

## 快捷登录集成教程

---

**服务名称：alipay.auth.authorize**

**create\_partner\_trade\_by\_buyer**

**create\_direct\_pay\_by\_user**

**trade\_create\_by\_buyer**

版本号：1.0.1

需要做的

● 集成

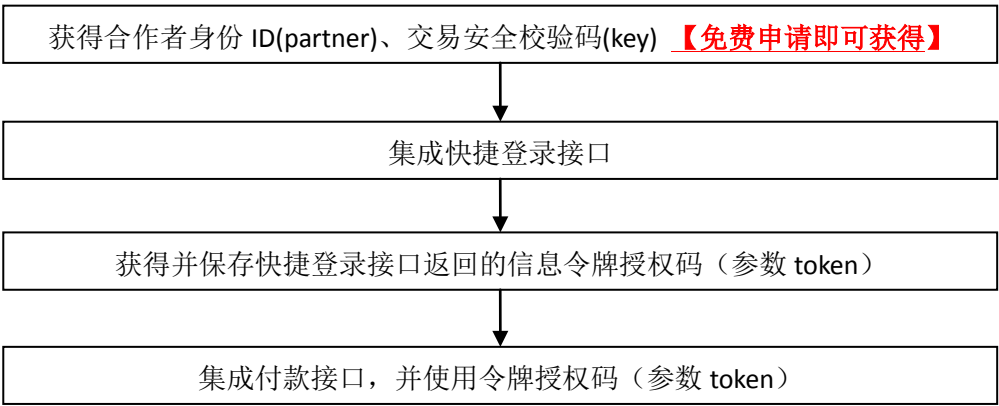


图 集成流程

- 1、合作者身份ID（partner）、交易安全校验码（key）需要登录[www.alipay.com](http://www.alipay.com)，点击“商家服务”栏目，在该界面中获得。此查询操作需要签约的支付宝账号、登录密码、支付密码。
- 2、配置快捷登录接口、付款接口（如即时到帐接口、担保交易接口、标准双接口等）。需要配置的文件有（详细可见[6 集成](#)）：

类别	规范点
所有接口	alipay_config 文件
付款接口	alipayto 文件
快捷登录接口	alipay_auth_authorize 文件
存在 return_url 的所有接口	return_url 文件
存在 notify_url 的所有接口	notify_url 文件

表 需要配置的文件列表

- 3、令牌授权码（参数 token）是通过快捷登录接口返回获得的（详细可参见快捷登录接口技术文档 5.2 章节）
- 4、把获得到的令牌授权码（参数token）也同其他的请求参数一起，参与付款接口（如即时到帐接口、担保交易接口、标准双接口等）签名之中，和其他请求参数一样都传递给支付宝。（详细可参见[6.2 数据传输配置](#)）

### ● 测试

- 1、单独测试快捷登录接口（详细可参见[7.2 调用快捷登录接口](#)）；
- 2、单独测试付款接口（如即时到账接口、担保交易接口、标准双接口等）（详细可参见[7.3 调用支付宝付款接口](#)）；
- 3、测试令牌授权码（参数 `token`）在付款接口（如即时到账接口、担保交易接口、标准双接口等）中是否有效果，即：是否省去了支付宝登录界面；
- 4、结合商户的完整业务逻辑测试整个流程：  
快捷登录→购买下单→使用支付宝付款的接口→订单处理完成。

## 目录

需要做的.....	2
● 集成.....	2
● 测试.....	3
目录.....	4
1 准备工作 .....	6
1.1 确认工作.....	6
1.2 准备工作.....	6
2 操作演示 .....	7
3 操作流程 .....	14
4 集成规范 .....	16
5 集成规范—ETAO .....	18
6 集成.....	19
6.1 基础信息配置.....	19
6.2 数据传输配置.....	20
6.3 返回数据处理.....	21
7 测试.....	22
7.1 步骤 .....	22
7.2 调用快捷登录接口.....	22
7.3 调用支付宝付款接口.....	23

7.4	服务器异步通知.....	24
7.5	日志记录.....	24
8	常见问题 .....	25
8.1	调试错误.....	25
8.2	订单更新异常.....	29
8.3	关掉了跳转页面订单未更新.....	31
8.4	支付宝的交易通知顺序.....	32
8.5	判断买家交易方式.....	33
9	技术服务 .....	34

## 1 准备工作

### 1.1 确认工作

#### 1.1.1 确认商家网站的基本信息

- 确认合同是否包含快捷登录接口功能。

合同查询，用签约的支付宝账号，登录支付宝官方网站，点击“商家服务”→“我的产品”。

- 确认网站的开发语言是什么

开发语言是用指定的一种语言来开发网站。支付宝提供的开发语言代码示例目前包含 ASP、PHP、ASP.NET、JAVA 四种。

- 确认网站编码格式是 GBK（GB2312）还是 UTF-8 的

查询网站使用的是何种编码格式，可通过在网站网页上右键，找到“编码”，打开扩展箭头后查看，默认选中的编码即为网站的编码格式。

#### 1.1.2 确认拿到的接口资料

- 确认拿到的是接口资料是快捷登录的接口资料
- 确认选择要使用的代码示例包是与商户网站的开发语言及编码格式都一致的代码示例包。

### 1.2 准备工作

#### 1.2.1 准备调试环境

- 本机电脑的调试环境

开发工具：Dreamweaver、Zend Studio、NetBeans、Microsoft Visual Studio、Myeclipse 等；

编辑工具：EditPlus、NotePad++、电脑自带的“记事本”等

环境配置：使用 ASP 和 ASP.NET 开发需要配有 IIS 服务；使用 PHP 开发需要 xampp 或 PHP 环境包；使用 JAVA 开发需要 JAVA 的环境配置。使用其他语言开发需要配置相对该语言的环境。

