

■ 目录

1.	概述.....	3
1.1.	业务术语.....	3
2.	交互模式.....	3
2.1.	请求/响应交互模式.....	3
2.1.1.	处理流程.....	4
2.2.	主动通知交互模式.....	5
2.2.1.	处理流程.....	5
2.2.2.	通知验证.....	6
3.	安全规范.....	7
3.1.	数字签名.....	7
3.1.1.	签名机制.....	7
3.1.2.	签名方式.....	8
3.2.	外部接入接口.....	8
3.2.1.	业务功能.....	8
3.2.2.	请求参数列表.....	9
3.3.	标准即时到账通知.....	13
3.3.1.	返回参数列表.....	13
3.3.2.	交易状态枚举表.....	16

3.3.3. 退款状态枚举表.....	16
3.3.4. 错误代码列表.....	16
3.4. 通知返回的区别 (return_url 和 notify_url)	18
3.4.1. 交易流程图例.....	19
3.5. 交互实例.....	20
3.5.1. 产生待签名数据.....	20
3.5.2. 计算 sign 值.....	21
3.5.3. 商户系统发起请求.....	21
3.5.4. 支付宝系统返回处理结果.....	22
3.6. 通知返回结果枚举表.....	22
4. 时间戳查询接口 (query_timestamp)	22
4.1. 功能描述.....	22
4.2. 访问形式.....	22
4.3. 请求参数列表.....	23
4.4. 请求输出参数列表.....	23
5. 附录.....	24
5.1. 处理钓鱼网站机制.....	25
5.1.1. 支付宝目前处理钓鱼网站机制.....	25
5.1.2. 防钓鱼不利会遇到的问题.....	25
5.1.3. 防钓鱼中处理机制中的问题.....	25

5.2. 外部商户防钓鱼网站改造功能说明	27
5.3. 外部商户防钓鱼网站改造点	27
5.3.1. 细化商品描述信息	27
5.3.2. 提供请求来源白名单	28
5.3.3. 改造接口调用方式	28
5.3.4. 改造注意点	28

1. 概述

阅读对象：商户系统（商户）的技术开发人员。

版本号	作者	修订章节	修订原因	修订日期	审阅人
1.0		全部	起草		
1.1	技术支持	部分传入参数	增加错误通知地址	2009-8-1	
1.2	技术支持	部分传入参数、增加附录说明	增加防钓鱼属性	2009-8-5	

1.1. 业务术语

名词	说明	备注
支付宝合作 ID (partner)	商户与支付宝合作后在支付宝产生的用户 ID。	登录签约支付宝帐号 —>商家服务 可获得相关信息。
支付宝合作验证码 (key)	商户与支付宝合作后在支付宝产生 32 位加密验证码。	
接口名称	是支付宝针对一些外部接口业务的名字。(service)	
防钓鱼	为了解决目前伪造支付宝交易而作的防范措施	

2. 交互模式

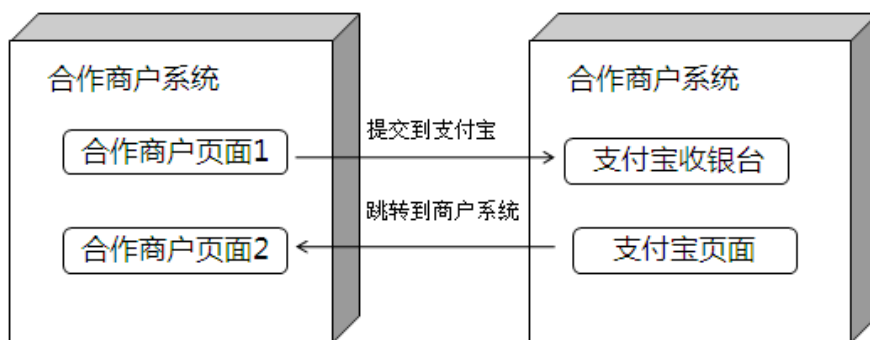
2.1. 请求/响应交互模式

请求/响应模式是最常用的一种交互模式。在这种交互模式下，商户系统向支付宝系统

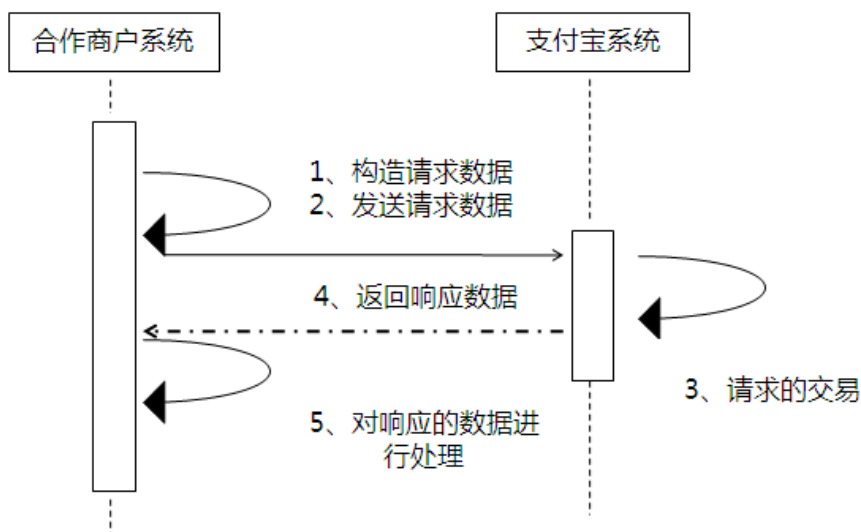
发送请求数据，并同步等待支付宝系统处理完毕之后返回的响应数据。

请求/响应模式根据页面流程，可以分为系统调用和页面跳转，系统调用只需要调用相关接口文件就可以完成相关的业务操作，而页面跳转则需要进入支付宝系统的页面，完成相关操作。

如果买家在跳转到支付宝页面完成相关操作之后，需要支付宝系统将处理结果立即返回给商户网站的下一步操作页面，让用户继续完成整个操作流程，必须传递参数 return_url（即进入商户系统的下一个操作页面）。



