



接入与使用规则

标准双接口(trade_create_by_buyer)

附录文档

版本号：1.1

目 录

1 文档说明	3
1.1 文档说明	3
1.2 业务术语	3
2 责任归属	4
3 技术接入规则	4
4 接口使用规则	12
5 交易超时列表	14
6 测试流程规则	15

1 文档说明

1.1 文档说明

本文档是《标准双接口(trade_create_by_buyer)》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。

1.2 业务术语

表1-1 业务术语

术语	解释
标准双接口	支持两种支付方式：即时到账付款、担保交易付款。但不等于开通了即时到账接口和担保交易接口。这两种支付方式由买家付款时选择。
CTU	支付宝风险稽查系统。
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。
快捷登录	快捷登录产品主要有以下功能： <ul style="list-style-type: none">• 用户在商户的网站上，可以使用支付宝快捷登录，并共享支付宝的收货地址等物流信息给商户；• 如果用户在商户网站上使用了支付宝快捷登录，那么在支付宝支付时，不需要再次在支付宝登录。
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（标准双接口(trade_create_by_buyer).pdf）、代码示例、本文档（标准双接口(trade_create_by_buyer)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 技术接入规则

表3-1 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
	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。
	如果商户签约的是平台商协议，那么本接口支持收款账号（请求参数 seller_email）与签约账号（请求参数 partner）不是同一个支付宝账号。否则收款账号与签约账号必须相同（即 seller_email 与 partner 为同一个支付宝账号）。	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。
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

类型	细则	原因
	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 <code>sign</code> 、 <code>sign_type</code> 。	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 <code>_input_charset</code> 保持一致; 如果传递了 <code>_input_charset</code> 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 <code>url encoding</code> 之后的值, 例如: 调用某接口需要对请求参数 <code>email</code> 进行数字签名, 那么待签名数据应该是 <code>email=test@msn.com</code> , 而不是 <code>email=test%40msn.com</code> 。	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
	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
	推荐设置 <code>notify_url</code> 。 <code>notify_url</code> 的作用是当交易的交易状态或退款状态发生变更时, 支付宝系统会根据该路径发送服务器异步通知。	只有设置了请求参数 <code>notify_url</code> 或支付宝系统中绑定了 <code>notify_url</code> , 支付宝系统才会发送服务器异步通知。
	<code>return_url</code> 的作用是用户付款完成以后, 当前付款完成界面会自动跳转至该路径去。	只有设置了请求参数 <code>return_url</code> 或支付宝系统中绑定了 <code>return_url</code> , 付款完成时的支付宝界面才会自动跳转至该路径。
	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 <code>http://localhost</code> 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。