- 服务器的调试环境

1、服务器需要支持各开发语言对应的环境，如：支持 ASP 开发语言的 IIS 服务

2、能够把本地文件上传到服务器上的通道与权限，如：FTP，SFTP 传输。

### 1.2.2 确认接口放置位置

商户根据业务情况，需要确认支付宝的接口是用于什么样的场景，及确认所属业务流程中的第几个步骤，如：要把支付宝的接口功能放置在哪个页面里。

## 2 操作演示

- 先购买后登录

步骤一：选择商品并购买。



图 2.1 商户网站商品详细页面

步骤二：放入购物车，下订单。



图 2.2 商户网站收银台界面

步骤三：选择支付宝用户登录。



图 2.3 商户网站会员登录界面

步骤四：商户网站调用支付宝快捷登录接口，让买家填写支付宝登录账号与登录密码。

操作完登录后，须立刻到达填写订单页面（即[步骤五](#)），不能出现登录成功提示，甚至是让用户输入商户网站的账号和密码的界面等操作步骤。





图 2.4 商户网站会员登录界面

- 先登录后购买

步骤一：选择支付宝用户登录。



图 2.5 商户网站会员登录界面

步骤二：商户网站调用支付宝快捷登录接口，让买家填写支付宝登录账号与登录密码。



图 2.6 商户网站会员登录界面

步骤三：跳转回登录前页面

例如：用户浏览的登录前的界面是某个商品的详细界面（如下图）



图 2.7 商户网站商品详细页面

步骤四：放入购物车，下订单。



图 2.8 商户网站收银台界面

步骤五：填写订单信息

结算步骤：1、我的购物车      **2、填写核对订单信息**      3、成功提交订单



图 2.9 商户网站订单填写完成界面

步骤六：完成订单提交，立即付款。

商户网站创建订单,调用支付宝的付款接口，如：担保交易接口、即时到帐接口、标准双接口。此时商户网站的付款方式上只允许存在支付宝付款，不能有支付宝之外的其他在线支付方式。

结算步骤：1、我的购物车      2、填写核对订单信息      3、成功提交订单



Copyright 2007-2011 © Mbaobao.com, All Rights Reserved. 麦包包版权所有

图 2.10 商户网站订单提交成功界面

步骤七：完成支付宝的付款操作：



图 2.13 支付宝收银台-交易详细界面(担保交易)



