



# 接入与使用规则

---

## 国际卡网关支付接口

**(alipay.trade.direct.forcard.pay)**

## 附录文档

版本号：1.2

## 目 录

<b>1 文档说明</b>	<b>3</b>
1.1 前言	3
1.2 服务介绍-国际卡支付	3
1.3 业务术语	3
<b>2 责任归属</b>	<b>4</b>
<b>3 接入流程</b>	<b>4</b>
3.1 接入前期准备	4
3.2 获取支付宝分配的合作商户ID和安全校验码	4
3.3 开发	6
<b>4 测试流程</b>	<b>9</b>
4.1 测试部署	9
4.2 页面跳转案例	9
<b>5 技术接入规则</b>	<b>13</b>
<b>6 接口使用规则</b>	<b>19</b>

# 1 文档说明

本文档是《国际卡网关支付接口(alipay.trade.direct.forcard.pay)》附录文档，它详细解释了在技术接入与使用过程中需要注意的地方，以帮助商户避免风险产生。

阅读后如有疑问，请联系支付宝相关技术支持。

## 1.1 前言

- 面向读者

本文档主要面向需要接入国际卡网关支付、内嵌支付的商户的开发人员。

- 读者所需技能

读者需有 php、java、C#程序开发背景。

- 开发环境要求

OS: Windows XP 或以上。

## 1.2 服务介绍-国际卡支付

**步骤1:** 用户选择相应的银行卡种，目前支付宝国际卡支付支持 VISA、MasterCard、JCB（相对应 default\_bank 的银行简码为：boc-visa、boc-master、boc-jcb）。



**步骤2:** 用户提交订单信息，商户调用支付宝国际卡支付接口 (alipay.trade.direct.forcard.pay)，跳转到支付宝收银台。

**步骤3:** 商户处理支付宝系统返回的通知

支付宝系统的返回机制分为：同步通知（return\_url）和异步通知（notify\_url）两种，商户通过获取支付宝返回的通知，对发起的订单进行处理，更新订单的状态，来完成网站的交易。

## 1.3 业务术语

表1-1 业务术语

术语	解释
CTU	支付宝风险稽查系统。

术语	解释
定向支付	用户事先指定支付金额的收款方，对应的交易只能将金额转入指定的收款方账户中。
返回	支付宝根据得到的数据处理完成后，支付宝将处理完成的结果信息反馈给商户网站。
防钓鱼	“网络钓鱼”攻击利用欺骗性的电子邮件和伪造的 Web 站点来进行诈骗活动，受骗者往往会泄露自己的财务数据，如信用卡号、账户用户名、口令和社保编号等内容，造成损失。防钓鱼用来防止以上情况的发生。
分润	分润是指将交易金额中的一部分转账给其它账户。
敏感词	带有敏感政治倾向、暴力倾向、不健康色彩或不文明的词。
请求	通过 HTTP 协议把需要传输的数据发送给接收方的过程。
特殊字符	用做 url 转义字符，或在接口参数中用作分割符的特殊字符，包括：#、%、&、+、^、 。

## 2 责任归属

文档中所涉及到的规则都是根据在接入与使用支付宝接口的过程中出现的一些主要风险而做的防范措施，请商户予以关注。请在接入及使用支付宝接口的过程中，严格依照支付宝提供的接口技术文档（国际卡网关支付接口（alipay.trade.direct.forcard.pay.pdf）、代码示例、本文档（国际卡网关支付接口（alipay.trade.direct.forcard.pay）接入与使用规则）等接口资料，否则由此导致的风险以及资金损失或者扩大情形需商户自行承担。

## 3 接入流程

### 3.1 接入前期准备

接入前期准备工作包括商户签约和权限审核通过（已完成商户可略过）。

### 3.2 获取支付宝分配的合作商户ID和安全校验码

首先，商户在签约成功后可获取支付宝分配的合作商户 ID（PartnerID）和账户 ID（SellerID），步骤如下：

- (1) 用您的签约支付宝账号登录支付宝网站（[www.alipay.com](http://www.alipay.com)）；
- (2) 点击“商家服务”（<https://b.alipay.com/order/myOrder.htm>）；
- (3) 点击“查询合作者身份（PID）”、“查询安全校验码（Key）”。

如下图所示：



图3-1 商户 ID、Key 获取示意图

签约过程中需要任何帮助请联系相应的业务经理或致电：**0571-88158090**（支付宝商户签约服务专线）。

### 3.3 开发

首先：请把浏览器默认语言改成英文(收银台语言根据浏览器默认语言自动切换)。



图3-2 修改服务器默认语言

#### 步骤1： 配置账号、PID、Key

- (1) 解压获取开发代码压缩包，找到文件夹“alipay.trade.direct.forcard.pay”放入网站项目中，并记录该文件夹存放的路径，准备集成外卡接口。
- (2) 配置“alipay.config.php”文件，配置好相应的 partner 与 Key，注意不要有空格。
- (3) 账号相关配置：

签约账号的PID和Key查询参见“3.2 获取支付宝分配的合作商户ID和安全校验码”。

如果签约账号未审核通过，可用 alipay-test17@alipay.com 进行线上测试：具体的PID 和 Key 咨询相关的技术支持。

线上测试交易请联系技术支持做退款操作，完成后请更换成商家签约账号进行收款。

#### 步骤2： 设置国际卡支付接口的请求部分

- (1) 由于支付宝的国际卡支付支持卡种前置，所以需要在订单提交页面中添加卡种的 LOGO，分别是：



- (2) 打开请求参数文件“alipayapi.php”，设置 notify\_url 和 return\_url 的路径（即服务器异步通知返回地址和页面跳转同步返回地址）。在

“alipay.trade.direct.forcard.pay”文件夹根目录下可以找到这 2 个文件，此文件在网站项目中的路径就是需要设置的路径。



注意：

return\_url 和 notify\_url 的路径必须是完整的 url 地址。在本地测试时 return\_url 建议使用 127.0.0.1 这个 IP 地址或绑定域名来设置路径。notify\_url 必须是能够在互联网上被访问到的地址。