2.1.1. 处理流程



接入 URL：

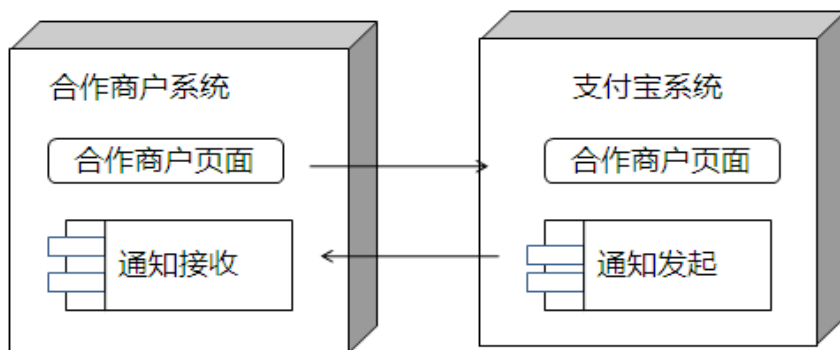
<https://www.alipay.com/cooperate/gateway.do>

2.2. 主动通知交互模式

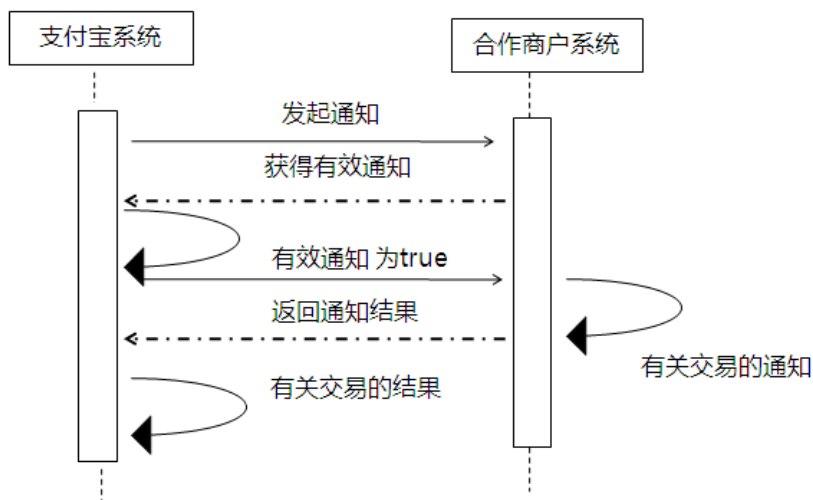
买家从商户网站跳转到支付宝网站,在支付宝网站完成最后操作,买家不用再回到商户网站。支付宝系统会将商户关注的事件采用主动通知的方式提交给商户系统。

这种交互模式如果需要异步返回结果,必须传递 notify_url 参数,以指定通知返回的地址;如果不需要异步返回结果,那么可以不用传递 notify_url 参数。

例如,通过商户系统创建交易,当交易状态(参见:[交易状态枚举表](#))发生改变时,支付宝系统将最新的交易状态以及其它交易有关的信息主动通知给商户系统,达到使双方的系统业务联动的目的。



2.2.1. 处理流程



1. 支付宝系统向商户系统发出通知,即访问商户提供的通知接收 URL(参数 notify_url)。
2. 商户系统接到通知请求, 通过 notify_id 询问支付宝系统这个通知的真实性,通知验证。
3. 支付宝系统判断通知是否是自己发送, 如果是返回 true, 否则返回 false。
4. 商户系统得到支付宝系统的确认后, 对通知进行处理。处理完毕后, 返回结果给支付宝系统, 处理结果的值见附件: [通知返回结果枚举表](#)。
5. 支付宝系统处理商户系统返回的处理结果。

支付宝系统是通过 HTTP/HTTPS 协议的 POST 方法将通知数据发送给商户系统的。商户系统的通知 URL 可以在合作协议中静态配置,则针对该笔交易的所有事件的主动通知, 支付宝系统都会通过该 notify_url 发送给商户系统。

如果支付宝系统发送通知数据不成功, 或者没有收到商户系统处理成功的响应, 则支付宝系统会按照一定的重试策略(1 分钟、3 分钟、5 分钟、10 分钟。。。), 定期重新发送主动通知, 以提高主动通知消息的到达率。但支付宝单系统不保证所有的主动通知消息一定能够送达。

由于存在重新发送主动通知的情况, 因此同样的通知可能会多次发送给商户系统, 而且业务上存在先后的事务的主动通知, 并不一定按照正确的次序发送。商户系统必须能够正确忽略重复的主动通知, 并能正确处理通知次序颠倒的情况。支付宝系统推荐的做法是, 当收到通知并进行处理时, 需要检查本系统内对应业务数据的状态, 以判断该主动通知是否已经处理过, 或者主动通知对应的事件次序是否正确。在对业务数据进行状态检查与处理之前, 要求采用数据锁或者时间戳判断进行并发控制。

2.2.2. 通知验证

从系统健康性角度考虑, 在接收到支付宝系统通知以后, 验证支付宝系统通知的正确性 (合法性) 是非常有必要的。强烈建议商户系统加入此应用。为了保证该接口被合法利用, 商户系统只能查找 1 分钟之内 (目前为 1 分钟, 以后若有调整, 恕不另行通知) 的通知。

✧ 基于 HTTPS 协议的通知验证接口

程序在使用时按照以下要求发起一个 HTTPS 请求，获取该请求的结果即可，所有可能出现的结果见以下的输出参数表，这种验证通知的方式需要网站支持 HTTPS 访问，若网站不支持 https 的访问，可以使用另外一种验证方式：基于 HTTP 协议的通知验证接口。

接入 URL：

```
https://www.alipay.com/cooperate/gateway.do
```

一个完整的验证请求实例：

```
https://www.alipay.com/cooperate/gateway.do?service=notify_verify&partner=1234567890&notify_id=abcdefghijklmnopqrst
```

✧ 基于 HTTP 协议的通知验证接口

程序在使用时按照以下要求发起一个 HTTP 请求，获取该请求的结果即可，所有可能出现的结果见以下的输出参数表。

接入 URL：

```
http://notify.alipay.com/trade/notify_query.do
```

一个完整的验证请求实例：

```
http://notify.alipay.com/trade/notify_query.do?partner=1234567890&notify_id=abcdefghijklmnopqrst
```

通知验证接口输出参数：

输出内容	说明
invalid	传入的参数无效
true	验证通过
false	验证失败

3. 安全规范

3.1. 数字签名

数据传输过程中的数据真实性和完整性，我们需要对数据进行数字签名，在接收签名数据之后进行签名校验。

3.1.1. 签名机制

待签名数据是请求参数按照以下方式组装成的字符串：

- ✧ 请求参数按照参数名字符升序排列，如果有重复参数名，那么重复的参数再按照参数值的字符升序排列。
- ✧ 所有参数（除了 sign 和 sign_type）按照上面的排序用&连接起来，格式是：
p1=v1&p2=v2。