图 2.14 支付宝收银台-选择支付方式界面（即时到帐）



图 2.15 支付宝收银台-选择交易方式界面（标准双接口）

步骤八：商户网站成功完成订单处理，显示支付成功。



© 2007-2011 Mbaobao All Rights Reserved. 麦包包版权所有

图 2.16 商户网站支付成功界面

📖 说明：

操作演示中的图片仅供参考（部分图片图源来自麦包包网站），请按照商户网站自身情况设计流程与界面。

### 3 操作流程

流程一：先登录后购买

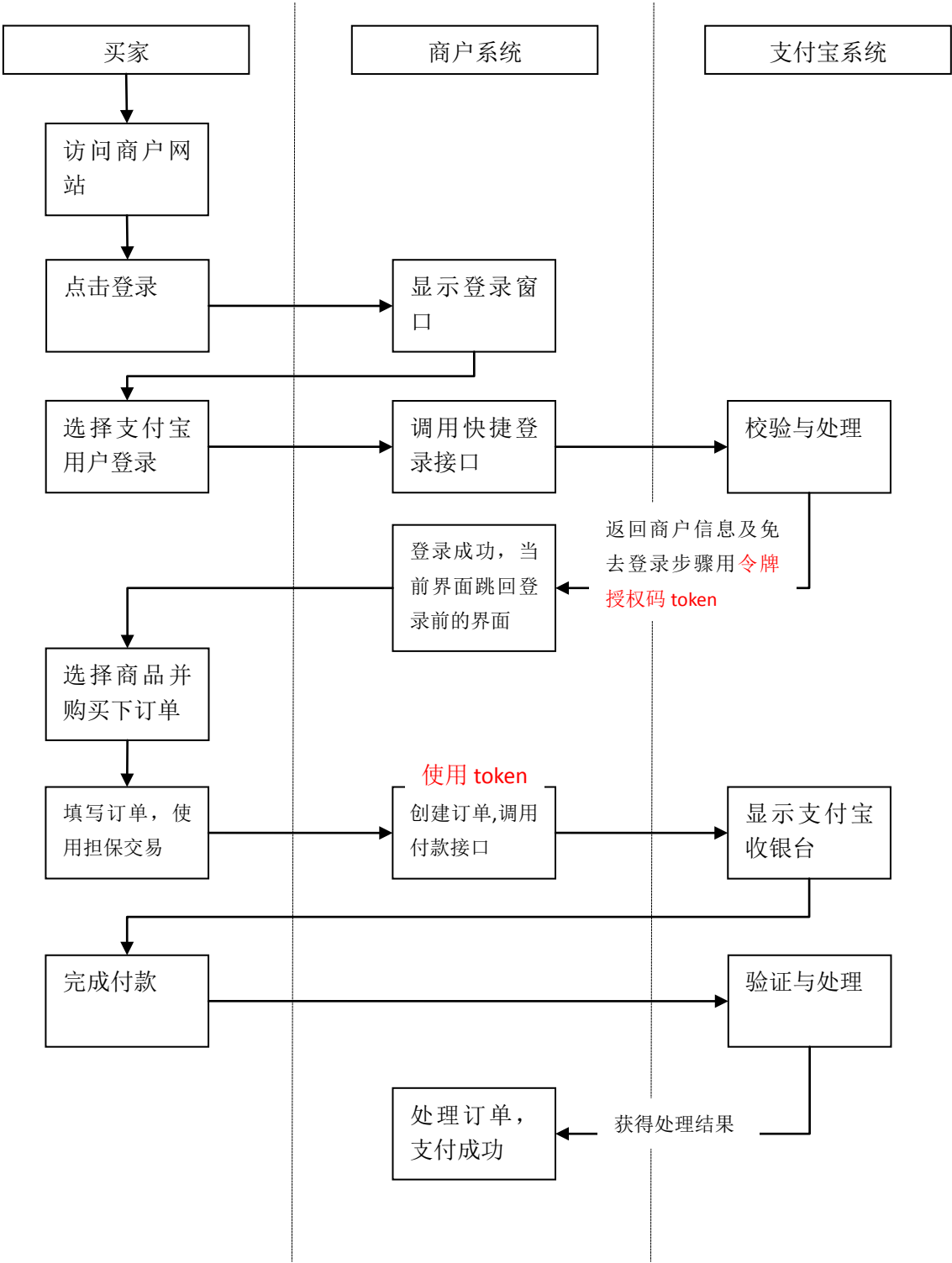


图 3.1 买家、商户网站、支付宝间流程界面——先登录后购买

流程二：先购买后登录

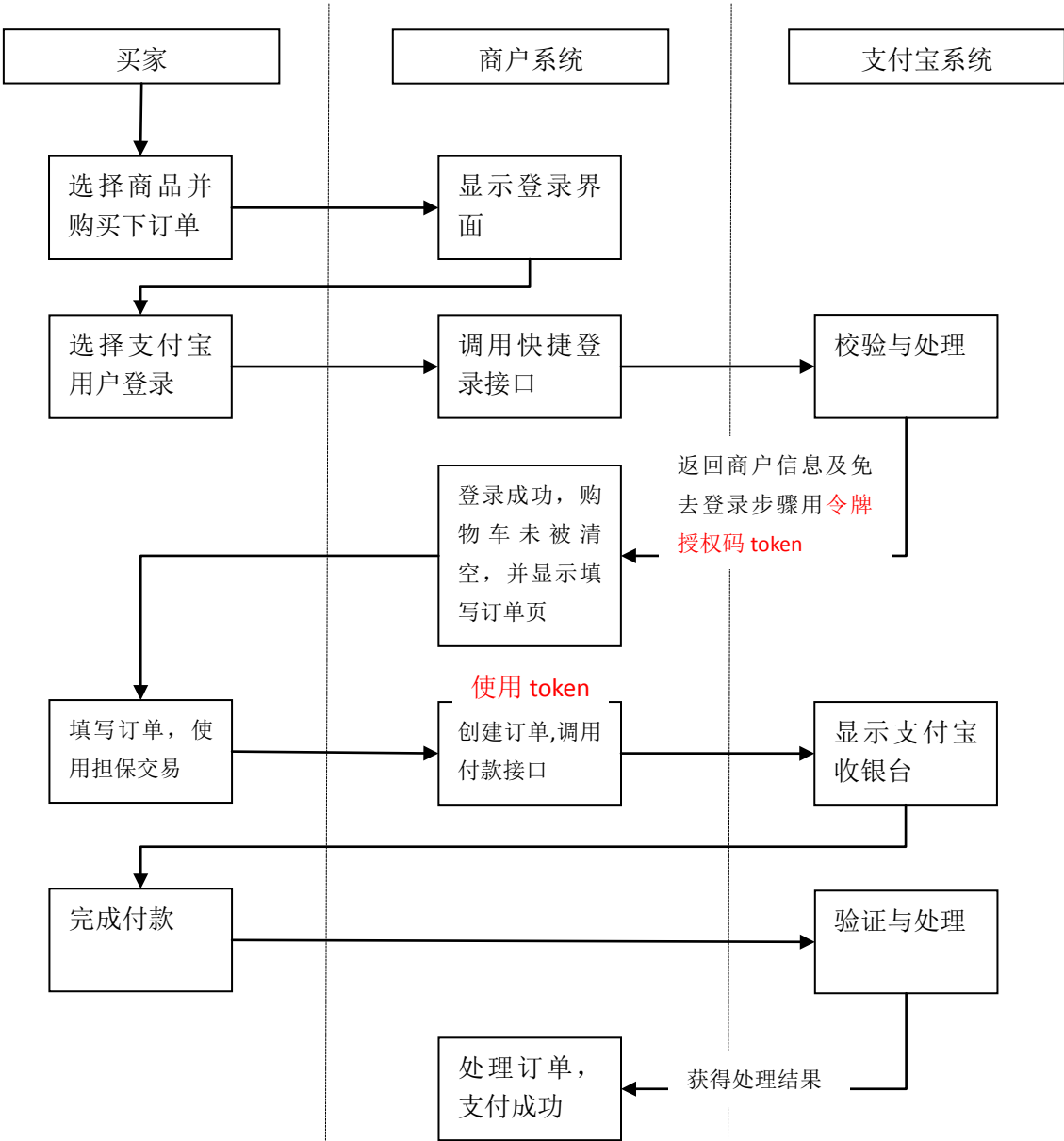


图 3.2 买家、商户网站、支付宝间流程界面——先购买后登录

4 集成规范

类别	规范点
图片	需选择支付宝提供的图片
流程	需拥有先登录后购买下订单、先购买后登录下订单两种业务流程。
流程	买家进入商户网站后，不论买家当前浏览的是哪个页面，当使用“快捷登录”入口登录，登录



	完成后需跳转回登录之前这个页面。
流程	当买家选好商品去结算时，通过“快捷登录”入口登录，登录完成后需跳转至商户网站的第一个填写订单的界面。
流程	当买家选好商品去结算时，通过“快捷登录”入口登录，登录完成后，需要保证购物车没有因为登录后而被清空
快捷登录	使用“快捷登录”入口登录后，不要出现登录成功的显示界面，更不能出现让用户设置商户网站的会员密码的操作界面等把支付宝会员转换为商户网站自己的会员的操作。
快捷登录	通过“快捷登录”入口登录后，显示会员名格式为：支付宝用户+姓名。
快捷登录	在商户网站中必须使用支付宝用户号（参数 <code>user_id</code> ）作唯一标识。
快捷登录	必须保存 <code>token</code> ，为担保交易接口、即时到账接口、标准双接口准备。
支付	当使用付款接口时，必须把 <code>token</code> 也加入请求参数，传递给支付宝
支付渠道	只要是通过“快捷登录”入口登录的，买家在支付时，只能出现支付宝支付一个选项，不能出现其他在线支付方式。
通知返回	必须在 <code>notify_url</code> 对应的页面文件中、 <code>return_url</code> 对应的页面文件中设置“是否已经做过相同的处理”的判断，如没有做过才执行这次的商户网站的业务逻辑处理。