- (3) 在“alipayapi.php”请求文件中分别设置商户订单号、商品名称、总金额等参数。

### 步骤3： 编写国际卡支付接口的通知返回文件 return\_url 和 notify\_url

根据《国际卡网关支付接口(alipay.trade.direct.forcard.pay)》文档中的“5 页面跳转同步通知参数说明”，需要获取通知返回的信息，例如商户订单号、总金额、币种、交易状态、支付宝交易号等，根据不同的交易状态编写不同的业务逻辑代码，实现商户网站中的订单信息的同步（即根据支付宝的交易状态来更新商户网站中该笔订单的交易状态），如未付款变成已付款。



注意：

页面跳转同步通知文件 return\_url 和服务器异步通知文件 notify\_url 都必须判断在商户网站中是否已经对该笔交易的通知结果数据做过同样的处理，以防止数据库逻辑处理代码被重复执行 2 次。这 2 个文件的区别，可以参考《国际卡网关支付接口(alipay.trade.direct.forcard.pay)》中的“7 通知返回数据处理”。

### 下面着重讲解：notify\_url

请查看：买家手动关闭了返回页面，交易订单如何更新？

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=243128&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=243128&sh=Y&tab=null&info_type=9)

交易完成后，支付宝主动发起通知调用 notify\_url，将交易信息发送给商户服务器。这是服务器间的交互，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。商户通过支付宝的通知判断交易是否成功，具体如下：

- **notify\_url 地址：**支付宝将以 **POST** 方式调用该地址，因此该页面中获取参数的方式为 post。

- **通知触发条件：**只有在支付宝的交易管理中存在该笔交易，且发生了交易状态的改变，支付宝才会通过该方式发起服务器通知，如从“等待买家付款”到“支付成功”。
- **商户返回信息：**商户收到服务器异步通知并仅打印出 **success**，不能包含其他任何 HTML 语言的标签。
- **通知重发：**若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）。
- **交易判断条件：**当收到 **trade\_status=TRADE\_SUCCESS** 或 **TRADE\_FINISHED** 的通知后才可判定交易成功（其它 **trade\_status** 状态请求可不作处理）。

#### 步骤4： 开启日志文件

打开 lib 类文件夹中的支付宝通知处理类文件“alipay\_notify.class.php”，开启“alipay\_notify.class.php”里的写日志函数，即去掉注释。

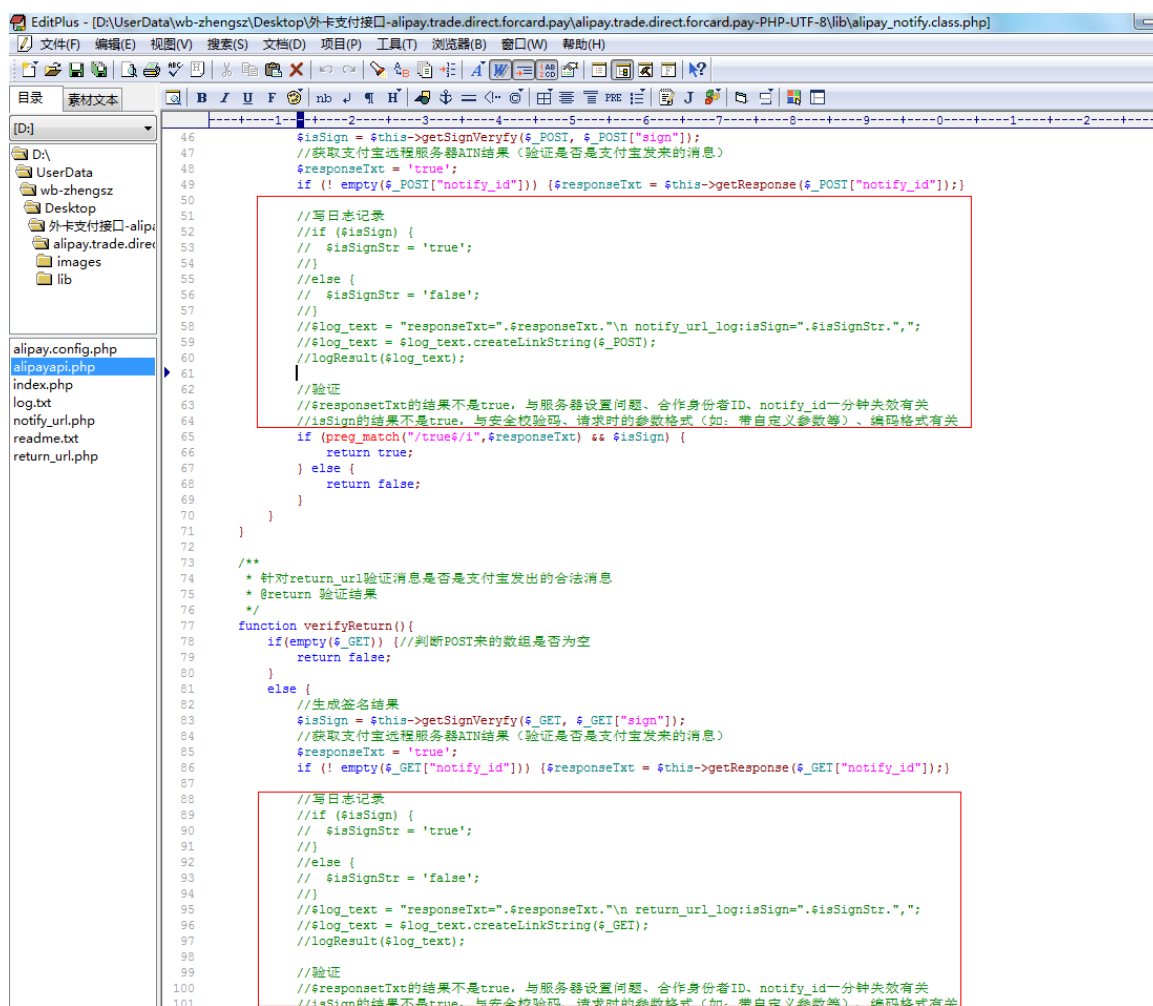


图3-3 开启写日志函数



测试一笔交易后，log.txt 里会有记录：支付宝返回的签名结果 sign 等于服务器异步通知文件生成的签名结果 mysign；是否是支付宝发来验证 responseText = 'true'；以及支付宝发送回来的所有参数信息。