调用某接口需要以下参数：

service=create_direct_pay_by_user,partner=20880063000,email=test@msn.com

那么待签名数据就是：

email=test@msn.com&partner=20880063000&service=create_direct_pay_by_user

注意事项：

- ✧ 没有值的参数无需传递，也无需包含到待签名数据中。
- ✧ 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。
- ✧ 根据 HTTP 协议要求，传递参数的值中如果存在特殊字符（如：&、@等），那么该值需要做 URL Encoding，这样请求接收方才能接收到正确的参数值。这种情况下，待签名数据应该是原生值而不是 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。

3.1.2. 签名方式

按照 sign_type 参数指定的签名算法对待签名数据进行签名。

3.2. 外部接入接口

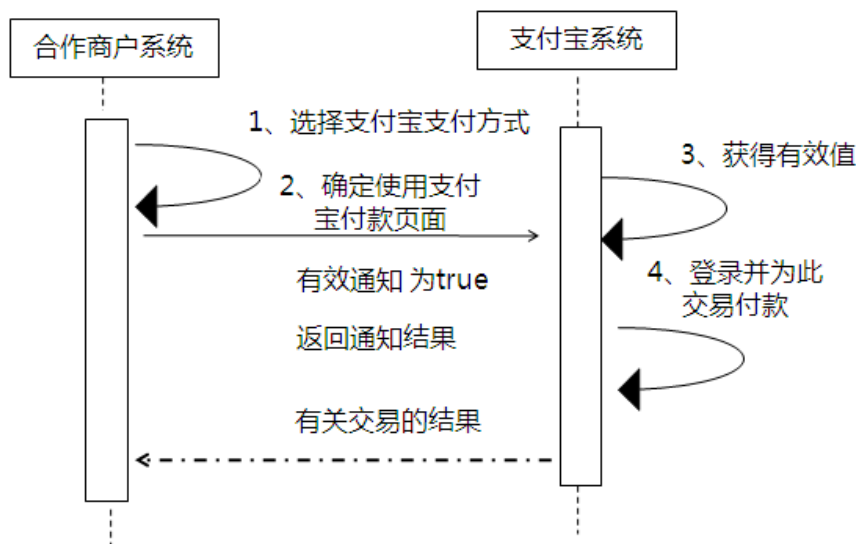
3.2.1. 业务功能

调用此接口，根据用户传过来的参数创建交易，买家再付款。目前该接口的交易全部为即时到账，即只要买家一付款，钱就会转到卖家的支付宝账号上。同时该接口还支持分润，

商家传过来分润的账号和金额，系统会自动打款到该账号上。

交互模式

请求/响应交互模式，页面跳转



3.2.2. 请求参数列表

字段名	变量名	类型	说明	可空
业务参数				
接口名称	service	String	create_direct_pay_by_user	N
合作伙伴 ID	partner	String(16)	合作伙伴在支付宝的用户 ID	N
通知 URL	notify_url	URL (String(190))	针对该交易的交易状态同步通知接收 URL。	N
返回 URL	return_url	URL	结果返回 URL，仅适用于立即返回处理结果的接口。支付宝处理完请求后，立即将处理结果返回给这个 URL。	N
错误通知地址(需要支付宝功能人员开通)	error_notify_url	URL (String(190))	如果在交易过程中出错(如签名错误等)，则通过此 URL 将错误码返回给合作伙伴系统	Y

标准支付宝交易服务接口规范



外部商家技术文档

			统	
签名	sign	String	见 签名机制	N
签名方式	sign_type	String	见 签名方式	N
参数编码字符集	_input_charset	String (默认为 GBK)	<p>合作伙伴系统与支付宝系统之间交互信息时使用的编码字符集。合作伙伴可以通过该参数指定使用何种字符集对传递参数进行编码。同时,支付宝系统也会使用该字符集对返回参数或通知参数进行编码。</p> <p>注：该参数必须在 queryString 中传递，不论使用的是 POST 还是 GET 方式发送请求。如：</p> <p>http://www.alipay.com/cooperate/gateway.do?_input_charset=utf-8</p>	N
商品展示网址	show_url	String(400)	<p>详情的一个连接地址：譬如：http://www.alipay.com</p> <p>点击商品链接后跳转的页面</p>	Y
商品名称	subject	String(256)	商品的标题	N
商品描述	body	String(400)	商品的具体描述	Y
外部交易号	out_trade_no	String(64)	<p>合作伙伴交易号(确保在合作伙伴系统中唯一)，譬如：</p> <p>ali2009041189234</p>	N
商品单价	price	Number(13,2)	price: 单位为 RMB Yuan	N
交易金额	total_fee	Number(13,2)	0.01~100000000.00	
购买数量	quantity	Number(6,0)	<p>total_fee: 单位为 RMB Yuan</p> <p>0.01~100000000.00</p> <p>quantity: 0<quantity < 1000000</p> <p>规则：</p>	

标准支付宝交易服务接口规范



外部商家技术文档

			<p>1.如果是多个商品可以汇总后传入,这个时候商品数量默认为 1</p> <p>2.总价和单价不可以同时出现</p> <p>3.如果使用总价 total_fee , 则数量为 quantity =1.</p> <p>3.如果使用单价 price , 则商品数量 quantity 传递实际商品数量</p>	
支付类型	payment_type	String(4)	1, 目前支持值为 "1"	N
默认支付方式	paymethod	String	bankPay(网 银);cartoon(卡通); directPay(余额)	Y
默认网银	defaultbank	String	见常见 银行表	Y
卖家 Email	seller_email	String(100)	卖家在支付宝的注册 Email 或注册 ID, 两者任何一个。	N
卖家 ID	seller_id	String(16)		
买家 Email	buyer_email	String(100)	买家在支付宝的注册 Email 或注册 ID, 如果买家还没有确定可以为空。	Y
买家 ID	buyer_id	String(30)		
提成类型	royalty_type	String(2)	目前只支持一种类型 卖家给第三方提成 (目前参数=10)	Y
提成信息集	royalty_parameters	String(500)	<p>信息格式下面两种都可：</p> <p>1、收款方 Email_1^金额^备注 付款方 Email^收款方 Email_2^金额^备注</p> <p>2、收款方 Email_1^金额^备注 收款方 Email_2^金额^备注</p> <p>3. (属于多级分润)收款方 Email_1^金额^备注 收款</p>	Y

