

即时到帐接口集成教程

服务名称：create_direct_pay_by_user

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	8
3.4 返回数据处理	8
4 测试	9
4.1 步骤	9
4.2 支付	10
4.3 页面跳转同步通知	11
4.4 服务器异步通知	11
4.5 日志记录	12

5 常见问题	13
5.1 调试错误.....	13
5.2 订单更新异常.....	16
5.3 关掉了跳转页面订单未更新.....	18

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

- 确认合同是否包含即时到帐接口功能。

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即时到帐的接口资料
- 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**（直接付款模板界面文件）、**alipayto**（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）、**notify_url**（服务器异步通知处理页面文件）、**return_url**（页面跳转同步通知处理页面文件）、**log**（日志文件夹或日志文本）、**images**（支付宝图片文件夹）。

- 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。

- 直接付款模板界面文件

该页面文件用于商户网站中没有下订单等业务流程，只需要买家填写好相关付款信息后，进入支付宝收银台界面完成支付。

- 接口接入文件

商户网站中有下订单等业务流程时，需要把支付宝的功能接入其中，可直接使用此文件，略过直接付款模板界面文件，作为支付宝接口入口页面文件。它是连接商户网站和支付宝收银台的主页面，把商户网站的商品名称，订单号，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 **URL 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。

- 服务器异步通知处理页面文件

每次交易状态变更，支付宝都会主动向该页面发出通知，使得该页面被调用，以便能够自动执行商户的业务逻辑程序。

- 页面跳转同步通知处理页面文件

当付款完成以后，当前页面会自动跳转到该页面。该页面用于处理付款完成以后支付宝返回的数据，商家可以根据处理结果在商户网站上显示给用户或进入下一步流程操作。

- 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 GB2312(GBK)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: 它是订单号生成的类文件

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>。

- 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。

- show_url

它是订单商品展示页面，建议直接设置成商家网站的会员订单列表首页，如“我的订单”。

若指向订单详细页面，会带上订单号的自定义参数（如：
show_url=http://www.xxx.com/alipay/order_detail.asp?id=123）有可能产生接口报错。

- **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。

- **mainname**

收款方名称。显示在直接付款模板界面文件中“收款方”一栏中。用于提示买家收款者是谁。一般该处建议设置为商户的公司网站名称。

3.3 数据传输配置

如果不需要使用直接付款模板界面文件，只想把支付宝接口与商户网站的下单流程紧密结合，那么需要配置 alipayto 文件（入口文件），不使用直接付款模板界面文件。

打开口文件，“/*以下参数是需要通过下单时的订单数据传入进来获得*/”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——

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（total_fee）、支付方式（paymethod）、网银选择（defaultbank）、自定义变量（extra_common_param）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"
total_fee    = "该笔订单应付总额"
paymethod    = "买家选择的支付方式"
defaultbank  = "买家选择的网上银行"
extra_common_param = "商户自行定制的数据"
```

引号中的内容便是需要商户放置的订单数据。

3.4 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：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`）、自定义参数（参数 `extra_common_param`）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或存储了会员的相关信息的自定义参数（`extra_common_param`），所以不需要再在返回文件 `return_url.asp` 做登陆验证等处理。
2. `session`、`cookie` 在这两份文件中会失效，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。

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。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；

4.2 支付

第一步： 保证订单号（out_trade_no）的唯一性，不要有重复的订单号出现。



图 4.1 支付宝接口入口界面

第二步： 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。

第三步： 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。

第四步： 点击支付宝付款的按钮，页面会跳转至支付宝收银台；



图 4.2 支付宝收银台界面

第五步： 在支付宝的收银台里，点击“确认无误，付款”，完成支付宝里的付款操作。

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。

4.3 页面跳转同步通知

第一步： 在支付宝收银台里，完成付款操作，支付宝页面显示付款成功，如图：



图 4.3 支付宝付款成功界面

第二步： 等待几秒，页面自动跳转到返回页面（参数 `return_url` 对应的路径地址），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。

第三步： 断点调试或用写日志函数 `log_result()` 记录数据库程序代码段的执行情况。

4.4 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 `log_result()` 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: `alipay_notify.asp` 文件的 `return_verify()` 函数中的 `if mysign = request.Form("sign") and responseTxt = "true" then`，三个参数是 `mysign`、`sign`、`responseTxt`。

PHP : alipay_notify.php 文件的 return_verify() 函数中的 if (preg_match("/true\$/i",\$verify_result) && \$this->mysign == \$_GET["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: alipay_notify.jsp 文件的中的 if (mysign.equals(request.getParameter("sign"))&& responseTxt.equals("true")), 三个参数是 sign、mysign、responseTxt。

第五步: 用写日志函数 log_result()记录数据库程序代码段的执行情况;

第六步: 用写日志函数 log_result()记录整个页面打印出来的信息是否只有 success。

4.5 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 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 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。

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 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](#)

- 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](#)

- 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 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](#)

- 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](#)

- 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_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](#)

- 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](#)

- 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](#)

- 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](#)

- 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](#)

- 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：

在 alipayto（接口接入文件）中，找到请求参数 `exter_invoke_ip`，把它设置成能够自动获取当前客户端 IP 地址的编程方法，让程序自动运行时能够得到当前客户端 IP 地址并赋值给它，如：`exter_invoke_ip="202.1.1.2"`。该参数一旦设置后，变为必填参数，不允许为空或空值。

5.2 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` 的路径在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：121.0.26.0～121.0.26.255（支付宝主站所使用的 IP 地址是：121.0.26.11，支付宝主站备用 IP 地址：110.75.132.21、110.75.134.101，支付宝通知信息的 IP 主地址：121.0.26.1、121.0.26.2、121.0.27.1、121.0.27.2、110.75.130.1、110.75.130.2、110.75.132.1、110.75.132.2、110.75.136.1、110.75.136.2）

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 return_verify()、notify_verify()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：log_result）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[3.4 返回数据处理 第二步](#)）

5.3 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 5.2 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[5.2 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）