通过日志功能可以检查服务器异步通知文件的执行情况，测试到这一步的情况，再到商户网站的订单系统查看该笔订单状态是否更新，如果已变更，则国际卡支付接口已集成完毕。

## 4 测试流程

### 4.1 测试部署

提交发送给支付宝的请求url请求网关是<https://mapi.alipay.com/gateway.do?>。

发送给支付宝的请求，建议使用 form 表单传输，以防止 get 传输数据过长而发生数据丢失的情况，参照如下规范：

- action的值必须为：  
“ [http://mapi.alipay.net/gateway.do?\\_input\\_charset= 该 值](http://mapi.alipay.net/gateway.do?_input_charset=utf-8) ”，  
如：[http://mapi.alipay.net/gateway.do?\\_input\\_charset=utf-8](http://mapi.alipay.net/gateway.do?_input_charset=utf-8)；  
不允许写成完整的请求链接地址，即禁止<http://mapi.alipay.net/gateway.do?>后带有所有要请求给支付宝的请求参数数据。
- <form>与</form>之间需包含所有要请求给支付宝的参数，且每个参数的格式为<input type="hidden" name="参数名" value="参数值" />。
- 在众多请求参数中，请求参数\_input\_charset（编码格式）必须存在于 form 表单中，即 form 表单中必须含有 <input type="hidden" name="\_input\_charset" value="参数值">。
- <form>与</form>之间包含的数据只允许是要请求给支付宝的参数，禁止出现商户自行命名，不在接口技术文档请求参数列表中的其他数据。
- form 表单的 method 属性，可自行选择 get、post 两种。

### 4.2 页面跳转案例

**步骤1：** 生成的跳转 URL。

```
https://mapi.alipay.com/gateway.do?extend_param=SHIP_TO_FIRSTNAME%5ENicholas+%7CSHIP_TO_LASTNAME%5ECapotosto+%7CSHIP_TO_POSTALCODE%5E60047+%7CSHIP_TO_PHONENUMBER%5E8478779280+%7CSHIP_TO_STREET1%5E1295+Charleston+%7CSHIP
```

```
_TO_STREET2%5E1295+Charleston+%7CSHIP_TO_CITY%5Ehawthorn+%7CSHIP_TO_STATE%5EIL+%7CSHIP_TO_COUNTRY%5EUS+%7CSHIP_TO_SHIPMETHOD%5Eyunda+%7CLOGISTICS_COST%5E0.00+%7CJS_RETURN%5E109sJL%2BkFkT547hj62Xtd2p3a2XuJemlriJvZOhk72MuJf6k%2BvFrdm0lrqY6o%2FpjP6M6Zu5g6GDr434i%2B6c3brfscXn3f%2By3afOos6vgLWbq4uj9J3zl%2FiP%2FNySxubQ%2Fs%2Fmxof3h%2Bu02bzelfyIp5KhlriLvZ21%2FrBir%2BPP74PqgeTEg%2BaF7oGoiMuj0b7Ttpmrk72No5Knl6eJsIWl9pfxkOKLpJGIlbuIvpywkuaJ4ofpy%2FHT4tHm0OnR59fj0efR5prjgemLzP7I%2F8%2BdsoDvnKvopuOniOPbjd2V2fKa0bDj0vCN&subject=%E5%A4%96%E5%8D%A1%E6%94%B6%E5%8D%95&default_bank=cybs-visa&sign_type=MD5&notify_url=http%3A%2F%2Fapi.test.alipay.net%2Fatinterface%2Freceive_notify.htm&out_trade_no=5378832513215254&return_url=http%3A%2F%2Fapi.test.alipay.net%2Fatinterface%2Freceive_return.htm&currency=USD&sign=12ed698aa8ef1026d0eb0d20a0c6ec26&_input_charset=UTF-8&total_fee=1&service=alipay.trade.direct.forcard.pay&paymethod=jvm-3d&partner=2088201564895543&seller_logon_id=alipay-test17%40alipay.com
```

解析（具体要提交的参数）：

```
1:https://mapi.alipay.com/gateway.do?extend_param=SHIP_TO_FIRSTNAME^Nicholas|SHIP_TO_LASTNAME^Capotosto|SHIP_TO_POSTALCODE^60047|SHIP_TO_PHONENUMBER^8478779280|SHIP_TO_STREET1^1295Charleston|SHIP_TO_STREET2^1295Charleston|SHIP_TO_CITY^hawthorn|SHIP_TO_STATE^IL|SHIP_TO_COUNTRY^US|SHIP_TO_SHIPMETHOD^yunda|LOGISTICS_COST^0.00|JS_RETURN^109sJL+kfKT547hj62Xtd2p3a2XuJemlriJvZOhk72MuJf6k+vFrdm0lrqY6o/pjP6M6Zu5g6GDr434i+6c3brfscXn3f+y3afOos6vgLWbq4uj9J3zl/iP/NySxubQ/s/mxof3h+u02bzelfyIp5KhlriLvZ21/rBir+PP74PqgeTEg+aF7oGoiMuj0b7Ttpmrk72No5Knl6eJsIWl9pfxkOKLpJGIlbuIvpywkuaJ4ofpy/HT4tHm0OnR59fj0efR5prjgemLzP7I/8+dsoDvnKvopuOniOPbjd2V2fKa0bDj0vCN
2:subject=外卡收单
3:default_bank=cybs-visa
4:sign_type=MD5
5:notify_url=http://api.test.alipay.net/atinterface/receive_notify.htm
6:out_trade_no=5378832513215254
7:return_url=http://api.test.alipay.net/atinterface/receive_return.htm
8:currency=USD
9:sign=12ed698aa8ef1026d0eb0d20a0c6ec26
10:_input_charset=UTF-8
11:total_fee=1
12:service=alipay.trade.direct.forcard.pay
13:paymethod=jvm-3d
14:partner=2088201564895543
15:seller_logon_id=alipay-test17@alipay.com
```