标准支付宝交易服务接口规范



外部商家技术文档

			方 Email_1^收款方 Email_2^金额^备注 如有多条则用 “ ” 隔开，最多不能超过 10 条，备注最多 1000 个字符，即 500 个汉字，详见“提成数据集”	
超时时间(需要支付宝功能人员开通)	it_b_pay	String(3)	商户开通自定义超时功能才有用(支付宝控制权限)，定义该交易的超时时间，h 表示小时，d 表示天 参数有： (1h, 2h, 3h, 1d, 3d, 7d, 15d) 默认 15d 关闭交易	Y
外部调用 IP	exter_invoke_ip	String(15)	用户在外系统创建交易时，由外部系统记录的用户 IP 地址	N
防钓鱼时间戳	anti_phishing_key	String	通过时间戳查询接口(见文档 4) 获取的加密支付宝系统时间戳，有效时间：30 秒。	N
备注：防钓鱼网站设置需要联系支付宝技术工程师协助配置				

➤ 提成数据集：

1. 收款方 Email_1^金额^备注|付款方 Email^收款方 Email_2^金额^备注

例如：[gwl251@163.com^0.02^分润 1](#)|[gwl251@126.com^gwl061@163.com^0.01^分润 2](#)

2. 收款方 Email_1^金额^备注|收款方 Email_2^金额^备注

例如：[gwl251@163.com^0.02^分润 1](#)|[gwl061@163.com^0.01^分润 2](#)

3.收款方 Email_1^金额^备注|收款方 Email_1^收款方 Email_2^金额^备注

例如：[gwl251@163.com^0.02^分润 1](#)|[gwl251@163.com^gwl061@163.com^0.01^分润 2](#)

注意：只有按照支付宝交易服务接口规范中制定的签名机制，对请求参数进行签名，才能够被支付宝系统接收。

一个完整的支付接入请求实例：

```
https://www.alipay.com/cooperate/gateway.do?seller_email=test%40126.com&discount=0&logistics_fee=0.01&notify_url=http%3A%2F%2Flocalhost%3A8088%2Fjsp_shi_gbk%2Falipay_notify.jsp&payment_type=1&ser
```



标准支付宝交易服务接口规范



外部商家技术文档

```
vice=trade_create_by_buyer&partner=2088002123456782&_input_charset=utf-8&logistics_type=EMS&price=0.01&out_trade_no=20081115162330&subject=AAA20081115162330&logistics_payment=SELLER_PAY&quantity=1&body=%E6%94%AF%E4%BB%98%E5%AE%9D%E6%B5%8B%E8%AF%95&return_url=http%3A%2F%2Flocalhost%3A8088%2Fjsp_shi_gbk%2Falipay_return.jsp&sign=a7ffc1c8ba85df972bb472adc3d199ba&sign_type=MD5
```

正常情况输出：

```
<?xml version="1.0" encoding="gb2312"?>
<alipay>
  <is_success>T</is_success>
  <!-- 处理结果 -->
  <response>
    <trade>
      <trade_no></trade_no>
      <out_trade_no></out_trade_no>
      <subject></subject>
      <trade_status></trade_status>
    </trade>
  </response>
</alipay>
```

异常情况输出：

```
<?xml version="1.0" encoding="gb2312"?>
<alipay>
  <is_success>F</is_success>
  <error>SELLER_NOT_EXIST</error>
</alipay>
```

3.3. 标准即时到账通知

3.3.1. 返回参数列表

注意：只有在跳转页面中输入正确登陆密码支付密码后才能创建交易和通知返回

1. 合作伙伴通过“标准快速付款”接口创建交易时，如果在参数中传递了 *notify_url*，那么当该交易的通知触发条件发生改变时，支付宝会向合作伙伴发送同步通知。

从集成后的系统健壮性考虑，收到支付宝发出的通知后，合作伙伴系统须判断接收到的交易

标准支付宝交易服务接口规范



外部商家技术文档

状态是否与自己系统中的参数对应。如果不判断,存在潜在的风险,合作伙伴自行承担因此而产生的所有损失。

A. 页面跳转返回参数列表:

字段名	变量名	类型	说明	可空
基本信息				
是否成功	is_success	String(1)	表示 该次操作 是否成功	N
签名	sign	String(32)	见 HTTP 参数签名机制	N
签名类型	sign_type	String	见 签名方式	N
业务参数				
接口名称	extinterface	String	使用支付宝的服务接口名称,用此参数可以确定是支付宝的服务	Y
通知时间	notify_time	Timestamp	通知时间(支付宝时间),格式: YYYY-MM-DD hh:mm:ss	Y
通知 ID	notify_id	String	支付宝通知流水号,合作伙伴可以用这个流水号询问支付宝该条通知的合法性	Y
通知类型	notify_type	String	trade_status_sync	Y
支付宝交易号	trade_no	String(16)	该交易在支付宝系统中的交易流水号	Y
支付类型	payment_type	String(4)	值为 1:商品购买	Y
外部交易号	out_trade_no	String(64)	该交易商户提交给支付宝的的交易流水号	Y
商品描述	body	String(400)	商品描述	Y
商品名称	subject	String(256)	商品名称	Y
总价	total_fee	Number(13,2)	总价	Y
买家 Email	buyer_email	String(100)	买家 Email	Y
买家 ID	buyer_id	String(30)	买家 ID	Y
卖家 Email	seller_email	String(100)	卖家 Email	Y
卖家 ID	seller_id	String(30)	卖家 ID	Y
交易状态	trade_status	String	见 交易状态枚举表	Y

跳转 url 样例:

http://10.2.5.100/api/apireceive/returnDisplay.php?body=Hello.&buyer_email=xinxj_xj%40163.



