



接入与使用规则

即时到账交易接口、纯网关接口、大额信用卡
接口、快捷支付前置接口、快捷支付网关接口、
信用卡分期支付接口

(create_direct_pay_by_user)

附录文档

版本号：1.4

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 11

5 测试流程规则 16

1 文档说明

1.1 文档说明

本文档是《即时到账交易接口、纯网关接口、大额信用卡接口、快捷支付前置接口、快捷支付网关接口、信用卡分期支付接口(create_direct_pay_by_user)》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
CTU	支付宝风险稽查系统。
定向支付	用户事先指定支付金额的收款方，对应的交易只能将金额转入指定的收款方账户中。
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。
分润	分润是指将交易金额中的一部分转账给其它账户。
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（即时到账交易接口、纯网关接口、大额信用卡接口、快捷支付前置接口、快捷支付网关接口、信用卡分期支付接口(create_direct_pay_by_user).pdf）、代码示例、本文档（即时到账交易接口、纯网关接口、大额信用卡接口、快捷支付前置接口、快捷支付网关接口、信用卡分期支付接口(create_direct_pay_by_user)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。
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
	当设置 <code>paymethod</code> (默认支付方式)为 <code>directPay</code> (余额支付)时,请求参数 <code>defaultbank</code> (默认网银)不要设置或不要传递。	避免该交易按其他支付方式执行
	只有开通了自定义超时功能,才能使用请求参数 <code>it_b_pay</code> (超时时间)。	避免接口报错误码 <code>SELF_TIMEOUT_NOT_SUPPORT</code>
	只有开通了防钓鱼功能且开通了防钓鱼时间戳,才能使用请求参数 <code>anti_phishing_key</code> (防钓鱼时间戳)。	避免接口无法正常使用
	只有开通了防钓鱼功能且开通了 IP 地址检查,才能使用请求参数 <code>exter_invoke_ip</code> (客户端 IP)。	避免接口无法正常使用
	只有开通了网银支付时是否做 CTU 校验,才能使用请求参数 <code>need_ctu_check</code> (网银支付时是否做 CTU 校验)。	避免接口无法正常使用
	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

类型	细则	原因
	请求参数 subject 、 body 的值，以及 extend_param 、 item_orders_info 、 royalty_parameters 的备注表述信息中不要使用敏感词。	避免接口无法正常使用
	请求参数 notify_url 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 return_url 的设置不能是 http://localhost/ 这类地址，必须是服务器 ip 地址或者域名方式。（例如:127.0.0.1）	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 return_url 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
	royalty_parameters （分润账号集）、 extend_param （公用业务扩展参数）参数中的备注描述信息中不能出现用作字段分割符的 “^”、“ ” 特殊字符。	避免出现数据格式错误，导致分润失败。
	当使用了分润功能时，在设置请求参数 royalty_parameters （分润账号集）的值时，分润的总金额不能超过付款总金额减去支付宝手续费所余下的金额。	避免分润失败，而导致接口无法正常使用。
	当使用了分润功能时，在设置请求参数 royalty_parameters （分润账号集）的值时，分润的收款账户必须是有效的收款账号。	避免分润失败，而导致接口无法正常使用。
	seller_id （卖家支付宝用户号）、 seller_account_name （卖家别名支付宝账号）、 seller_email （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： seller_id > seller_account_name > seller_email	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 buyer_email 等），那么买家支付宝账号不能与卖家支付宝账号相同，即： buyer_email 与 seller_email 不能相同、 buyer_id 与 seller_id 不能相同、 buyer_account 与 seller_account 不能相同）。	避免报错，如错误码： BUYER_SELLER_EQUAL 。
	price （商品单价）、 quantity （购买数量）会替换 total_fee （交易金额）。即 total_fee 不能与 price 、 quantity 同时存在；存在 price 、 quantity ，就不能存在 total_fee 。	防止出现支付金额错误
	在给请求参数 defaultbank 、 paymethod 赋值时，需注意区分大小写。	否则会引起银行直连调用失败