其中 **extend\_param** 标准样例：

```
extend_param=SHIP_TO_FIRSTNAME^Nicholas|SHIP_TO_LASTNAME^Capotosto|SHIP_TO_POSTALCODE^60047|SHIP_TO_PHONENUMBER^8478779280
```

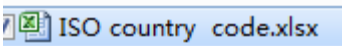
```
|SHIP_TO_STREET1^1295      Charleston      |SHIP_TO_STREET2^1295      Charleston
|SHIP_TO_CITY^hawthorn      |SHIP_TO_STATE^IL          |SHIP_TO_COUNTRY^US
|SHIP_TO_SHIPMETHOD^yunda    |LOGISTICS_COST^0.00
|JS_RETURN^109sJL+kfKT547hj62Xtd2p3a2XuJemlriJvZOhk72MuJf6k+vFrmd0lrqY6o/
pjP6M6Zu5g6GDr434i+6c3brfscXn3f+y3afOos6vgLWbq4uj9J3zl/iP/NySxubQ/s/mxof3
h+u02bzelfyIp5KhlriLvZ2l/rbir+PP74PqgeTEg+aF7oGoiMuj0b7Ttpmrk72No5Kn16eJs
IWl9pfxkOKLpJGIlbuIvpywkuaJ4ofpy/HT4tHm0OnR59fj0efR5prjgemLzP7I/8+dsoDvnK
vopuOniOPbjd2V2fKa0bDj0vCN
```



注意：

- 10 个 SHIP 均不可空（SHIP\_TO\_STREET2 可以省略，其他均不可空）；
- 3 个 REGISTRATION 均不可空；
- Js\_return 获取方式：先在页面引入 js 的脚本：`<script type="text/javascript" src="https://{rds domain}/merchant/merchant.js"></script>`  
rds domain 可以选择：rds.alipay.com 或 irds.alipay.com；  
引入 js 脚本后通过 `var js_return = window["alipay-merchant-result"]` 获取；
- php 获取 js\_return 赋值 extend\_param 样例：  
`$("#alipay_expend_param").val('js_return' + window["alipay-merchant-result"]);`

extend\_param 中的参数值规范如下：

- SHIP\_TO\_COUNTRY 和 SHIP\_TO\_STATE，如果是美国（US）和加拿大（CA）要用标准的二位码，见 ，国家编码在第一个 sheet 中，州编码在第二和第三个 sheet 中，只有美国和加拿大需要单选框选择，其他国家由输入框输入。  
也就是美国、加拿大的国家和州必须存在（数据库必须有这 2 个字段），而且按标准二位码传。
- 参数都需要有值，要求姓名、电话、收货地址在商户网站的订单系统里是必填项，SHIP\_TO\_SHIPMETHOD 可传 7，其他参数如果商户自身网站系统无法更改为必填项，请技术人员不要传输此字段。
- 如果商户面向的市场是小语种国家，建议对需要买家填写的相关字段进行 encode 编码，以防止参数提交到网关时因乱码导致报 ILLEGAL\_SIGN 签名错误，后续如果需要对这些返回的参数做处理，可进行 decode 解码。



注意：

- 由于国际卡收单交易不支持修改金额数值的特性，所以如果商户存在多个站点，建议区分各分站点的订单号，以防交易号冲突而报 w\_033 错误；
- 判断订单返回结果需根据字段 trade\_status 值: trade\_success、trade\_closed、trade\_finished 这 3 个状态，建议将通知返回的逻辑放在异步通知（notify\_url）里处理，以防止买家付款成功之后直接关掉浏览器而无法触发同步通知。