标准支付宝交易服务接口规范



外部商家技术文档

com&buyer_id=2088101000082594&extinterface=create_direct_pay_by_user&ii_success=T¬ify_id=RqPnCoPT3K9%252Fvwbh3l%252BODmZS9o4qChHwPWbaS7UMBjPUnBJlzU42y9A8gQlzU6m3fOhG¬ify_time=2008-10-23+13%3A17%3A39¬ify_type=trade_status_sync&out_trade_no=6402757654153618&payment_type=1&seller_email=chao.chenc1%40alipay.com&seller_id=2088002007018916&subject=%E5%A4%96%E9%83%A8FP&total_fee=10.00&trade_no=2008102303210710&trade_status=TRADE_FINISHED&sign=b1af584504b8e845ebe40b8e0e733729&sign_type=MD5

B. 通知返回处理列表：

字段名	变量名	类型	说明	可空
通知类型	notify_type	String	trade_status_sync	N
通知 ID	notify_id	String	支付宝通知流水号，合作伙伴可以用这个流水号询问支付宝该条通知的合法性	N
通知时间	notify_time	Timestamp	通知时间（支付宝时间），格式：YYYY-MM-DD hh:mm:ss	N
签名	sign	String	见 签名机制	N
签名方式	sign_type	String	见 签名方式	N
支付宝交易号	trade_no	String(16)	该交易在支付宝系统中的交易流水号	Y
外部交易号	out_trade_no	String(64)	该交易在合作伙伴系统的流水号	Y
支付类型	payment_type	String(4)	值为 1：商品购买	Y
商品名称	subject	String(256)		Y
商品描述	body	String(400)		Y
商品单价	price	Number(13,2)	单位为 RMB Yuan 0.01 ~ 100000000.00	Y
购买数量	quantity	Number(6,0)	>0	Y
交易金额	total_fee	Number(13,2)	单位为 RMB Yuan 0.01 ~ 1000000.00	Y
折扣	discount	Number(8,2)	-10000000.00 ~ 10000000.00	Y
交易状态	trade_status	String	见 交易状态枚举表	Y
退款状态	refund_status	String	见 退款状态枚举表	Y
卖家 Email	seller_email	String(100)	卖家 Email	Y
卖家 ID	seller_id	String(30)	卖家 ID	Y
买家 ID	buyer_id	String(30)	买家 Email	Y

标准支付宝交易服务接口规范



外部商家技术文档

买家 Email	buyer_email	String(100)	买家 ID	Y
交易创建时间	gmt_create	Timestamp	该笔交易创建的时间	Y
是否调整总价	is_total_fee_adjust	String(1)	该交易是否调整过价格	Y
交易付款时间	gmt_payment	Timestamp	该交易买家的付款时间	Y
交易关闭时间	gmt_close	Timestamp	交易关闭时间	Y
*退款时间	gmt_refund	Timestamp	卖家退款的时间，退款通知时会发送（支付宝站内退款）	Y
是否使用红包	use_coupon	String(1)	买家是否在交易过程中使用了红包	Y
错误代码（ 需要在传入参数增加错误通知地址 ）	error_code	String(说明接口调用过程中的出错信息	N

3.3.2. 交易状态枚举表

交易状态列表值（变量值： trade_status ）	说明
WAIT_BUYER_PAY	交易创建
TRADE_FINISHED	◆支付成功，标准即时到账成功状态
TRADE_SUCCESS	◆支付成功，高级即时到账返回成功状态
TRADE_CLOSED	交易关闭

3.3.3. 退款状态枚举表

状态代码（变量值： refund_status ）	状态名称
REFUND_SUCCESS	◆退款成功
REFUND_CLOSED	退款关闭

备注：如果在支付宝平台操作退款，交易状态（**trade_status**）会出现 2 中情况：

- 不是全额退款，交易成功状态变成 TRADE_SUCCESS，增加一个退款状态 **refund_status=REFUND_SUCCESS**
- 全额退款，交易成功状态变成 TRADE_CLOSED，增加一个退款状态 **refund_status=REFUND_SUCCESS**。

3.3.4. 错误代码列表



标准支付宝交易服务接口规范



外部商家技术文档

错误代码 (error_code)	说明
ILLEGAL_SIGN	签名验证出错
ILLEGAL_ARGUMENT	参数不正确
HASH_NO_PRIVILEGE	没有权限访问该服务
ILLEGAL_SERVICE	Service 参数不正确
ILLEGAL_PARTNER	商户 ID 不正确
HAS_NO_PUBLICKEY	没有上传公钥
USER_NOT_EXIST	会员不存在
OUT_TRADE_NO_EXIST	外部交易号已经存在
TRADE_NOT_EXIST	交易不存在
ILLEGAL_PAYMENT_TYPE	无效支付类型
BUYER_NOT_EXIST	买家不存在
SELLER_NOT_EXIST	卖家不存在
BUYER_SELLER_EQUAL	买家、卖家是同一帐户
ILLEGAL_SIGN_TYPE	签名类型不正确
COMMISSION_ID_NOT_EXIST	佣金收取帐户不存在
COMMISSION_SELLER_DUPLICATE	收取佣金帐户和卖家是同一帐户
COMMISSION_FEE_OUT_OF_RANGE	佣金金额超出范围
ILLEGAL_LOGISTICS_FORMAT	无效物流格式
TOTAL_FEE_LESSEQUAL_ZERO	交易总金额小于等于 0
TOTAL_FEE_OUT_OF_RANGE	交易总金额超出范围
ILLEGAL_FEE_PARAM	非法交易金额格式 (参考单价、总价、数量三个组合规则)
DONATE_GREATER_THAN_MAX	小额捐赠总金额超出最大值限制
DIRECT_PAY_AMOUNT_OUT_OF_RANGE	快速付款交易总金额超出最大值限制
DIGITAL_FEE_GREATER_THAN_MAX	虚拟物品交易总金额超出最大值限制
SELF_TIMEOUT_NOT_SUPPORT	不支持自定义超时
COMMISSION_NOT_SUPPORT	不支持佣金
VIRTUAL_NOT_SUPPORT	不支持虚拟发货方式

标准支付宝交易服务接口规范



外部商家技术文档

ILLEGAL_CHARSET	字符集不合法
ROYALTY_SELLER_ENABLE_STATUS_FORBID	有提成情况下，卖家状态不正常
ROYALTY_SELLER_NOT_CERTIFY	有提成情况下，卖家未通过认证
ROYALTY_FORAMT_ERROR	提成信息错误，请检查后重新集成
ROYALTY_TYPE_ERROR	提成类型不支持，请检查后重新集成
ROYALTY_PAY_EMAIL_NOT_EXIST	提成付款帐户不存在
ROYALTY_RECEIVE_EMAIL_NOT_EXIST	提成收款帐户不存在
DEFAULT_BANK_MUST_NOT_NULL	默认网银不能为空
LLEGAL_PAYMENT_TYPE	错误的 Payment_type 参数
ILLEGAL_OUTTIME_ARGUMENT	自定义超时参数错误
CANT_CREDIT_PAY	不能使用信用支付
TRADE_NOT_ALLOWED_PAY	交易不允许付款
DUP_OUT_ORDER	浏览器安全空间问题
NAVIGATION_INCOME_OF_ROYALTY_ACCOUNT	在提成情况下的多级分润，二次分润帐户次序问题
ILLEGAL_PARTNER_EXTERFACE	没有开通查询时间戳权限，出现这个问题需要联系支付宝技术工程师
ILLEGAL_REQUEST_REFERER	防钓鱼检查不支持该请求来源，可能是现在请求域名和支付宝后台配置域名不一致
ILLEGAL_ANTI_PHISHING_KEY	防钓鱼检查非法时间戳参数
ANTI_PHISHING_KEY_TIMEOUT	防钓鱼检查时间戳超时
ILLEGAL_EXTER_INVOKE_IP	防钓鱼检查非法外部调用 IP
SESSION_TIMEOUT	session 超时
ILLEGAL_CLIENT_IP	客户端 IP 地址无权访问服务
SYSTEM_ERROR	系统错误