图 4.1 集成规范点列表

- 需选择支付宝提供的图片

若需要用到与支付宝接口有关的一些按钮图片、LOGO，那么请务必使用各个接口资料中提供的图片的。一般这些图片是放在 `images` 文件夹中，如：`alipay_fl_v_m.png`。

- 需拥有两种流程

根据不同买家的操作习惯，商户网站需要拥有以下两种流程：

- 流程一：先登录商户网站，再购买下订单（如[图 3.1](#)）

当通过快捷登录后，商户网站需要做到当前界面直接跳回登录前的页面。如：支付宝会员在浏览商户网站中的某一件商品，此时点击登录，出现登录窗口，支付宝会员通过快捷登录入口登录进商户网站，商户网站需让当前登录窗口自动跳回支付宝会员刚才浏览的那一件商品的界面。

- 流程二：选好商品购买下订单时，再登录（如[图 3.2](#)）

当会员选好商品，把商品放入购物车，点击“结账”或“去收银台付款”等要下订单付款操作时，当前购物车界面跳到商户网站的登录窗口，支付宝会员通过快捷登录入口登录进商户网站后，商户网站需要做到保证购物车没有因为登录后而被清空，并且当前界面须跳转至商户网站的第一个填写订单的界面。

- 不能转换会员

当通过快捷登录后，商户网站此时请让买家直接进入下单流程即当前界面为填写订单的

页面，或者让当前界面跳回买家登录前的那一个页面。请不要出现登录成功的显示界面，更不能出现让用户设置商户网站的会员密码的操作界面等把支付宝会员转换为商户网站自己的会员的操作。

- 快捷登录后显示会员名的格式

当通过快捷登录后，商户网站需对支付宝的会员，显示“你好，支付宝用户 2088XXXXXXXXXXXX”、“亲爱的支付宝用户 2088XXXXXXXXXXXX”等，即“支付宝用户”+ 支付宝用户号（参数 `user_id`）的格式。如：你好，支付宝用户 2088XXXXXXXXXXXX。

- 必须使用参数 `token`。

当通过快捷登录等接口登录后，支付宝会记下该会员的登录状态，该状态的有效时间为 10 分钟，商户网站也会拿到一个令牌授权码（参数 `token`），这个 `token` 的有效时间为 30 分钟。当使用支付宝的付款接口时，支付宝会先判断该登录状态是否过期，再判断 `token` 是否有效。为了买家在支付时省去再登录支付宝这一步骤，商户网站须使用令牌授权码（参数 `token`）。

- 必须使用支付宝用户号（参数 `user_id`）做唯一标识

当通过快捷登录后，商户网站会获取到该会员的支付宝唯一用户号（参数 `user_id`）及其他信息。此时，请不要使用支付宝邮箱地址（参数 `email`）来匹配商户网站的会员数据，而是使用支付宝唯一用户号（即，需要在商户网站的会员系统中增加一个数据字段来保存这个支付宝唯一用户号），因为支付宝会员能够随时修改这个支付宝邮箱地址（参数 `email`），这样会导致商户网站的会员数据混乱。

- 在支付流程不能有支付宝之外的其他在线支付方式（可含线下收单）

当通过支付宝的快捷登录后，用户下订单付款时，商户网站的支付方式中只能让用户使用支付宝来付款，即商户网站的支付的界面中只能调用支付宝的担保交易接口或即时到账接口或标准双接口。

## 5 集成规范—etao

签约了 `etao` 的商户在集成时还需要注意以下内容：

类别	规范点
快捷登录	快捷登录接口中，执行完 <code>return_url</code> 文件后，程序须自动跳转到 <code>target_url</code> 指定页面路径中
快捷登录	在快捷登录接口中获取到的参数 <code>real_name</code> 需匹配一致性
快捷登录	商户网站必须显示当时传递给商户网站时的参数 <code>real_name</code> 的信息。
即时到账	从 <code>etao</code> 引入到商户网站的用户，当使用即时到账接口付款时，须把商户回传业务参数（参数 <code>item_orders_info</code> ）也加入请求参数中传递给支付宝。

图 5.1 etao 集成规范点列表

- 须从 `return_url` 自动跳转到 `target_url`

当快捷登录接口中的 `return_url` 文件被调用时，除了先判断这个用户是否登录，再核对这个用户的信息，让其登录进商户网站中之外，还要判断获取到的返回参数中是否存在 `target_url`，如果存在那么执行完 `return_url` 文件中的所有业务逻辑以后，需自动跳转到 `target_url` 这个页面路径去。

- 参数 `real_name` 需匹配

通过快捷登录接口获取到的真实姓名（参数 `real_name`，即支付宝会员名），与通过 `etao` 调用获取到的支付宝会员名或淘宝昵称（参数 `real_name`）有可能造成 `real_name` 不一致，因为同一个用户他的支付宝会员名（一般为真实姓名）与淘宝昵称是不一样的。商户可以依照自己的业务逻辑把这两个信息都储存下来，也可以只选取其中一个存储。总而言之，在做会员信息匹配时需要注意参数 `real_name` 的匹配问题。

- 商户网站必须显示参数 `real_name` 的信息

如果用户是通过快捷登录接口访问商户网站，商户网站上须显示此时获取到的参数 `real_name` 的信息，即支付宝用户真实姓名。如果用户是通过 `etao` 且该用户已登陆支付宝或淘宝并访问商户网站，商户网站上需显示此时获取到的参数 `real_name` 的信息，即支付宝用户真实姓名或淘宝昵称。

- 必须使用参数 `item_orders_info`

从 `etao` 引入到商户网站上的用户且该用户在 `etao` 的网站时已经登录，这类型的用户在使用支付宝付款时，商户须按照“即时到账交易接口(`create_direct_pay_by_user`).pdf”文档 4.3 章节“商户回传业务参数”的要求把参数 `item_orders_info` 也加入到即时到账接口的请求参数中，传递给支付宝。



**注意：**

如何判断是否从 `etao` 引入到商户网站上，只需要判断快捷登录接口中的 `return_url` 获取参数时是否存在 `target_url` 参数。如果存在 `target_url` 参数，那么该会员就是从 `etao` 过来的。