**步骤2：** 用 Visa\Master 测试支付、查看通知返回。

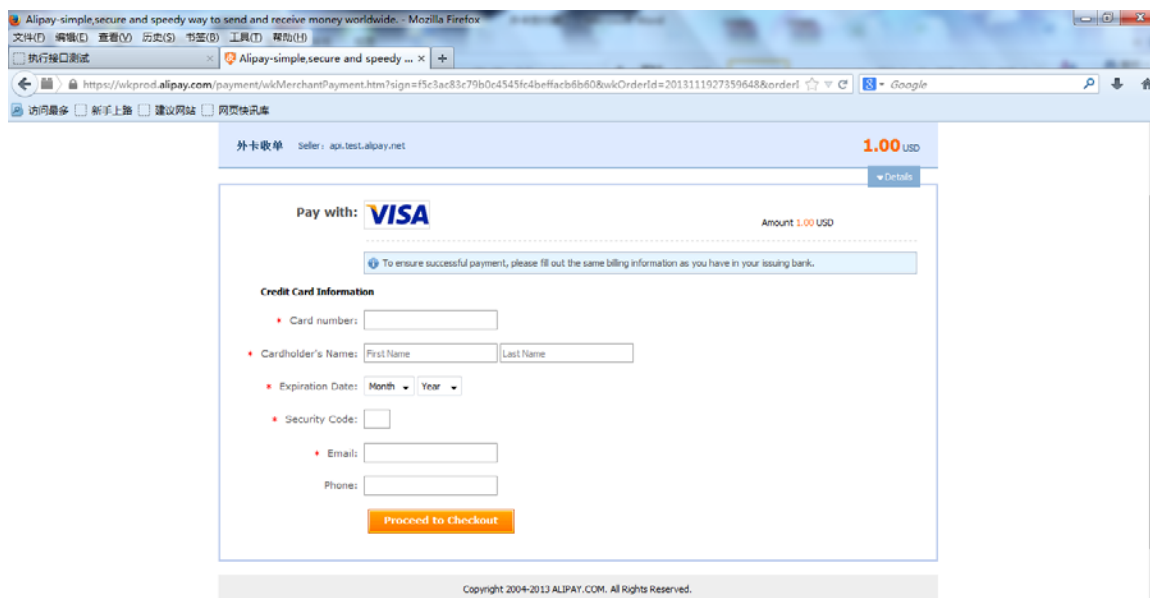


图4-1 填写 Credit Card 示意图（页面跳转）

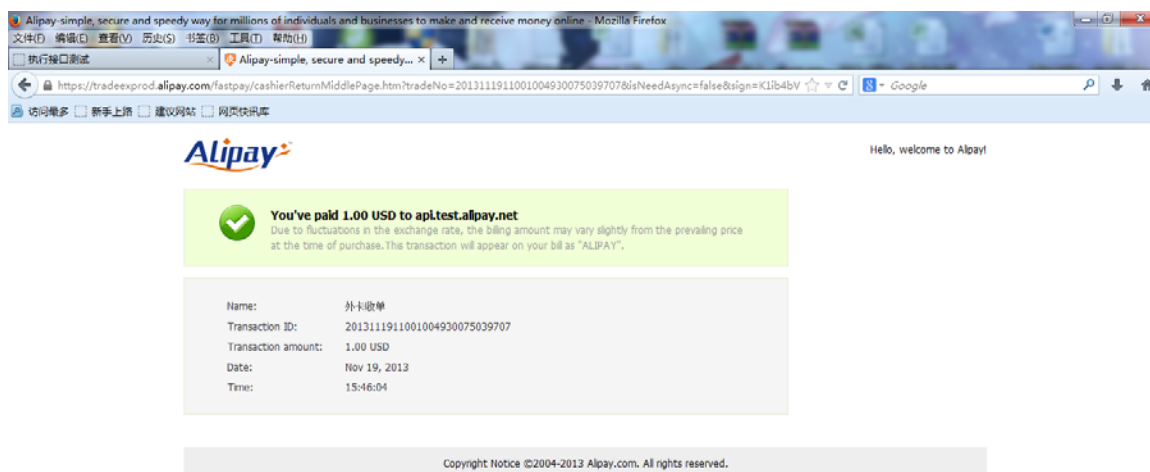


图4-2 支付成功示意图

**步骤3：** 获取 return\_url。

return\_url 样例:

```
http://api.test.alipay.net/atinterface/receive_return.htm?bank_seq_no=2013111901152379300&currency=USD&extface=alipay.trade.direct.forcard.pay&forex_total_fee=1.00&is_success=T&notify_id=RqPnCoPT3K9%252Fvwbh3I75K66XB6XjZXCzAUBkZ9f2s7gloy3xCuQN%252F51AbTVaNxRsdr1V&notify_time=2013-11-19+15%3A49%3A05&notify_type=trade_status_sync&out_trade_no=6694528631410923&payment_type=600&seller_email=alipay-test17%40alipay.com&subject=%E5%A4%96%E5%8D%A1%E6%94%B6%E5%8D%95&total_fee=6.02&trade_no=2013111911001004930075039707&trade_status=TRADE_SUCCESS&sign=7c259755fe65d4b655f5b396e6151493&sign_type=MD5
```

解析（具体返回的参数）：

```
1:http://api.test.alipay.net/atinterface/receive_return.htm?bank_seq_no=2013111901152379300
2:currency=USD
3:extface=alipay.trade.direct.forcard.pay
4:forex_total_fee=1.00
5:is_success=T
6:notify_id=RqPnCoPT3K9%2Fvwbh3I75K66XB6XjZXCzAUBkZ9f2s7gloy3xCuQN%2F51AbTVaNxRsdr1V
7:notify_time=2013-11-19 15:49:05
8:notify_type=trade_status_sync
9:out_trade_no=6694528631410923
10:payment_type=600
11:seller_email=alipay-test17@alipay.com
12:subject=外卡收单
13:total_fee=6.02
14:trade_no=2013111911001004930075039707
15:trade_status=TRADE_SUCCESS
16:sign=7c259755fe65d4b655f5b396e6151493
17:sign_type=MD5
```

## 5 技术接入规则

表5-1 技术接入规则

类型	细则	原因
账号	配置的合作者身份 ID 与安全校验码 Key 必须保证与签约信息匹配	防止接口无法正常使用或出现资金损失

类型	细则	原因
	必须保护合作者身份 ID 与安全校验码 Key 的隐私性	防止签约的账号信息被盗用，导致资金受损、被他人恶意利用等。
	测试完毕后，要把测试账号立刻更换成签约账号。	使用测试账号时，手续费按照 3% 扣除。
安全	该接口必须使用 https 请求	避免请求参数暴露
	支付宝的通知 IP 是 121.0.26.1 与 121.0.26.2。该 IP 地址不是商户访问支付宝的地址，而是支付宝发送通知给商户的出口地址。	如果商户网站设置了 IP 白名单（即 IP 过滤），需要把支付宝的通知 IP 地址加入白名单中。
	商户必须以 DNS 解析的方式访问支付宝接口，不要设置 DNS cache，不要绑定支付宝 IP。如果为了商户自身安全必须绑定支付宝 IP 时，必须向支付宝的技术支持人员备案。	支付宝 IP 地址一旦变更，会导致商户无法请求或访问支付宝，致使商户业务直接不可用。
签名	请求的所有参数，需要根据参数名=参数值的格式，按首字符字典顺序（ascii 值大小）排序，若遇到相同首字符，则判断第二个字符，以此类推，待签名字符串需要以“参数名 1=参数值 1&参数名 2=参数值 2&...&参数名 N=参数值 N”的规则进行拼接。	避免接口无法正常使用
	在对请求的参数做签名时，这些参数必须来源于请求参数列表，并且除去列表中的参数 sign、sign_type。	避免接口无法正常使用
	在对请求的参数做签名时，对于请求参数列表中那些可空的参数，如果选择使用它们，那么这些参数的参数值必须不能为空或空值。	避免接口无法正常使用
	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。	避免接口无法正常使用
	待签名数据应该是参数原始值而不是 url encoding 之后的值，例如：调用某接口需要对请求参数 email 进行数字签名，那么待签名数据应该是 email=test@msn.com，而不是 email=test%40msn.com。	避免接口无法正常使用
参数配置	在请求参数列表中，不可空的参数必须配置。	避免接口无法正常使用
	在请求参数列表中，可空的但需要多选一的多个参数中，必须配置至少一个。	避免接口无法正常使用
	必须按照请求参数列表中各参数的格式要求配置	避免接口无法正常使用