3.4. 通知返回的区别 (return_url 和 notify_url)

return_url : 客户端返回

支付完成后立刻返回到商户网站的客户端上，是可见的，返回机制：以 GET 的方式返回，返回信息包括提交给支付宝的订单信息，可根据这个返回信息做相应的操作显示给客户看。

notify_url : 服务器的返回

服务器的通知返回是由支付宝的服务器发起，以 POST 的方式返回到商户的网站上。

返回信息包括提交给支付宝的订单信息，在返回的文件中，**注意**需要输出 success 做为支付宝通知返回信息成功，若没有这个 success 的输出，那么支付宝的服务器会在交易产生后的 24 小时内返回支付订单信息，返回的时间频率会逐渐减弱（1 分钟、3 分钟、5 分钟、10 分钟、15.....）

1、notify_url 页面中只能做对数据库的业务操作。return_url 的返回页面中也可以做数据库的更新也可以做显示。

2、return_url 和 notify_url 可以都设置，一个做数据显示，一个做更新数据库

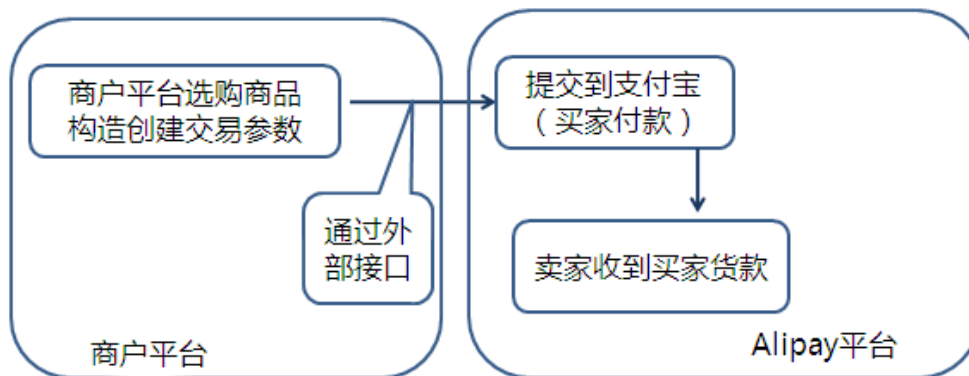
建议，做接口测试的时候，首先获得对应的 Demo 代码后，返回页面写好您的合作 id 和 Key 的信息后进行测试，确定没有问题再添加对数据库更改的操作语句。

获得此交易状态方式([交易状态枚举表](#))：例如 asp 编程语言中：

notify_url 页面 request.Form("trade_status")

return_url 页面 request("trade_status") 其它相对信息也是以此方式获得

3.4.1. 交易流程图例



3.5. 交互实例

买家在商户网站拍下商品后,选择支付宝方式付款,商户系统调用支付宝支付接入接口,根据请求参数的格式,我们给出如下例子:

```
<formname="alipaysubmit"method="post"action="https://www.alipay.com/cooperate/gateway.do?_input_charset=utf-8">
<input type="hidden" name="body" value="a">
<input type="hidden" name="notify_url" value="http://www.alipay.com">
<input type="hidden" name="out_trade_no" value="20081118123456">
<input type="hidden" name="partner" value="2088001958571234">
<input type="hidden" name="payment_type" value="1">
<input type="hidden" name="seller_email" value="test@msn.com ">
<input type="hidden" name="service" value="create_direct_pay_by_user">
<input type="hidden" name="sign" value="275e8da3612b06ab01030f180ca6253d ">
<input type="hidden" name="sign_type" value="MD5">
<input type="hidden" name="subject" value="AAA:20081119125731">
<input type="hidden" name="total_fee" value="0.01">
<input type="hidden" name="show_url" value="www.sina.com.cn ">
<input type="hidden" name="return_url" value="ww.sina.com.cn/alipay_notify.jsp">
<input type="hidden" name="paymethod" value="1">
<input type="hidden" name="defaultbank" value="ICBCB2C "></form>
```

假设使用 MD5 签名算法,并且签名密钥是 abc123,即请求参数 sign_type=MD5。

3.5.1. 产生待签名数据

根据签名机制,我们首先对请求参数进行排序,结果如下:

```
_input_charset=GBK
body=adefaultbank=ICBCB2C
notify_url=http://10.2.17.136:8081/jsp_direct_gbk/alipay_notify.jsp
out_trade_no=20081119125731
partner=2088001958572034
payment_type=1
paymethod=bankPay
return_url=http://localhost:8081/jsp_direct_gbk/alipa
```

```
y_return.jsp  
seller_email=fivipmall@sina.com  
service=create_direct_pay_by_user  
show_url=www.sina.com.cn  
subject=AAA:20081119125731  
total_fee=0.01
```

使用 “&” 符号把参数串联起来，产生待签名数据：

```
_input_charset=GBK&body=a&defaultbank=ICBCB2C&notify_  
url=http://10.2.17.136:8081/jsp_direct_gbk/alipay_not  
ify.jsp&out_trade_no=20081119125731&partner=208800195  
8572034&payment_type=1&paymethod=bankPay&return_url=h  
ttp://localhost:8081/jsp_direct_gbk/alipay_return.jsp  
&seller_email=fivipmall@sina.com&service=create_direc  
t_pay_by_user&show_url=www.sina.com.cn&subject=AAA:20  
081119125731&total_fee=0.01
```

