y | 1.01 |
| pay_expire | 交易自动关闭时间 | Int | 买家如未能在该设定值范围内支付成功，交易将被关闭。单位：“分钟”，值区间 0<pay_expire，默认值 21600（15 天）。最终关闭时间点误 | N | 10 |

| | | | | | |
|---|---|---|---|---|---|
| | | | 差 1-2 分钟。 | | |
| seller_account_name | 卖家帐号 | String(100) | 交易卖方的支付宝帐号，交易成功后该笔交易的资金会转入这个支付宝帐号中 | Y | tbbusi003@126.com |
| out_user | 商户系统用户唯一标识 | String(32) | 买家在商户系统的唯一标识，当该 out_user 支付成功一次后再来支付时，30 元内无需密码。 | N | 21321211111 |
| notify_url | 商户接受通知的 url | String(200) | 商户接受通知的 url | Y | http://www.yoursite.com/notifyurl.htm |
| merchant_url | 返回商户链接 | String | 用户在支付宝页面可返回商户的链接 | N | http://www.yoursite.com/partnerurl.htm |
| call_back_url | 支付成功跳转链接 | String(200) | 由商户提供，只有当交易支付成功之后，才会跳转到该 url。 | Y | http://www.yoursite.com/callbackurl.htm |
| cashierCode | 支付前置银行代码 | String | 调用支付前置接口，由支付宝返回给商户所支持的银行代码 | N | CREDITCARD_ICBC |
| request_token | token | String(40) | 前面调用交易创建接口成功返回后获得的**（注当此参数为页面返回时，为固定值）** | | 20081113f9d49c20e8e5c8e40b6107ec42259e41 |
| trade_no | 交易号 | String(64) | 交易号，该笔交易在支付宝系统的交易号 | | 2009092904171521 |
| notify_data | 通知业务参数 | String | 通知的业务参数，包含交易号、外部交易号、交易状态等信息。 | | 见例子 |
| payment_type | 支付方式 | String | 用户的支付方式(商户可不关心该参数) | | 1 |
| buyer_email | 买家账号 | String(100) | 买家的支付宝账号 | | chenf002@yahoo.cn |
| gmt_create | 创建时间 | String | 交易创建时间 | | 2009-09-29 19:59:24 |
| notify_type | 通知类型 | String | 该通知的类型，暂时只有交易状态同步(商户可不关心该参数) | | trade_status_sync |
| quantity | 数量 | String | 购买商品数量 | | 1 |
| notify_time | 通知时间 | String | 发送通知的时间 | | 2009-09-29 19:59:25 |
| seller_id | 卖家 id | String | 卖家的支付宝账号 id | | 2088102001058148 |

| | | | | | |
|---|---|---|---|---|---|
| trade_status | 交易状态 | String | 交易的状态。TRADE_FINISHED（支付成功），WAIT_BUYER_PAY(等待买家付款) | | TRADE_FINISHED/ WAIT_BUYER_PAY |
| is_total_fee_adjust | 总价是否被修改 | String | 交易价格是否被修改，Y 或 N(本接口创建的交易不会被修改) | | N |
| total_fee | 交易总价 | String | 即订单金额。单位：元 | | 2.21 |
| gmt_payment | 付款时间 | String | 交易的付款时间，如果交易未付款，没有该属性 | | 2009-09-29 19:59:25 |
| seller_email | 卖家账号 | String(100) | 卖家的支付宝账号 | | youngbeckham@gmail.com |
| gmt_close | 交易结束时间 | String | 交易结束的时间 | | 2009-09-29 19:59:25 |
| price | 单个商品价格 | String | 目前和 total_fee 值相同。单位：元 | | 2.21 |
| buyer_id | 买家 id | String | 买家的支付宝账号 id | | 2088101000137393 |
| notify_id | 通知 id | String | 唯一识别通知内容，重发相同内容的通知 notify_id 值不变。 | | 2311b764be6fba98f593ba98f7eb7470 |
| use_coupon | 是否使用红包 | String | 交易时是否使用红包，Y 或 N | | N |
| _input_charset | 参数编码字符集 | String | 见签名机制 | N | GBK |