类型	细则	原因
	必须设置请求参数 <code>_input_charset</code> （编码格式），即该参数不能为空，并让该参数加入签名运算。	避免接口无法正常使用
	只有开通了自定义超时功能，才能使用请求参数 <code>it_b_pay</code> （超时时间）。	避免接口报错误码 <code>SELF_TIMEOUT_NOT_SUPPORT</code>
	只有开通了防钓鱼功能且开通了防钓鱼时间戳，才能使用请求参数 <code>anti_phishing_key</code> （防钓鱼时间戳）。	避免接口无法正常使用
	只有开通了防钓鱼功能且开通了 IP 地址检查，才能使用请求参数 <code>exter_invoke_ip</code> （客户端 IP）。	避免接口无法正常使用
	请求参数 <code>subject</code> 、 <code>body</code> 的值，以及 <code>extend_param</code> 、 <code>royalty_parameters</code> 的备注表述信息中不要使用敏感词。	避免接口无法正常使用
	请求参数 <code>notify_url</code> 的设置必须是互联网上能访问到且访问正常的路径地址	避免商户网站无法收到支付宝的主动通知
	请求参数 <code>return_url</code> 的设置不能是 <a href="http://localhost/">http://localhost/</a> 这类地址，必须是服务器 IP 地址或者域名方式。（例如： <code>127.0.0.1</code> ）	避免付款成功后，当前页面停留在支付宝交易完成提醒页面，而不做任何跳转
	请求参数 <code>return_url</code> 的设置不能是含有“!” 这类特殊字符的地址	避免付款成功后，当前页面停留在支付宝交易完成提醒页面，而不做任何跳转
	<code>royalty_parameters</code> （分润账号集）、 <code>extend_param</code> （公用业务扩展参数）参数中的备注描述信息中不能出现用作字段分割符的“^”、“ ” 特殊字符。	避免出现数据格式错误，导致分润失败。
	当使用了分润功能时，在设置请求参数 <code>royalty_parameters</code> （分润账号集）的值时，分润的总金额不能超过付款总金额减去支付宝手续费所余下的金额。	避免分润失败，而导致接口无法正常使用。
	请求参数 <code>return_url</code> 必须设置，必须是域名方式。（例如： <code>http://www.***.com/return_url.php</code> ）	收银台显示同一个 PID 的不同商户网址
	当使用了分润功能时，在设置请求参数 <code>royalty_parameters</code> （分润账号集）的值时，分润的收款账户必须是有效的收款账号。	避免分润失败，而导致接口无法正常使用。
	<code>seller_id</code> （卖家支付宝账户号）、 <code>seller_logon_id</code> （卖家支付宝账号）不能全部为空，至少有一项不为空。在都不为空的情况下，优先级顺序为： <code>seller_id &gt; seller_logon_id</code>	数据完整一致，避免出现卖家信息错误。



类型	细则	原因
	如果设置了买家支付宝账号（如 <code>buyer_email</code> 等），那么买家支付宝账号不能与卖家支付宝账号相同，即： <code>buyer_email</code> 与 <code>seller_logon_id</code> 不能相同、 <code>buyer_id</code> 与 <code>seller_id</code> 不能相同。	避免报错，如错误码： <code>BUYER_SELLER_EQUAL</code> 。
	<code>price</code> （商品单价）、 <code>quantity</code> （购买数量）会替换 <code>total_fee</code> （交易金额）。即 <code>total_fee</code> 不能与 <code>price</code> 、 <code>quantity</code> 同时存在；存在 <code>price</code> 、 <code>quantity</code> ，就不能存在 <code>total_fee</code> 。	防止出现支付金额错误
	在给请求参数 <code>default_bank</code> 赋值时，需注意中间“-”，即 <code>boc-visa</code> 、 <code>boc-master</code> 、 <code>boc-jcb</code> 。	否则会引起银行调用失败
接口构造	必须使用支付宝的网关发送请求信息给支付宝，请求网关： <a href="https://mapi.alipay.com/gateway.do">https://mapi.alipay.com/gateway.do</a> 。	避免被钓鱼网站利用
	发送给支付宝的请求，请求参数不仅包含参与签名的参数，还包含参数 <code>sign</code> 、 <code>sign_type</code> 。	避免接口无法正常使用
	如果集成 <code>iframe</code> ，则加上请求参数 <code>error_return_url</code> ，并按 <code>iframe</code> 集成文档，更改补充 <code>extend_param</code> 。	方便商户定位接口报错及实现站内支付
	<p>发送给支付宝的请求，如果使用 <code>form</code> 表单传输，需要按照以下要求编写：</p> <ul style="list-style-type: none"> <li><code>action</code> 的值必须为 “<a href="https://mapi.alipay.com/gateway.do?_input_charset=utf-8">https://mapi.alipay.com/gateway.do?_input_charset=utf-8</a>”， 如：<a href="https://mapi.alipay.com/gateway.do?_input_charset=utf-8">https://mapi.alipay.com/gateway.do?_input_charset=utf-8</a>。</li> <li>不允许写成完整的请求链接地址，即禁止 <a href="https://mapi.alipay.com/gateway.do">https://mapi.alipay.com/gateway.do</a> 后带有所有要请求给支付宝的请求参数数据；</li> <li><code>&lt;form&gt;</code> 与 <code>&lt;/form&gt;</code> 之间需包含所有要请求给支付宝的参数，且每个参数的格式为 <code>&lt;input type="hidden" name="参数名" value="参数值" /&gt;</code>；</li> <li>在众多请求参数中，请求参数 <code>_input_charset</code>（编码格式）必须存在于 <code>form</code> 表单中，即 <code>form</code> 表单中必须含有 <code>&lt;input type="hidden" name="_input_charset" value="参数值"&gt;</code>；</li> <li><code>&lt;form&gt;</code> 与 <code>&lt;/form&gt;</code> 之间包含的数据只允许是要请求给支付宝的参数，禁止出现商户自行命名，不在接口技术文档请求参数列表中的其他数据；</li> <li><code>form</code> 表单的 <code>method</code> 属性，可自行选择 <code>get</code>、<code>post</code> 两种。</li> </ul>	<ul style="list-style-type: none"> <li>避免请求支付宝时报错，错误码为 <code>ILLEGAL_SIGN</code>；</li> <li>在 win7 系统下，如果浏览器是 IE8 以上，有可能出现发送请求链接时会无法跳转到支付宝，当前页面为空白页的情况。</li> </ul>