3.5.2. 计算 sign 值

这个实例我们前面已经假设使用 MD5 签名算法，并且给出签名密钥为 abc123，那么在计算 sign 值之前，就需要在待签名数据的后边加上签名密钥，即最终的待签名数据如下：

```
_input_charset=GBK&body=a&defaultbank=ICBCB2C&notify_u  
rl=http://10.2.17.136:8081/jsp_direct_gbk/alipay_notif  
y.jsp&out_trade_no=20081119125731&partner=208800195857  
2034&payment_type=1&paymethod=bankPay&return_url=http:  
//localhost:8081/jsp_direct_gbk/alipay_return.jsp&sell  
er_email=fivipmall@sina.com&service=create_direct_pay_  
by_user&show_url=www.sina.com.cn&subject=AAA:200811191  
25731&total_fee=0.01abc123
```

根据请求参数 sign_type 来判断使用哪种签名算法，这里我们采用 MD5 签名算法，最终计算出来，sign=275e8da3612b06ab01030f180ca6253d

3.5.3. 商户系统发起请求

根据请求/响应交互模式处理流程的支付宝系统服务接入 URL，我们按照以下 URL 发起请求，请求参数的顺序不作要求：

```
https://www.alipay.com/cooperate/gateway.do?seller_email=test%40msn.com&total_fee=0.01&notify_url=http%3A%2F%2F10.2.17.136%3A8081%2Fjsp_direct_gbk%2Falipay_notify.jsp&payment_type=1&defaultbank=ICBCB2C&service=create_direct_pay_by_user&partner=2088001958512345&paymethod=bankPay&_input_charset=GBK&out_trade_no=20081119125731&subject=AAA%3A20081119125731&body=%B0%A2&return_url=http%3A%2F%2Flocalhost%3A8081%2Fjsp_direct_gbk%2Falipay_return.jsp&show_url=www.sina.com.cn&sign=275e8da3612b06ab01030f180ca6253d&sign_type=MD5
```

3.5.4. 支付宝系统返回处理结果

支付宝系统接收到商户系统发起的请求，处理成功后返回的参数中同样包含有参数 sign、sign_type，商户需根据 sign_type 计算 sign 值，最终检验支付宝系统返回的 sign 值。这里要注意的是商户需要对每一个返回参数的值先进行 decode 后再验证 sign。

附录

3.6. 通知返回结果枚举表

返回结果	结果说明
success	处理成功，结束发送
fail	处理失败，重新发送

4. 时间戳查询接口 (query_timestamp)

4.1. 功能描述

时间戳查询接口是用来获取支付宝的系统时间戳，添加在请求 URL 的参数中并进行签名，防止这组参数被恶意重播。

4.2. 访问形式

系统调用

调用端口：<https://mapi.alipay.com/gateway.do>

4.3. 请求参数列表

请注意：

此接口只支持 https 请求

字段名	变量名	类型	说明	可空
业务参数				
接口名称	service	String	query_timestamp	N
合作伙伴ID	partner	String(16)	合作伙伴在支付宝的用户ID	N
签名方式	sign_type	String	见 签名方式 ，默认 MD5，可以支持 DSA、RSA	N
参数编码字符集	_input_charset	String (默认为 GBK)	合作伙伴系统与支付宝系统之间交互信息时使用的编码字符集。合作伙伴可以通过该参数指定使用何种字符集对传递参数进行编码。同时，支付宝系统也会使用该字符集对返回参数或通知参数进行编码。 注：该参数必须在 queryString 中传递，不论使用的是 POST 还是 GET 方式发送请求。如： http://www.alipay.com/cooperate/gateway.do?_input_charset=utf-8	N

请求样例：

```
https://mapi.alipay.com/gateway.do?service=query_timestamp&partner=2088002861972726&_input_charset=GBK&sign=4808032447fe4c56f290f6130d4b3433&sign_type=MD5
```

4.4. 请求输出参数列表

A . xml 返回的参数列表

标准支付宝交易服务接口规范



外部商家技术文档

字段名	变量名	类型	说明	可空	
基础参数					
是否成功	is_success	String(1)	外部接口调用是否成功	N	T/F
接口名称	service	String	query_timestamp	N	
运营 商 Partner	partner	String(16)	运营商签约网游支付的支付宝账号的 UserId	N	
Response 参数					
时间戳密文	encrypt_key	String	支付宝系统时间戳密文	N	KPtAFySgfM KAvQ0sZw= =

注：由于时间戳是根据加密算法计算出的密文，因此该值中包括一些特殊字符，诸如：+（加号）/（斜杠）=（等号）等，这些字符在 URL 传递时需要使用 URL 编码防止字符被解析为其他 URL 链接字符

输出格式

正常输出

```
<?xml version="1.0" encoding="GBK" ?>
- <alipay>
  <is_success>T</is_success>
  <request>
    <param name="_input_charset">GBK</param>
    <param name="partner">2088002861972726</param>
    <param name="service">query_timestamp</param>
    <param name="sign">4808032447fe4c56f290f6130d4b3433</param>
    <param name="sign_type">MD5</param>
  </request>
  <response>
    <timestamp>
      <encrypt_key>KPtAFySgfMKAvQ0sZw==</encrypt_key>
    </timestamp>
  </response>
  <sign>327b0dfa85358944b946d50347556f76</sign>
  <sign_type>MD5</sign_type>
</alipay>
```

发生错误时：

```
<?xml version="1.0" encoding="GBK" ?>
- <alipay>
  <is_success>F</is_success>
  <error>SYSTEM_ERROR</error>
</alipay>
```

5. 附录

5.1. 处理钓鱼网站机制

5.1.1. 支付宝目前处理钓鱼网站机制

- A. 获取用户请求支付的 IP
- B. 验证时间戳的时效性（默认 30 秒）
- C. 设置防钓鱼网站白名单

以上三个条件同时满足时才可以提交支付

5.1.2. 防钓鱼不利会遇到的问题

类似钓鱼造成的资金最终流入商户平台，网银支付的用户会在去银行报案后知道钱最终到了商户帐户，可能会造成一下问题：

1. 客服方面的压力
2. 对于最终资金归属判断的风险
3. 资金退回流程的问题
4. 声誉方面的损失
5. 银行方面的压力

在不进行完整的收银台反钓鱼改造后很容易绕过白名单机制继续被钓鱼

5.1.3. 防钓鱼中处理机制中的问题

对于支付宝三种机制处理任何一种会引起的问题说明如下：

1. 如何跳过白名单设置以及验证防钓鱼网站白名单

举例来说：

ASP 语言：

```
dim http
```

```
set http=server.createObject("MSXML2.XMLHTTP") //MSXML2.serverXMLHTTP 也可以
```

```
Http.open "GET",url,false
```

```
Http.setRequestHeader("Referer", "http://www.05809.cn/")  
Http.send()
```