## 6 集成

### 6.1 基础信息配置

打开 `alipay_config` 文件，把所有信息全部填写完全。

- `partner`、`key`

需要用签约支付宝账号登录支付宝官方网站，在“商家服务”页面的右边框中点击查询：

<https://b.alipay.com/order/myOrder.htm?isFromMerchantWeb=true>。

- `seller_email`

收款支付宝账号，一般为签约的支付宝账号。

- `notify_url`、`return_url`

它们需要根据 `notify_url`、`return_url` 这两个页面文件所在服务器的 `http://` 格式的完整路径来设定。

- `input_charset`

商家网站可根据网站的编码（UTF-8 或 GBK/GB2312）下载对应编码的接口代码示例包。  
`input_charset` 参数的值是不需要去修改的，若想修改建议直接下载另一个编码格式的接口代码示例包。

## 6.2 数据传输配置

- 配置快捷登录接口

直接调用 `auth_authorize`（ASP.NET 中是 `default`）文件即可。

- 配置担保交易、即时到账、标准双接口

打开 `alipayto`（ASP.NET 中是 `default`）文件，在注释“请求参数”这部分中需要按照注释要求配置数据。


这些数据的获得方式是——

用户在完成一次购买的下单流程后，这笔订单便已经记录到商户的订单数据中，支付宝需要商户提供的数据正是商户的一部分订单数据，如：订单号（`out_trade_no`）、订单名称（`subject`）、订单描述（`body`）、订单总金额（`price`）、令牌授权码（`token`）等。商户只需要把这笔要付款的订单的数据赋值给支付宝指定的参数，就能完成数据传输的配置，如：

```
out_trade_no = "商户网站的唯一订单号"
subject      = "该笔订单的名称"
body         = "该笔订单的备注或描述"
price        = "该笔订单应付总额"
token        = "从快捷登录接口中获得的免去登录步骤的令牌授权码"
```

引号中的内容便是需要商户放置的订单数据。

---

 说明：

担保交易、即时到账、标准双接口改造注意——

在请求参数中增加一个请求参数：令牌授权码（参数 token），与其他参数一起参与签名，向支付宝发起请求。

---

## 6.3 返回数据处理

打开 return\_url（页面跳转同步通知）文件和 notify\_url（服务器异步通知）文件，分别在代码示例中注释指定位置增加商家网站自己的订单处理业务程序代码，即“//验证成功”这个判断中。

**第一步：** 获得支付宝的处理结果数据

获取众多返回数据中最重要的数据，如：令牌授权码（参数 token）、订单号（参数 out\_trade\_no）、订单名称（参数 subject）、订单描述（参数 body）、订单总金额（参数 total\_fee）、交易状态（参数 trade\_status）、支付宝交易号（参数 trade\_no）、买家支付宝账号（参数 buyer\_email）等，更多参数请参考页面跳转同步参数说明、服务器异步通知参数说明（见使用手册）。

**第二步：** 判断该笔订单是否在商户网站中已经做过处理

判断该笔订单是否有被处理过。由于同步通知和异步通知都会返回该笔订单的交易状态信息，那么该笔订单记录就有可能被重复刷新。因此，根据收到通知的这笔交易的订单号找到商户网站订单数据库中这笔订单详细信息，把这次收到的支付宝的交易状态或退款状态与商户网站订单数据库中这笔订单的交易状态或退款状态做对比。如果这笔订单没有做过处理则刷新该条记录；如果已经处理过该笔订单则不做处理。

**第三步：** 业务逻辑处理

编写业务订单业务流程处理程序代码。一般是在这里通过订单号，在订单数据库中更改该笔订单的相关数据，如订单状态从“未付款”改变成“已付款”等，并在订单数据库中存储此次订单对应的支付宝交易号是什么、买家账号是什么、处理时间是什么等。

---



**注意：**

1. 由于能够根据订单号（out\_trade\_no）SQL 语句查询得到商家订单数据库得到会员的相关信息，所以不需要再在返回文件 return\_url.asp 做登录验证等处理。
  2. session、cookie 在这 notify\_url 文件中会失效，因此不要使用。
  3. notify\_url 文件不能跳转到其他页面，因此请不要写页面自动跳转的程序
-

4. 各开发语言代码示例中，“//验证成功”的判断语句里的所有代码仅为样例代码，商户需要根据自身的业务情况来修改或重写。
5. 快捷登录中的 `return_url` 会被两种情况下调用，一是支付宝用户通过快捷登录接口触发，二是访问 `etao` 的买家通过点击 `etao` 里的某件商品（这件商品是商户网站上的）触发。即使如此，`return_url` 文件需要做的事没有多大改变，它依旧是需要先判断这个用户是否登录，再核对这个用户的信息，让其登录进商户网站中。唯一需要注意的是，商户网站需要增加判断——快捷登录接口中页面跳转同步通知参数 `target_url` 是否存在，这个判断能够让商户网站知道 `return_url` 文件是在什么情况下调用的，以便商户网站做下一步的业务处理，如：核对这名用户的 `real_name` 是否冲突、这名用户登录到商户网站后（即 `return_url` 文件执行完商户的业务逻辑处理后）需要跳转到 `target_url` 指定页面路径。
6. 如果通过 `etao` 来的买家没有登录支付宝也没有登录淘宝，此时快捷登录接口的 `return_url` 文件业务参数中只会收到 `target_url`、`notify_id` 参数。此时，商户可以跳过 `return_url` 原业务逻辑的执行程序，直接自动跳转到 `target_url` 页面地址去。

## 7 测试

### 7.1 步骤

- 第一步：** 仅仅把接口配置好，不要放在网站项目中。在本机单独对这个接口进行调试。调试内容包括支付、页面跳转同步通知；
- 第二步：** 本机调试没有问题后，再放入服务器中调试。调试内容包括：支付、页面跳转同步通知、服务器异步通知；
- 第三步：** 把调试好的接口与网站项目的下单流程进行衔接和融合；
- 第四步：** 在本机调试衔接到网站项目后的接口。调试内容包括：支付、页面跳转同步通知；
- 第五步：** 本机调试没有问题后，再放入服务器中调试。调试内容包括：支付、页面跳转同步通知、服务器异步通知；