类型	细则	原因
数据传输	必须使用 https 协议，支持 get、post 或者 httpclient 方式提交。	避免接口无法正常使用
	设置了请求参数 extend_param 的情况下，建议使用 post 方式提交请求。	避免地址栏中地址数据过长，导致传递的数据丢失。
通知返回验证	如果有设置通知路径及触发条件，则必须使用获取到的参数 notify_id 再次请求支付宝，获取是否是支付宝发送的验证结果。该请求链接是： <a href="https://mapi.alipay.com/gateway.do?partner=合作者身份ID&amp;notify_id=通知ID的值">https://mapi.alipay.com/gateway.do?partner=合作者身份ID&amp;notify_id=通知ID的值</a>	验证是否是支付宝发来的请求
	在对通知的参数做签名时，这些参数必须来源于支付宝通知回来的参数，并且除去列表中的参数 sign、sign_type，根据参数名=参数值的格式，按首字符字典顺序（ascii 值大小）排序，若遇到相同首字符，则判断第二个字符，以此类推，待签名字符串需要以“参数名 1=参数值 1&参数名 2=参数值 2&...&参数名 N=参数值 N”的规则进行拼接，得到的签名结果与获取到的参数 sign 值做比较。	验证返回的签名
返回数据处理	支付宝主动发送通知，当商户接收到通知数据后必须给支付宝返回“success”字符串，不允许返回其他多余字符。	如果商户返回给支付宝的信息不是“success”，支付宝最多重复发送 8 次通知。 <b>说明：</b> 一旦商户收到异步通知返回了纯字符串 success 给支付宝，支付宝就不会再发送异步通知，否则会继续按照发送时间发送通知。
	必须保证设置的通知路径互联网上能访问得到，且访问顺畅。	避免接收不到支付宝发送的通知
	必须对返回的所有结果数据进行处理	以便商户能够了解接口的使用情况，以及进行商户的后续业务操作。
	必须判断发送支付请求以后的业务逻辑处理程序是否有重复执行	防止出现商户的业务操作被重复执行，导致出现资金损失，如重复充值、重复付款等。
	如果交易付款完成时发送的交易状态是 TRADE_SUCCESS（可对交易做其他操作，如退款、分润等），则当超过签约合同指定的可退款时间段时，支付宝会主动发送 TRADE_FINISHED（不能对该交易再做任何操作）交易状态。此时，需要根据商户自身业务情况，来判断是否需对这次的交易完成通知进一步处理。	防止出现商户的业务操作被重复执行，导致出现资金损失，如重复充值、重复付款或订单数据错乱等。


类型	细则	原因
	建议每一次支付操作需以日志形式记录到商户网站的日志操作数据库中	用来在必要时检查或跟踪业务处理情况
错误码处理	遇到特殊错误码（如：UNKNOWN_EXCEPTION、GENERIC_FAILURE、SYSTEM_ERROR），须暂停后续一切操作，或者联系支付宝技术支持协助处理。	一般这种情况有可能请求已经提交到支付宝，支付宝系统失去响应但有可能由恢复机制自动回复并提交请求。
	如果遇到错误码“TRADE_BUYER_NOT_MATC”，表明支付使用的账户与原来的不一致，请使用原来的账户，或重新创建交易付款。	出现这种情况是买家账户不一致，或者买家使用多个账户，支付时账户必须与购买时账户（创建交易账户）一致。
	如果遇到错误码“ILLEGAL_FEE_PARAM”，表明交易金额错误，请检查交易金额是否正确。	遵循如下规则： <ul style="list-style-type: none"> <li>总价和单价不可以同时出现；</li> <li>如果存在总价，则应该输入商品数量，如果未输入商品数量，系统不报错，默认为 1；</li> <li>如果存在单价，则商品数量必须存在。</li> </ul>
	如果遇到错误码“FOREIGN_TRADE_UPDATE_ADJUST_FEE_ERROR”、“FOREIGN_TRADE_TOTALFEE_NOT_MATCH”，表明商品价格或总价可能发生了变化，请重新创建交易付款。	出现这种情况，可能是商品价格发生了变更。（例如：买家走到收银台后联系卖家修改了价格再支付；隔天因为汇率波动导致支付金额发生变动。）
	如果遇到错误码“FAIL_CREATE_CASHIER_PAY_ORDER”，可能是支付宝收银台系统出现异常，请稍候重试或联系支付宝技术支持协助处理。	一般是支付宝收银台系统出现异常
	错误码“INTERNATIONAL_CARD_EXTENDPARAMETER_PARSE_ERROR”、“This transaction failed because of system error. Please contact your merchant for more information. (NO. w_010)”表示外卡收单商户传入的扩展信息解析失败，即 extend_param 传值错误。	extend_param 的参数值必须有值，且 country、state 对于 US、CA 必须使用 ISO 标准，且是二位码。参照“ISO country code.xlsx”。
	错误码“wk.0004、wk.0005、wk.0006、wk.0007、wk.0012”表示网络系统忙，请重新发起支付。	可能原因： <ul style="list-style-type: none"> <li>网络不稳定；</li> <li>浏览器需要先清缓存。</li> </ul>

类型	细则	原因
	错误码 “This transaction failed because we were unable to verify this credit card through our card validation process. Please return to merchant's website and try a different payment method. (NO.wk_0008)” ，属于 CTU 校验失败。	可能原因： <ul style="list-style-type: none"> <li>extend_param 参数值有空值、country 和 state 对于 US、CA 未按 ISO 标准；</li> <li>风险交易。</li> </ul>
	错误码 “The amount of your paymrnt exceeds your credit card limit. Please return to merchant's website and try a different payment method. (NO.wk_boc_007)” ，信用卡额度不足。可以更换其他卡付款。	信用卡额度不足，可以更换其他卡付款。
	错误码 “This transaction failed because your issuing bank declined. Please return to merchant's website and try a different payment method. (NO.wk_boc_011)” ，银行拒绝交易。	可以更换其他卡付款
	如果交易状态出现 “TRADE_PENDING” ，则表明收款方账号未激活或收款方账号被锁定，请重新登录支付宝激活账号。	收款方账号未激活或收款方账号被锁定
自主编写接口代码规则	如果不使用支付宝提供的代码示例来集成接口，那么必须根据技术文档中签名机制和通知返回数据处理章节及本文档的技术接入规则、接口使用规则、测试流程规则，来编写符合商户网站项目的接口代码。	避免接口无法正常使用