类型	细则	原因
	请求参数 <code>payment_type</code> 必须设置为 1	1 代表商品购买
	请求参数 <code>subject</code> 、 <code>body</code> 的值中不要使用敏感词	避免接口无法正常使用
	必须设置 <code>price</code> （商品单价）、 <code>quantity</code> （购买数量），不能使用 <code>total_fee</code> （交易金额）。	防止出现支付金额错误
	<code>price</code> （商品单价）最小为 0.01 元（RMB）	设置为 0 或负数时报错，错误码是：ILLEGAL_ARGUMENT。
	<ul style="list-style-type: none"> 物流信息最多设置三组。每一组物流信息分别是： <ol style="list-style-type: none"> 第一组：<code>logistics_type</code>、<code>logistics_fee</code>、<code>logistics_payment</code> 第二组：<code>logistics_type_1</code>、<code>logistics_fee_1</code>、<code>logistics_payment_1</code> 第三组：<code>logistics_type_2</code>、<code>logistics_fee_2</code>、<code>logistics_payment_2</code>。 必须至少设置第一组物流信息。 	<ul style="list-style-type: none"> 每一组物流信息包含三个物流信息请求参数，分别是物流类型、物流费用、物流支付方式； 缺少第一组三个参数中的任何一个都会导致接口报错，如错误码 ILLEGAL_LOGISTICS_FORMAT。
	若设置了某组物流信息，则该组的三个物流参数都需要按照格式要求填写完整。	避免组物流信息不显示甚至报错，如错误码 ILLEGAL_LOGISTICS_FORMAT。
	必须有第一组物流信息才能有第二组或第三组，并且建议每组的物流类型（ <code>logistics_type</code> 、 <code>logistics_type_1</code> 、 <code>logistics_type_2</code> ）各不相同。	避免报错，如错误码 ILLEGAL_LOGISTICS_FORMAT。如果物流类型有相同，那么会使相同类型的物流信息混乱，使用户产生困扰，影响交易。
	物流费用（ <code>logistics_fee</code> 、 <code>logistics_fee_1</code> 、 <code>logistics_fee_2</code> ）可以设置为 0。	支持物流费用免费，不能为负数。
	如果设置请求参数 <code>discount</code> （折扣），建议将其赋予负数值。	<code>discount</code> （折扣）支持正负数金额
	最后应付款金额计算公式是： $\text{price} \times \text{quantity} + \text{discount} + \text{logistics_fee}$	该金额是显示在支付宝收银台时的应付款金额
	如果设置了收货人信息（请求参数为 <code>receive_name</code> 、 <code>receive_address</code> 、 <code>receive_zip</code> 、 <code>receive_phone</code> 、 <code>receive_mobile</code> ），那么用户付款时，支付宝付款流程中会减少让用户填写收货信息这一步骤。	支付宝支持用户在商户网站下单时的收货信息同步到对应的支付宝交易中

类型	细则	原因
	如果要使用收货人信息，需至少保证请求参数 <code>receive_name</code> 、 <code>receive_address</code> 不为空或空值。	如果缺少这两个请求参数中的一个或两个，则支付宝付款流程中依然会显示让用户填写收货信息这一界面。
	如果要设置收货人信息，则须严格按照姓名、地址、邮编、电话、手机的格式填写。	避免报错，错误码如： <code>ILLEGAL_ARGUMENT</code> 。
	<code>seller_id</code> （卖家支付宝用户号）、 <code>seller_account_name</code> （卖家别名支付宝账号）、 <code>seller_email</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 > seller_account_name > seller_email</code>	数据完整一致，避免出现卖家信息错误。
	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_email</code> 不能相同、 <code>buyer_id</code> 与 <code>seller_id</code> 不能相同、 <code>buyer_account</code> 与 <code>seller_account</code> 不能相同。	避免报错，如错误码： <code>BUYER_SELLER_EQUAL</code> 。
	只有开通了自定义超时功能，设置的请求参数 <code>t_s_send_1</code> 、 <code>t_s_send_2</code> 、 <code>t_b_rec_post</code> 、 <code>it_b_pay</code> 才有效。	在未开通自定义超时功能的情况下设置了这些参数不会产生效果。并且会报错，错误码如： <code>SELF_TIMEOUT_NOT_SUPPORTED</code> 。
	只有开通了快捷登录，才能使用请求参数 <code>token</code> （授权令牌码），且必须设置 <code>token</code> 。	减少用户付款时重复登录支付宝
	<ul style="list-style-type: none"> 如果是 <code>etao</code> 接入的商户，那么必须设置请求参数 <code>error_notify_url</code> 和 <code>item_orders_info</code>； 如果不是 <code>etao</code> 接入商户，不要设置 <code>item_orders_info</code>。 	<ul style="list-style-type: none"> <code>error_notify_url</code> 是使用接口时请求支付宝报错时商户能收到报错信息的地址； <code>item_orders_info</code> 是买家通过 <code>etao</code> 购买的商品的详细清单。该数据能同步到 <code>etao</code> 的订单列表中。
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： https://mapi.alipay.com/gateway.do 。	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