### 7.2 调用快捷登录接口

- 第一步：** 访问 `index (default)` 文件（如：本机路径是 `http://localhost/接口文件夹名/index.asp`；服务器上路径是 `http://网站域名/接口文件夹名/index.asp`）
- 第二步：** 输入支付宝登录账号和登录密码，点击“登录”





图 7.1 支付宝快捷登录界面

**第三步：** 从页面跳转同步通知页面（`return_url` 文件）GET 方式获得令牌授权码（参数 `token`）及该支付宝用户相关信息，如支付宝用户 ID（参数 `user_id`）

**第四步：** 查看返回的页面是否打印出返回的数据，根据返回的数据来判定是验证失败还是成功。可用断点调试或用写日志函数 `LogResult(sWord)`记录数据库程序代码段的执行情况。

## 7.3 调用支付宝付款接口

**第一步：** 从快捷登录接口的页面跳转同步通知页面（`return_url` 文件）GET 方式获得令牌授权码（参数 `token`）

**第二步：** 保证订单号（`out_trade_no`）的唯一性，不要有重复的订单号出现。

**第三步：** 若包含下单流程，须购买一件商品，商品设定的金额 0.01 元，下单完成。订单总额用 0.01 元做支付测试。

**第四步：** 访问 `index` 文件（如：本机路径是 `http://localhost/接口文件夹名/index.asp`；服务器上路径是 `http://网站域名/接口文件夹名/index.asp`）即：进入确认订单页面。

**第五步：** 点击支付宝付款的按钮，页面会跳转至支付宝收银台；

**第六步：** 在支付宝的收银台里，完成付款操作。

**第七步：** 从页面跳转同步通知页面（`return_url` 文件）GET 方式或从服务器异步通知页面（`notify_url` 文件）POST 方式获得该次交易的支付宝处理结果数据。

**第八步：** 查看返回的页面是否打印出返回的数据，根据返回的数据来判定是验证失败还是成功。可用断点调试或用写日志函数 `LogResult(sWord)`记录数据库程序代码段的执行情况。

一般情况下，页面能成功到达支付宝收银台，没有出错、没有乱码，那么就不会有什么问题。

## 7.4 服务器异步通知

- 第一步：** 必须把接口放置在服务器里做调试；
- 第二步：** `notify_url` 的路径互联网上可以访问得到，且只显示 `fail` 或无通知参数。其他如空白或是程序错误提示等内容，均属程序执行出错；
- 第三步：** 付款完成一次 1 分钱的交易；
- 第四步：** 用写日志函数 `LogResult(sWord)` 记录验证是否通过，记录这验证的两个判断中三个参数的数据。

各开发语言验证的代码位置——

**ASP:** `class\alipay_notify.asp` 文件的 `VerifyNotify()` 函数中的 `if mysign = request.Form("sign") and responseTxt = "true" then`，三个参数是 `mysign`、`sign`、`responseTxt`。

**PHP :** `lib\alipay_notify.class.php` 文件的 `verifyNotify()` 函数中的 `if (preg_match("/true$/i",$responseTxt) && $mysign == $_POST["sign"])`，三个参数是 `$verify_result`、`mysign`、`sign`。

**ASP.NET(C#):** `alipay_notify.aspx.cs` 文件的 `if (responseTxt == "true" && sign == mysign)`，三个参数是 `responseTxt`、`sign`、`mysign`。

**JAVA:** `\src\com\alipay\util\AlipayNotify.java` 文件的 `verify(Map<String, String> params)` 函数中的 `if (mysign.equals(sign) && responseTxt.equals("true"))`，三个参数是 `sign`、`mysign`、`responseTxt`。

- 第五步：** 用写日志函数 `LogResult(sWord)` 记录数据库程序代码段的执行情况；
- 第六步：** 用写日志函数 `LogResult(sWord)` 记录整个页面打印出来的信息是否只有 `success`。

## 7.5 日志记录

在服务器异步通知页面文件（`notify_url`）、页面跳转同步通知页面文件（`return_url`）中都包含着 `LogResult(sWord)` 写日志函数，这个函数的功能是调试支付宝接口、记录通知情况等。

**使用方法：**

- 1、事先确保调试用本机电脑或服务器中是否有对日志文件夹（`log` 文件夹）或日志文件（`log.txt`）开启读写权限。
- 2、确认日志文件夹和日文件在哪。一般生成的日志会直接存放在 `LOG` 日志文件或文件夹中。`JAVA` 代码的默认日志存放地点在 `D` 盘中，所以需要先设置日志创建的本机电脑绝对路径。



3、把函数 `LogResult(sWord)` 放置到想要调试的地方,编写好想要输出来观察数据的 `sWord` 的值,如:

`LogResult(sWord)` ("这里写入想要调试的代码变量值,或其他运行的结果记录")

4、当接口在运行并执行完该段代码后,日志文件夹(log 文件夹)或日志文件(log.txt)里就会被写进新的东西。

5、打开日志文件夹(log 文件夹)或日志文件(log.txt),就能看到想要观察的数据。

#### 建议:

1、在调试服务器异步通知页面文件(notify\_url)、页面跳转同步通知页面文件(return\_url)这两个页面文件时(尤其是调试服务器异步通知页面文件,因为是商户网站服务器与支付宝服务器间的交互,无法通过开发软件自带调试工具来监控运行的情况),可以通过写日志函数 `LogResult(sWord)` 来监测程序运行情况,这样在调试接口时可以大大的提高检查错误原因的速度。

2、可以把写日志函数 `LogResult(sWord)` 改为把支付宝发来的通知记录存入商户的数据库中,使得商户网站能更好的管理和记录支付宝发来的通知,以便在出现特殊情况时能够利用其记录来查明原因。

## 8 常见问题

### 8.1 调试错误

当点击支付宝付款,页面跳转到支付宝的网站时如果出现“调试错误,请回到请求来源地,重新发起请求。”(如下图所示),是因为接口在开发时出现问题,这些问题的原因有,传输给支付宝的数据不符合支付宝要求的格式、该接口对应的权限没有打开(即签约的合同中没有包含该接口)、商户网站的编码格式乱码等等。



