

支付宝发货接口集成教程

服务名称：send_goods_confirm_by_platform

版本号：1.1

目录

| | |
|------------------|----|
| 目录 | 2 |
| 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 |
| 1.4 技术服务 | 4 |
| 2 准备工作 | 4 |
| 2.1 确认工作 | 5 |
| 2.2 准备工作 | 5 |
| 3 集成 | 6 |
| 3.1 文件说明 | 6 |
| 3.2 基础信息配置 | 7 |
| 3.3 数据传输配置 | 7 |
| 3.4 返回数据处理 | 8 |
| 4 测试 | 9 |
| 4.1 步骤 | 9 |
| 4.2 发货 | 9 |
| 4.3 日志记录 | 10 |
| 5 常见问题 | 10 |
| 5.1 调试错误 | 10 |

| | |
|---------------------|----|
| 5.2 支付宝的交易通知顺序..... | 13 |
|---------------------|----|

1 文档说明

1.1 功能描述

结合接口技术文档，对代码示例进行解读，以便能快速的把接口整合到商户网站项目中。

1.2 阅读对象

支付宝会员通用登录接口技术文档是面向具有一定的网站开发能力，了解 ASP、PHP、JAVA、ASP.NET 等开发语言中的一种及 SQL 数据库语言的网站开发、维护和管理人员。

1.3 业务术语

| 术语名称 | 术语解释 |
|------|--|
| 请求 | 通过 HTTP 协议把需要传输的数据发送给接收方的过程 |
| 返回 | 页面跳转同步通知。支付宝根据得到的数据处理完成后，当前页面从支付宝的页面自动跳转回商户的网站，同时携带处理完成的结果信息反馈给商户网站。 |
| 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。 |

1.4 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

2 准备工作

2.1 确认工作

2.1.1 确认商家网站的基本信息

- 确认合同是否包含发货接口功能。

合同查询，用签约的支付宝账号，登陆支付宝官方网站，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。

2.1.2 确认拿到的接口资料

- 确认拿到的是接口资料是发货的接口资料
- 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。

2.2 准备工作

2.2.1 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。

2.2.2 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。

3 集成

3.1 文件说明

3.1.1 文件结构

各开发语言的代码示例中，基本包含的文件是：**alipay_config**（配置文件）、**index** 或 **default**（发货模板界面文件）、**sendgoods**（接口接入文件）、**alipay_function**（公用函数类文件）、**alipay_service**（请求处理类文件）、**alipay_notify**（通知返回处理类文件）、**log**（日志文件夹或日志文本）。

- 配置文件

它用于设置支付宝账号信息，配置文件路径等不需要动态变动的基础信息，以便被接口中其他几个文件调用。

- 发货模板界面文件

该页面文件用于商户网站后台管理的发货功能界面。

- 接口接入文件

该页面文件用于商户网站后台的发货发货功能管理与支付宝发货接口相接。它是生成请求支付宝数据的生成页面，把发货单号、物流公司名称、物流类型、支付宝交易号以及其它的一些参数按照支付宝指定的规则传递给支付宝，通过 **form** 表单 **POST** 或 **GET** 方式提交到支付宝。

- 公用函数类文件

它是辅助请求处理类文件、通知返回处理类文件的核心处理文件。它包含重复被调用的公共函数。不需要去修改。

- 请求处理类文件

它是辅助入口文件的请求数据处理的核心处理类文件。不需要去修改。

- 通知返回处理类文件

它是辅助返回文件、通知文件的通知返回数据处理的核心处理文件。不需要去修改。

- 日志

它用于记录对接口进行断点调试时的信息。可在接口程序的关键步骤处打开写日志函数，在程序执行后，便可根据在该文件中查到的程序执行情况来定位错误。

3.1.2 各开发语言其他文件说明

- ASP

`alipay_md5.asp`: ASP 开发语言中是没有现有的 MD5 签名函数的。因此，它是针对 MD5 签名的类文件。它支持 UTF-8 和 GBK(GB2312)两种编码。不需要更改。

- JAVA

`src/com.alipay.util /Md5Encrypt.java`: 它是针对 MD5 签名的类文件。

`src/com.alipay.util /UtilDate.java`: 它是发货号生成的类文件

`src/filters`: 该文件夹是编码过滤器。用于整个网站项目中规范和统一编码。它是自动生成的配置文件，可对其忽略。

3.2 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陆支付宝官方网站，在“商家服务中心”→“技术服务”的页面中查询，<https://b.alipay.com/support/helperApply.htm?action=selfIntegration>。