类型	细则	原因
	<p>发送给支付宝的请求，如果使用 form 表单传输，需要按照以下要求编写：</p> <ul style="list-style-type: none"> • action 的值必须为 “https://mapi.alipay.com/gateway.do?_input_charset=该值”， 如：https://mapi.alipay.com/gateway.do?_input_charset=utf-8。 <p>不允许写成完整的请求链接地址，即禁止 https://mapi.alipay.com/gateway.do?后带有所有 要请求给支付宝的请求参数数据；</p> • <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。 	<ul style="list-style-type: none"> • 避免请求支付宝时报错，错误码为 ILLEGAL_SIGN； • 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。
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 item_orders_info 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	<p>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：</p> <p>https://mapi.alipay.com/gateway.do?partner=合作者身份ID&notify_id=通知ID的值</p>	验证是否是支付宝发来的请求
	<p>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。</p>	验证返回的签名

类型	细则	原因
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。 📖 说明: 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。
	需根据不同的交易状态或退款状态做对应的业务逻辑处理	防止商户的订单信息和支付宝的交易信息不同步
	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
交易及退款状态	担保交易方式的交易状态(TRADE_STATUS)的顺序是: WAIT_BUYER_PAY (等待买家付款) → WAIT_SELLER_SEND_GOODS (卖家已付款,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)	交易的通知顺序是以支付宝系统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
	即时到账方式的交易状态(TRADE_STATUS)的顺序是: WAIT_BUYER_PAY (等待买家付款) → TRADE_FINISHED (交易完成)	交易的通知顺序是以支付宝系统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

类型	细则	原因
	<p>退款状态（REFUND_STATUS）的顺序是：</p> <ul style="list-style-type: none"> 已收到货的情况 <p>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）</p> <ul style="list-style-type: none"> 未收到货的情况 <p>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）</p>	<p>交易的通知顺序是以支付宝系统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</p>
	<ul style="list-style-type: none"> 卖家未发货而退款成功，交易状态会变为 TRADE_CLOSED； 卖家已发货而退款成功后，交易状态会变为 TRADE_FINISHED。 	<p>非正常完成交易则交易状态会变成 TRADE_CLOSED。</p>
	<p>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。</p>	<p>当发生退款时，若交易退款状态不为 REFUND_CLOSED 或 REFUND_SUCCESS 时，买家可再申请退款。</p>
	<p>如果交易未发生退款且设置了 notify_url，那么服务器异步通知时不会返回退款信息。</p>	<p>未发生退款，通知里的退款信息为空。</p>
	<p>如果设置了 notify_url 且在默认配置下，那么支付宝会根据交易状态及退款状态来发送服务器异步通知。</p>	<p>支付宝的默认状态通知配置下，所有的交易状态和退款状态的变更均会发送服务器异步通知。</p>
	<p>如果设置了 return_url，那么该页面只会接收到买家付款成功的交易状态，即 WAIT_SELLER_SEND_GOODS。</p>	<p>return_url（页面跳转同步通知）是买家付款完成以后才会触发的</p>
	<p>判断买家支付方式是即时到账还是担保交易——</p> <p>前提：须设置 notify_url（服务器异步通知参数）和记录第一个交易状态 WAIT_BUYER_PAY（等待买家付款）。</p> <ul style="list-style-type: none"> 当买家付款完成时，如果收到的服务器异步通知的交易状态是 WAIT_SELLER_SEND_GOODS（买家已付款，等待卖家发货），那么买家使用的支付方式是担保交易模式； 当买家付款完成时，如果收到的服务器异步通知的交易状态是 TRADE_FINISHED（交易完成），那么买家使用的支付方式是即时到账模式。 	<p>标准双接口支持两种支付方式，供买家选择，而不是卖家控制。</p>