PHP 语言(前提是装了 curl)

```
$ch = curl_init();  
curl_setopt ($ch, CURLOPT_URL, "http://www.05809.cn/xxx.asp");  
curl_setopt ($ch, CURLOPT_REFERER, "http://www.05809.cn/");  
curl_exec ($ch);  
curl_close ($ch);
```

VB.NET/C#.NET 语言

```
Dim oXMLHttp As MSXML2.XMLHTTP30 = New MSXML2.XMLHTTP30()  
或者  
MSXML2.XMLHTTP30 oXMLHttp = new MSXML2.XMLHTTP30();  
oXMLHttp.open(...  
oXMLHttp.setRequestHeader(...  
oXMLHttp.send(..
```

Java 语言

```
URL url = new URL("http://reg.163.com/CheckUser.jsp");  
URLConnection uc = url.openConnection ();  
uc.setDoOutput(true);  
uc.setUseCaches(false);  
uc.setRequestProperty("Referer", "http://www.damn.japanese.com/");  
HttpURLConnection hc = (HttpURLConnection) uc;  
hc.setRequestMethod("POST");  
OutputStream os = uc.getOutputStream();  
DataOutputStream dos = new DataOutputStream(os);  
dos.writeBytes(content);  
dos.flush();  
dos.close();
```

以上是各种常用语言跳过白名单后的手法。通过程序可以了解如何绕过进行欺诈。

2. 只获取用户请求支付 IP

验证用户请求支付 IP，可能会出现在用户在某公司内网或“钓鱼者”使用代理服务器和用户代理服务器一样，那么这个出口 IP 是一致的，也不能有效的防止。那么支付宝需要验证 IP，如果出现两边获取到的出口 IP 不同，支付宝会弹出提示对话框提示用户，从而减少被骗的可能性

3. 只验证时间戳

如果只验证提交的时效性，可以做到一定的防范，但是如果不限 IP 和白名单的前提下会使这个安全性大大降低。支付宝目前验证时间戳的时效性是 30 秒 (s)，也就是说从商户网站提交到支付宝服务器支付页面的有效时间。在这个时间内提交都是允许支付，如果超过这个时间此订单不允许支付。

5.2. 外部商户防钓鱼网站改造功能说明

商户开通防钓鱼网站检测开关，需要对原有的即时到账接口进行改造。

- 防钓鱼检查开关打开后，则商家的请求输入参数中必须包含**外部调用 IP (exter_invoke_ip)**和**防钓鱼时间戳 (anti_phishing_key)**两个字段，**否则无法通过防钓鱼检查，导致无法使用支付宝的标准网络快速付款服务。**
- 白名单是一组用分号分隔的商户网站一级域名，比如：taobao.com;5173.com 等。这里使用的是一级域名来扩大请求来源的匹配程度，比如 taobao.com，可以匹配 www.taobao.com，也可以匹配 mall.taobao.com 等
- 外部调用 IP 是指用户在商户网站上点击“去支付宝付款”时，由商户系统获取的用户 IP 地址。该 IP 会用来与支付宝系统获取到用户的 IP 地址进行比较，防止 URL 请求中的参数被钓鱼网站利用，对其他用户实施钓鱼攻击。
- 防钓鱼时间戳用来防止 URL 重播，只有在时间戳生成 30 秒内有效，超时后需要重新生成来使用。因此，时间戳的生产必须放在提交支付宝操作的同时，避免用户在商户网站停留时间过长而导致时间戳超时。

5.3. 外部商户防钓鱼网站改造点

5.3.1. 细化商品描述信息

由于商品描述信息参数 (body) 主要用来描述商品的属性，而且当商户传来的外部调用 IP 与支付宝系统获取的 IP 不相同时，会用来提醒用户该交易有钓鱼风险，如下图：



诸如此类 body=支付宝直付的参数在警告信息中无法清晰地显示当前商品信息。因此，除了商品标题外，商户必须细化商品描述信息参数 (body)，从而使用户能够清楚地了解自己通过一个 URL 跳转到支付宝时购买的究竟是什么商品

5.3.2. 提供请求来源白名单

商户需要提供自己网站的一级域名，以及使用商户在支付宝的 PartnerID 的合作网站的一级域名，以防自己的正常业务被防钓鱼检查错误的拦截掉

5.3.3. 改造接口调用方式

使用防钓鱼检查后，生成的接口参数不应当存放在最后确认页面的隐藏域内，而应当在用户最终确认跳转到支付宝收银台前，调用时间戳查询接口获取支付宝的系统时间戳，并将商户系统中获取到的客户端 IP 一并追加到接口参数中，再进行签名，最终发起页面跳转到支付宝收银台。

在这一连串的处理中，尤其应当注意对时间戳的处理。由于时间戳是一段密文，其中可能会包含诸如加号等特殊字符。**在签名的时候需要对包含这些特殊字符的密文进行签名，而跳转前必须对这些特殊字符进行 URL 编码。如果未经 URL 编码，在 URL 传递中会将加号作为空格解析，产生签名错误的结果。因此，时间戳的处理必须小心谨慎！**

5.3.4. 改造注意点

- 对加密时间戳的处理必须谨慎。签名时必须对获取到的时间戳直接进行签名，而在跳转前必须对时间戳进行 URL 编码，否则会出现签名错误。也就是说，跳转到支付宝的 URL 中，anti_phishing_key 字段中不能出现+ (加号) 或者 (空格)，最多只能出现%2B (加号的 URL 编码)。
- 系统改造中如果使用了 Ajax 或正则式时，应防止¬ify_url 被转意成?ify_url。出现这种情况的原因一般是页面解析时把¬ 作为 HTML 编码的¬来解析了。因此，在最终生成的参数中必须留意¬ify_url 这个参数名是否正确。
- 对于 IP 地址的获取，我们采用较简单的 IP 获取方式，没有去判断代理的情况。因此，商户也必须采用同样的方式来获取，避免 IP 不一致。以下提供几个例子：

Java 环境：javax.servlet.http.HttpServletRequest.getRemoteAddr();

标准支付宝交易服务接口规范



外部商家技术文档

ASP 环境：Request.ServerVariables("REMOTE_ADDR")

ASP.net 环境：System.Web.HttpContext.Current.Request.UserHostAddress;

PHP 环境：\$_SERVER['REMOTE_ADDR'];