## 6 接口使用规则

表6-1 接口使用规则

类型	规范点	原因
业务操作	接口支持重复调用，前提是交易基本信息（买家、卖家、交易金额、超时时间等）在多次调用中保持一致，且交易尚未完成支付。	防止重复付款以及交易信息被篡改  说明： <ul style="list-style-type: none"> <li>如果发现买家不一致、卖家不一致或交易金额不一致，系统会报错不能继续支付；</li> <li>如果超时时间不一致，则交易状态为等待买家支付，但买家支付时系统会报错。</li> </ul>

类型	规范点	原因
	填写买家支付宝账号时，不能与卖家的支付宝账号相同。	避免报错，如错误码： BUYER_SELLER_EQUAL。
	外卡的交易不允许卖家在交易创建后修改价格	会影响同一笔交易的支付
	<p>支付宝每一次退款成功，都会向商户的即时到账异步通知地址 <code>notify_url</code> 发送一条交易成功的通知，其中状态有 <code>refund_status</code>、<code>refund_success</code>。</p> <p> 注意：</p> <ul style="list-style-type: none"> <li>● 商户要注意做防订单重复更新的代码处理。</li> <li>● 交易关闭的状态不会发送通知，因此当进行全额退款操作时，会出现收不到通知的情况，而在部分退款操作时，能收到即时到账接口的交易通知。</li> </ul> <p>解决方案是：</p> <p>商户使用退款接口，把退款的商户业务逻辑程序写到退款接口的通知页面里去，而即时到账接口中不对退款的通知做任何判断。</p>	如果不处理或者处理错误会将商户自己的业务逻辑执行 2 次以上
	如果商户的一笔交易在退款期限内，没有进行退款操作，则支付宝系统会默认将交易的状态改为“交易完成”（即不可退款模式），并主动发送一条交易状态为 <code>trade_finished</code> 的异步通知（可以联系支付宝关闭发送），商户需要根据通知，结合自身业务逻辑做交易不可退款的数据库状态变更。	处理错误会导致商户业务逻辑重复 2 次付款成功
平级分润规则	<p>平级分润格式：</p> <p>收款方 <code>Email_1</code>^金额 1^备注 1 收款方 <code>Email_2</code>^金额 2^备注 2</p> <p>功能效果：</p> <p>买家付出了交易金额 100 元，同时，金额 1 给了收款方 <code>Email_1</code>，金额 2 给了收款方 <code>Email_2</code>，<code>seller_logon_id</code> 获得剩下的金额（剩下的金额为：100-金额 1-金额 2-支付宝手续费）。</p> <p>金额计算规则：</p> <ul style="list-style-type: none"> <li>● 买家交易金额 = 金额 1 + 金额 2 + <code>seller_logon_id</code> 收款金额 + 支付宝手续费</li> <li>● 收款方 <code>Email_1</code> 实际获得金额 = 金额 1</li> <li>● 收款方 <code>Email_2</code> 实际获得金额 = 金额 2</li> <li>● <code>seller_logon_id</code> 实际获得金额 = 买家交易金额 - 金额 1 - 金额 2 - 支付宝手续费</li> </ul>	分润规则，避免出现分润错误。

类型	规范点	原因
多级分润规则	<p>多级分润格式： 收款方 Email_1^金额 1^备注 1 收款方 Email_1^收款方 Email_2^金额 2^备注 2</p> <p>功能效果： 买家付出了交易金额 100 元，同时，金额 1 给了收款方 Email_1，收款方 Email_1 把获得的金额中的金额 2 给了收款方 Email_2，seller_logon_id 获得剩下的金额（剩下的金额可以为：100-金额 1-支付宝手续费）。</p> <p>金额计算规则：</p> <ul style="list-style-type: none"> <li>• 买家交易金额=金额 1+seller_logon_id 收款金额+支付宝手续费</li> <li>• 收款方 Email_1 实际获得金额=金额 1-金额 2（金额 1 必须大于等于金额 2）</li> <li>• 收款方 Email_2 实际获得金额=金额 2</li> <li>• seller_logon_id 实际获得金额=买家交易金额-金额 1-支付宝手续费</li> </ul>	分润规则，避免出现分润错误。
业务应用注意事项	<p>一般情况下，异步通知比同步通知慢 3 秒左右。但在网络环境、服务器处理速度等因素影响下，有时异步通知会比同步通知快一些，所以商户需要使用数据库数据锁防止订单重复更新。</p>	外卡异步通知与同步通知时间
	<p>建议商户不要使用类似 js 的 window.open、location.replace 等没有访问记录的重定向函数跳转到支付网关，强烈建议使用 form 表单提交数据。</p>	规范代码使用规则及减少弹出控件拦截支付（导致支付成功率下降）
	<p>在集成接口的过程中，如果商户使用 seller_logon_id 作为收款账户，则当需要修改该支付宝账户名称时，商户必须在程序中做相应修改，否则在支付过程中会出现卖家账户不存在的错误。推荐使用 seller_id 做为商户收款账户。</p>	seller_id 值是支付宝分配给商户的唯一身份 ID，不会改变，使用 seller_id 作为收款账户可以避免 seller_logon_id 变更导致商户无法收款的问题。（可在集成时联系支付宝技术支持获取 seller_id 值。）
	<p>卖家发起退款操作后，买家是通过什么渠道付款的，则款项原路退回。比如，买家用的是支付宝账号余额付款，则钱退回到支付宝账号；信用卡付款，则退回信用卡。</p>	买家通过何种渠道付款，则款项原路退回。