- 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。

3.3 数据传输配置

如果不需要使用发货模板界面文件，只想把支付宝接口与商户网站后台的发货发货功能管理紧密结合，那么需要配置 `sendgoods` 文件（入口文件），不使用 发货模板界面文件。

打开口文件，“请求参数”这部分必填的数据便是要配置的参数。

这些数据的获得方式是——

商户网站后台管理的发货界面输入的发货信息时获取。支付宝需要的发货信息，如：支付宝交易号（`trade_no`）、物流公司名称（`logistics_name`）、发货单号（`invoice_no`）、物

流类型（`transport_type`）等。商户只需要把这些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：

```
trade_no      = "支付宝交易号"
logistics_name = "物流公司名称"
invoice_no    = "发货单号"
transport_type = "物流类型"
```

引号中的内容便是需要商户放置的发货数据。

3.4 返回数据处理

打开支付宝接口入口文件（`sendgoods`），在代码示例中注释指定位置增加商家网站自己的发货处理业务程序代码。

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：支付宝交易号（`trade_no`）、商户网站订单号（`out_trade_no`）、交易状态（`trade_status`）、发货时间（`send_time`）等，更多参数请参考页面跳转同步参数说明（见技术文档）。

第二步： 判断该次发货是否在商户网站中已经做过处理

判断该次发货是否有被处理过。由于一次交易能够多次操作发货（实际上第一次操作发货成功后，再多次操作是不会让交易变更的）从支付宝获取的信息也是成功的返回数据，但商户网站的业务处理程序会因此可能被重复执行。因此，先要获取到发货操作完成后的这笔交易的交易状态，再找到商户网站订单数据库中这笔订单详细信息，把这两者的交易状态做对比，来判断发货操作是否有被重复执行。如果这笔发货没有做过处理则刷新该条记录；如果已经处理过该笔发货则不做处理。

第三步： 业务逻辑处理

编写业务发货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“已付款”改变成“卖家已发货”、记录发货时间、记录发货操作员等。



远程 XML 解析注意：

- 1、asp、asp.net 开发语言的代码是否正常支持远程 XML 解析与 IIS 配置有关
- 2、php、java 开发语言的代码是否正常支持远程 XML 解析与服务器的配置有关
- 3、php 要支持远程 XML 解析须支持 DOMDocument 与 SSL

4 测试

4.1 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。

第二步： 本机调试没有问题后，再放入服务器中调试。

第三步： 把调试好的接口与网站项目的发货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。

第五步： 本机调试没有问题后，再放入服务器中调试。

4.2 发货

第一步： 使用担保交易接口或双功能接口完成一笔付款，并记录下该笔交易的支付宝交易号。

| 所有买入交易 | 进行中的交易 | 等待付款 | 等待发货 | 等待确认收货 | 退款交易 | 成功交易 | 失败交易 |
|---|------------------|----------------|---------------------------------------|---|-------|--------|---------|
| 交易笔数：5 笔 | | 排序：按交易创建时间从近到远 | 选择显示信息 | 下载账单：Excel格式 Txt格式 | | | |
| 类型 | 创建时间 | 名称 | 商户订单号 交易号 | 对方 | 金额(元) | 状态 | 操作 备注 |
|  | 2010.12.17 19:30 | 订单名称 | 2010121719309 交易号：2010121702571835 |  | 0.01 | 等待对方发货 | 退款 详情 |

图 4.1 支付宝交易管理界面

第二步： 访问发货模板界面文件（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：进入发货页面。

第三步： 填写发货信息，点击“发货”按钮；

支付宝发货通道

| | |
|---------|-------------------------------------|
| 支付宝交易号： | <input type="text"/> |
| 发货类型： | 快递 <input type="button" value="v"/> |
| 物流公司名称： | <input type="text"/> |
| 物流发货单号： | <input type="text"/> |
| | <input type="button" value="发 货"/> |

图 4.2 支付宝发货接口模板界面

第四步： 当前页面自动跳转到支付宝页面上并以 XML 格式呈现支付宝的处理结果。

4.3 日志记录

在 `alipay_function`（公用函数类文件）中存在着 `log_result(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志文件在哪。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 `log_result(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：

```
log_result("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 xml 返回数据处理时，可以通过写日志函数 `log_result(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 `log_result(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返回的数据，以便在出现特殊情况时能够利用其记录来查明原因。

5 常见问题

5.1 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 5.1 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=241181&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=241182&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

- 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

- 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

- 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

- 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

- 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

- 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

5.2 支付宝的交易通知顺序

发货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发货系统里的这笔发货做同步更新，达到与支付宝的发货同步的效果。

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。