图 8.1 调试错误界面

- 错误代码 HAS\_NO\_PRIVILEGE

原因：没有该接口的权限

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=243729&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=243729&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_PARTNER\_EXTINTERFACE

原因：没有该接口的权限

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=243730&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=243730&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_PARTNER

原因：合作身份者 ID（partner）填写错误

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=243727&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=243727&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_ARGUMENT

原因：参数或参数对应的值的格式不正确

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241183&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241183&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_SIGN

原因：参数格式或交易安全校验码（key）不正确，编码问题

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241184&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241184&sh=Y&tab=null&info_type=9)

- 错误代码 seller\_not\_in\_specified\_sellers

原因：收款支付宝账号（参数 seller\_email）不是签约的支付宝账号

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241185&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241185&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_LOGISTICS\_FORMAT

原因：物流参数格式问题

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241186&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241186&sh=Y&tab=null&info_type=9)

- 业务数据被篡改

原因：篡改请求链接、物流参数格式问题

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241193&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241193&sh=Y&tab=null&info_type=9)

- 错误代码 TRADE\_NOT\_ALLOWED\_PAY

原因：该笔交易已经关闭或已经付过款，不能再付款。

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241187&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241187&sh=Y&tab=null&info_type=9)

- 错误代码 SELLER\_NOT\_EXIST

原因：收款支付宝账号（参数 seller\_email）填写错误

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241188&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241188&sh=Y&tab=null&info_type=9)

- 错误代码 BUYER\_SELLER\_EQUAL

原因：买家支付宝账号与卖家支付宝账号相同，不允许付款。

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241189&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241189&sh=Y&tab=null&info_type=9)

- 错误代码 TRADE\_DATA\_MATCH\_ERROR

原因：第一次请求给支付宝的数据与第二次不匹配

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241190&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241190&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_SERVICE

原因：服务名称错误、请求参数全部丢失

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241191&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241191&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_FEE\_PARAM

原因：金额格式不正确

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241194&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241194&sh=Y&tab=null&info_type=9)

- 错误代码 ROYALTY\_FORAMT\_ERROR

原因：分润格式不正确

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241195&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241195&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_SIGN\_TYPE

原因：签名类型错误

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241196&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241196&sh=Y&tab=null&info_type=9)

- 错误代码 SELF\_TIMEOUT\_NOT\_SUPPORT

原因：自定义超时功能不支持

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241197&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241197&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_ANTI\_PHISHING\_KEY

原因：防钓鱼时间戳不正确

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=241198&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=241198&sh=Y&tab=null&info_type=9)

- 错误代码 ILLEGAL\_EXTER\_INVOKE\_IP

原因：防钓鱼的获取用户客户端 IP 没有设置或设置不正确

解决方法：

[http://help.alipay.com/support/help\\_detail.htm?help\\_id=243090&sh=Y&tab=null&info\\_type=9](http://help.alipay.com/support/help_detail.htm?help_id=243090&sh=Y&tab=null&info_type=9)

## 8.2 订单更新异常

当买家通过支付宝接口支付完成后，发现商户网站的订单系统中自己的订单没有同步被处理或是商户网站中对这笔交易做了多次处理（如付了一次钱，却反复充值多次）。出现这些情况的原因可能有：商户网站中订单更新的页面文件执行有问题、商户网站所在服务器的配置上出现问题。

**解决方法:**

检查服务器异步通知页面（参数 `notify_url` 对应的页面文件）与同步跳转返回页面参数 `return_url` 对应的页面文件）

**第一步:** 检查 `return_url`、`notify_url` 页面文件

A: 合作身份者 ID 和安全校验码是否设置正确，且没有空格，没有填反。

B、`return_url`、`notify_url` 的路径互联网上需要能够访问得到，确保访问到该页面时，页面没有为空白或开发软件自带系统提示错误信息

C、请求参数 `show_url`、`return_url`、`notify_url` 不要带自定义参数，如：

`http://www.xxx.com/alipay/return_url.asp?id=123`

`?id=123`，“？”以后带的参数都是自定义参数

D、检查编码格式是否设置正确

检查乱码的方法，可用写日志函数来记录中文部分的数据是否乱码。如果乱码则确定是编码格式设置不一致，导致乱码。具体请检查编码格式参数 `input_charset`、页面属性编码、网站项目整体编码等，确保传递数据时没有乱码。

E、因为返回的 `return_url` 有效时间是 1 分钟，超过 1 分钟，这个 URL 链接就会失效，`responseTxt` 会为 `false`，所以需要检查该链接是否已经失效

F、`notify_url` 页面里，有且只有输出 `success`，且必须保证为空白页面，无任何 HTML 标签、空格、回车换行等字符。

各开发语言输出 `success` 代码：

ASP.NET: `Response.Write("success");`

ASP: `response.write "success"`

PHP: `echo "success";`

JAVA: `out.println("success");`

G、`notify_url` 页面中不要有 `session` 或 `cookies`，不要做页面自动跳转

**第二步:** 检查服务器配置（以下为参考检查方向）

A、服务器是否开启写入权限

B、DNS 解析，是否能够解析支付宝 IP。支付宝 IP 范围是：

`121.0.26.0/23`（IP 范围：`121.0.26.1`---`121.0.27.254`）

`110.75.128.0/19`（IP 范围：`110.75.128.1`——`110.75.159.254`）

`115.124.16.0/24`（IP 范围：`115.124.16.1`——`115.124.16.254`）

- C、防火墙是否有拦截
- D、端口 443 或 80 是否有开启，且不是假死状态，也没有被占用
- E、服务器的 IP 地址是否有做过更换，如果有做过更换，则请把 notify\_url 页面文件重命名，参数 notify\_url 的值也一并更新。

**第三步： 调试**

A、检查程序运行到 alipay\_notify 文件的 verifyReturn()、verifyNotify()函数中，在 mysign 与 sign 是否相等的判断与 responseTxt 是否为 true（PHP 代码：\$verify\_result）的判断这两个判断上是哪个判断验证未通过。