类型	细则	原因
接入环境	不能把接口嵌入 <code>iframe</code> 框架中	避免接口无法正常使用
错误码处理	常见错误码问题，请参见： http://help.alipay.com/support/234878-235357/0-235357.htm?sh=Y&info_type=9	该链接中记载了常见错误码问题解决方案
	如果遇到错误码“ <code>ILLEGAL_PAYMENT_TYPE</code> ”，表明 <code>payment_type</code> （支付类型）设置错误，请检查 <code>payment_type</code> 是否正确。	请按照接口文档中的收款类型表填写该支付类型。
	如果遇到错误码“ <code>LOGISTICS_CHOOSE_ERROR</code> ”，表明当前获取到的物流信息与商户接口请求时传递的物流信息不一致。	数据不一致则报错，避免用户完成该笔有风险的交易。
	如果遇到错误码“ <code>TRADE_DATA_MATCH_ERROR</code> ”，表明当前获取到的交易信息与商户接口请求时传递的交易信息不一致。	数据不一致则报错，避免用户完成该笔有风险的交易。
	如果遇到错误码“ <code>ILLEGAL_FEE_PARAM</code> ”，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"> 总价和单价不可以同时出现； 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； 如果存在单价，则商品数量必须存在。
	如果交易状态是“ <code>TRADE_CLOSED</code> ”，可能是交易超时。	交易超过时限自动关闭
	如果交易状态出现“ <code>TRADE_PENDING</code> ”，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

4 接口使用规则

表4-1 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、交易金额等）在多次调用中保持一致，且交易尚未完成支付，即交易状态是 WAIT_BUYER_PAY。	防止重复付款以及交易信息被篡改 说明： 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。
	担保交易支持在交易创建后（即交易状态是 WAIT_BUYER_PAY），卖家修改价格。	该操作需要卖家（收款人）登录支付宝网站，在交易管理中执行。
	担保交易支持卖家在交易创建后（即交易状态是 WAIT_BUYER_PAY），卖家关闭交易。	该操作需要卖家（收款人）登录支付宝网站，在交易管理中执行。或者，使用交易关闭接口。
	担保交易支持买家付款完成后（即交易状态是 WAIT_SELLER_SEND_GOODS），卖家执行发货操作。	该操作需要卖家（收款人）登录支付宝网站，在交易管理中执行。或者使用确认发货接口。
	不论卖家使用确认发货接口还是登录支付宝网站在交易管理中执行发货操作，发货时卖家填写的物流信息，必须与接口请求时设置的物流信息保持一致。	如果接口请求时设置的物流信息与卖家发货时填写的物流信息不一致，以卖家发货时填写的物流信息为准。
	担保交易支持卖家发货后（即交易状态是 WAIT_BUYER_CONFIRM_GOODS），买家确认收货。	该操作需要买家（付款人）登录支付宝网站，在交易管理中执行。
	卖家可延迟收货时间，每次可延迟 10 天，一天可以多次。	该操作需要卖家（收款人）登录支付宝网站，在交易管理中执行。 该延迟需要买卖双方达成一致，由买家发起。避免支付宝系统自动帮买家确认收货，造成买家没有收到货，交易却完成了。
	担保交易支持退款流程	该系列操作，需要买卖双方登录支付宝网站，在交易管理中执行。
	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。
	本接口支持的众多支付通道中，储蓄卡支付和信用卡支付一旦进入网银系统页面，支付宝将无法控制订单的支付效率问题。	便于做支付渠道区分

类型	规范点	原因
	<p>如果想要买家不安装数字证书也能进行余额支付，商户需要申请开通非证书余额支付功能；否则，买家必须申请安装数字证书才能用支付宝余额支付。</p> <p> 说明：</p> <p>该种情况只有在买家从来没有申请过数字证书的情况下才有效。</p>	保护买家账户安全
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>	避免商户的订单业务处理重复
	<p>支付宝支付手续费扣取模式分为两种：</p> <ul style="list-style-type: none"> ● 实时扣费模式：一般都是收款账户扣费，或采用合同指定固定扣款支付宝账户扣费； ● 月结手续费模式：实时交易不扣手续费，月底账单统一结算。 	实时扣费时，在扣费支付宝账号的账务明细中可马上查到扣费记录；但月结模式下，则不会查到。
	<p>建议商户不要使用类似 js 的 <code>window.open</code>、<code>location.replace</code> 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 <code>form</code> 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 <code>seller_email</code> 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 <code>seller_id</code> 做为商户收款账户。</p>	<code>seller_id</code> 值是支付宝分配给商户的唯一身份 ID，不会改变，使用 <code>seller_id</code> 作为收款账户可以避免 <code>seller_email</code> 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 <code>seller_id</code> 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；借记卡付款，则退回借记卡；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。

5 交易超时列表

表5-1 交易超时列表

交易超时期限	计算起始时间	默认值
支付宝交易正向超时		
买家逾期不付款，自动关闭交易（支付宝交易）。	卖家最后修改交易时间	7 天
交易创建后，允许卖家关闭交易。	交易创建时间	0
买家去网银支付，但一定时间内未支付结束，允许卖家继续修改交易。	卖家最后修改交易时间	15 分钟
卖家逾期不发货，允许买家退款。	收到买家付款时间	1 天
卖家逾期不发货，建议买家退款。	收到买家付款时间	15 天
买家逾期不确认收货，自动完成交易（EMS 物流方式）。	卖家发货时间	10 天
买家逾期不确认收货，自动完成交易（平邮物流方式）。	卖家发货时间	30 天
买家逾期不确认收货，自动完成交易（推荐快递物流方式）。	卖家发货时间	10 天
买家逾期不确认收货，自动完成交易（其它快递物流方式）。	卖家发货时间	10 天
买家逾期不确认收货，自动完成交易（无需物流方式）。	卖家发货时间	10 天
支付宝交易凭证超时		
卖家已发货，买家已申请退款并声明未收到货，卖家逾期未上传凭证，自动完成退款。	退款申请时间	7 天
卖家已发货，买家已申请退款并声明未收到货，卖家逾期未二次上传凭证，自动完成退款。	凭证超时启动时间	7 天
“卖家已发货，买家未收到货”或者“买家已退货，卖家未收到货”，凭证已经上传一次或者二次上传，规定天数后凭证状态更改为等待初审。	凭证超时启动时间	7 天
买家已退货，卖家声明未收到退货，买家逾期未上传凭证，自动关闭退款。	退款修改时间	7 天
买家已退货，卖家已声明未收到退货，买家逾期未二次上传凭证，自动关闭退款。	退款修改时间	7 天

交易超时期限	计算起始时间	默认值
支付宝交易退款超时		
买家申请退款并声明已收到货；卖家逾期不响应退款申请，自动达成退款协议。	退款最后修改时间	15 天
卖家未发货，买家申请退款，卖家逾期不响应退款申请，自动达成退款协议。	退款最后修改时间	5 天
买家申请退款并声明已收到货，卖家不同意退款协议；买家逾期未修改退款协议，自动关闭退款。	退款最后修改时间	15 天
买家申请退款并退货，卖家同意退款，买家逾期不退货，自动关闭退款。	退款最后修改时间	15 天
买家申请退款并退货，卖家同意退款，买家已经退货；卖家逾期未确认收货，自动完成退款。	买家发货时间	<ul style="list-style-type: none"> ● EMS (EMS) : 3 天 ● 快递 (EXPRESS) : 10 天 ● 平邮 (POST) : 30 天

6 测试流程规则

表6-1 测试流程规则

步骤	调试内容	备注
Windows 环境（linux 环境）服务器配置要求： 推荐配置 Java 1.6 以上运行环境，符合 servlet2.4 规范的 WEB 容器。		
第一步： 在本机单独对这个接口进行调试。	<ul style="list-style-type: none"> ● 支付 ● 返回 	仅仅把接口配置好，不要放在商户的网站项目中。
第二步： 在服务器上单独对这个接口进行调试	<ul style="list-style-type: none"> ● 支付 ● 返回 ● 通知 	本机调试没有问题后，再放入服务器中调试。
第三步： 接口融合到网站项目中	无	把调试好的接口与商户网站项目的业务流程进行衔接和融合。

步骤	调试内容	备注
第四步: 在本机对融合后的网站项目进行调试	<ul style="list-style-type: none">● 整个业务操作流程● 支付● 返回● 业务后续的执行	在本机调试衔接到网站项目后的接口。
第五步: 在服务器对融合后的网站项目进行调试	<ul style="list-style-type: none">● 整个业务操作流程● 支付● 返回● 通知● 业务后续的执行	本机调试没有问题后，再放入服务器中调试。