**sign 与 mysign 不等的情况有：**

安全校验码（参数key）设置有错误、参数格式设置错误与编码乱码（见[第一步](#)）

**responseTxt 不是 true 的情况有：**

合作身份者ID（参数partner）设置有错误、服务器配置（见[第二步](#)）、页面跳转返回的地址栏里的链接超过 1 分钟的有效时间

B、在调试 notify\_url 页面时，必须把接口放置在服务器里做调试

C、检查程序运行到商户网站自己编写的业务逻辑处理程序代码段时，断点调试或使用写日志函数（函数名：LogResult）逐行检查代码，监控各个参数变量



**注意：**

商户网站中对这笔交易做了多次处理（如付了一次钱，却反复充值多次）这类情况，按照解决方法修正了商户网站的程序或服务器后，请务必在页面跳转同步通知页面文件（return\_url）与服务器异步通知页面文件（notify\_url）中，在商户的业务逻辑处理的程序代码执行之前，一定要先判断该次处理是否已经被做过相同的处理（即，该笔交易在已经付过款后，商户网站里的订单是否已经做过订单更新），如：是否已经为会员充值过一次。没有做过相同的处理，才执行下面的程序。（见[6.3 返回数据处理 第二步](#)）

## 8.3 关掉了跳转页面订单未更新

买家支付完成后，当前界面会停留在支付宝的交易付款成功的提示界面上，如：



图 8.2 支付宝交易成功提示界面

该页面有短暂的停留时间，买家看到这个提示信息后有可能会把当前页面关掉，导致页面没有自动跳转回商户网站（`return_url`）使得商户的业务数据没有被处理（如：订单没有更新）。出现这种情况的原因是：商户网站的支付宝接口并没有利用服务器异步通知页面文件（`notify_url`）或是该页面在订单执行过程中也出现异常。

#### 解决方法：

打开`return_url`页面文件，找到已经编写好的业务逻辑处理程序代码段，对其增加防止该笔交易被重复处理判断。打开`notify_url`页面文件，找到可插入商户的业务逻辑处理程序代码的地方，编写与`return_url`页面文件中相类似的程序，也需要包含防止该笔交易被重复处理的判断。如果是因为`notify_url`页面文件更新异常导致订单没有更新，其解决方法请参考[8.2 订单更新异常](#)。

#### 修改完成后效果：

当买家关掉自动跳转回商户网站的返回页面（参数 `return_url` 对应的 `return_url` 返回文件）时，商户的业务处理依旧被 `notify_url` 页面文件所执行，因此就不会发生业务没有同步的现象出现（如：订单没有被更新）

## 8.4 支付宝的交易通知顺序

担保交易、标准双接口中有诸多的交易状态及退款状态，如：`WAIT_BUYER_PAY`（等待买家付款）、`WAIT_SELLER_SEND_GOODS`（买家已付款，等待卖家发货）、`WAIT_BUYER_CONFIRM_GOODS`（卖家已发货，等待买家收货）、`TRADE_FINISHED`（买家已收货，交易完成）。每当支付宝的交易发生变更时，交易状态也会变更，如果有设置`notify_url`（服务器异步通知路径），那么支付宝都会发送产生变化的这笔交易的通知信息。商户就能够根据这些数据来把商户网站订单系统里的这笔订单做同步更新，达到与支付宝的订单同步的效果。



交易的通知顺序是以第一次发送这种交易状态和退款状态的通知时间来排序的。

交易状态（**TRADE\_STATUS**）的顺序是：

WAIT\_BUYER\_PAY（等待买家付款）→WAIT\_SELLER\_SEND\_GOODS（买家已付款，等待卖家发货）→WAIT\_BUYER\_CONFIRM\_GOODS（卖家已发货，等待买家收货）→TRADE\_FINISHED（买家已收货，交易完成）

退款状态（**REFUND\_STATUS**）的顺序是：

- 有收到货的情况

WAIT\_SELLER\_AGREE（等待卖家同意退款）→WAIT\_BUYER\_RETURN\_GOODS（卖家同意退款，等待买家退货）→WAIT\_SELLER\_CONFIRM\_GOODS（买家已退货，等待卖家收到退货）→REFUND\_SUCCESS（卖家收到退货，退款成功，交易关闭）

- 无收到货的情况

WAIT\_SELLER\_AGREE（等待卖家同意退款）→REFUND\_SUCCESS（卖家同意退款，退款成功，交易关闭）



注意：

- 1、退款成功后，交易状态会变为 **TRADE\_CLOSED**
- 2、退款流程中，买家申请退款时，卖家可以拒绝退款，那么退款状态会变为 **SELLER\_REFUSE\_BUYER**，买家还是可以再次申请退款，退款状态又会变为 **WAIT\_SELLER\_AGREE**。

## 8.5 判断买家交易方式

因为标准双接口是当进入支付宝收银台时，买家可以选择使用担保交易功能付款和即时到账付款两种方式中的一种，但这个付款方式卖家是无法知道买家是选择的哪一种的。

那么要判断买家是用的哪一种付款方式可以通过买家付款完成后，商户利用 **return\_url**（页面跳转同步通知文件）和 **notify\_url**（服务器异步通知文件），根据刚获得的交易状态与商户网站订单系统中记录的订单状态来做对比，以及根据交易状态的顺序就可以推断出——

- 如果收到的交易状态是 **WAIT\_SELLER\_SEND\_GOODS**（买家已付款，等待卖家发货），商户网站订单中记录的状态是 **WAIT\_BUYER\_PAY**（等待买家付款），那么买家就是用的担保交易的付款方式；
- 如果收到的交易状态是 **TRADE\_FINISHED**（买家已收货，交易完成），商户网站订单中记录的状态是 **WAIT\_BUYER\_PAY**（等待买家付款），那么买家就是用的即时到账的

付款方式。

## 9 技术服务

在开发或使用支付宝接口时，产生疑问或出现问题，可点击下面的链接填写表单，提交技术问题，支付宝技术支持人员会主动联系并及时处理。

<https://b.alipay.com/support/helperApply.htm?action=supportHome>

- 支付宝论坛接口集成专区：<http://club.alipay.com/thread.php?fid=703>

在论坛中可以下载接口资料，查找解决方案来自主解决问题，或是发帖寻求帮助。

- 支付宝商户客服热线：0571-88158090