类型	细则	原因
	只有开通了纯网关（即网银直连），且 paymethod 赋值为 bankPay 时，才有纯网关的效果。如果没有开通，则 paymethod 禁止赋值为 bankPay 。	如果没有开通该功能，而又设置了 paymethod 为 bankPay ，那么会出现以下两种情况： <ul style="list-style-type: none"> 从来没有开通过，报没有开通该产品的提示错误； 曾经有开通过，交易费率按照纯网关的测试费率 3%收取。
	只有开通了大额信用卡功能，且 <ul style="list-style-type: none"> paymethod=CREDITCARD credit_card_pay=Y credit_card_default_display=Y 如此设置以上 3 个参数值时，才有大额信用卡的效果。	如果没开通该功能，即使 paymethod 赋值为 CREDITCARD 也无效，甚至会报没有开通该产品的提示错误。
	只有开通了信用支付，且 paymethod 赋值为 creditPay 时，才有信用支付的效果。	如果没开通该功能，即使 paymethod 赋值为 creditPay 也无效，甚至会报没有开通该产品的提示错误。
	只有开通了信用卡分期，且 paymethod 赋值为 CCIP 时，才有信用卡分期的效果。	如果没开通该功能，即使 paymethod 赋值为 CCIP 也无效，甚至会报没有开通该产品的提示错误。
	只有开通了快捷支付前置，且 paymethod 赋值为 motoPay 时，才有快捷支付前置的效果。	如果没开通该功能，即使 paymethod 赋值为 motoPay 也无效，甚至会报没有开通该产品的提示错误。
	只有开通了快捷支付网关，且 paymethod 赋值为以下 3 个值之一： <ul style="list-style-type: none"> expressGatewayDebit（快捷支付网关借记卡单通道） expressGatewayCredit（快捷支付网关信用卡单通道） expressGateway（快捷支付网关双通道） 且 default_login 赋值为 Y ，以上 2 个参数必须都设置，才有快捷支付网关的效果。	如果没开通该功能，即使 paymethod 赋值为对应的值也无效，甚至会报没有开通该产品的提示错误。

类型	细则	原因
	<ul style="list-style-type: none"> 如果是 etao 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 etao 接入商户，<code>item_orders_info</code> 不要设置。 	<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 etao 购买的商品的详细清单。该数据能同步到 etao 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
	传递请求出错时的通知页面路径 <code>error_notify_url</code> (需要联系技术支持申请开通该参数权限)	方便商户定位接口报错
	<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"> <code>action</code> 的值必须为 “https://mapi.alipay.com/gateway.do?_input_charset=utf-8”， 如：https://mapi.alipay.com/gateway.do?_input_charset=utf-8。 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; <code><form></code> 与 <code></form></code> 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 <code><input type="hidden" name="参数名" value="参数值" /></code>; 在众多请求参数中，请求参数 <code>_input_charset</code> (编码格式) 必须存在于 form 表单中，即 form 表单中必须含有 <code><input type="hidden" name="_input_charset" value="参数值"></code>; <code><form></code> 与 <code></form></code> 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 <code>method</code> 属性，可自行选择 <code>get</code>、<code>post</code> 两种。 	<ul style="list-style-type: none"> 避免请求支付宝时报错，错误码为 <code>ILLEGAL_SIGN</code>; 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 <code>get</code> 、 <code>post</code> 或者 <code>httpClient</code> 方式提交。	避免接口无法正常使用

类型	细则	原因
	设置了请求参数 <code>item_orders_info</code> 的情况下, 建议使用 <code>post</code> 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	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 <code>notify_id</code> 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: <code>https://mapi.alipay.com/gateway.do?partner=合作者身份ID&notify_id=通知ID</code> 的值	验证是否是支付宝发来的请求
	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 <code>sign</code> 、 <code>sign_type</code> , 根据参数名=参数值的格式, 按首字符字典顺序 (<code>ascii</code> 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 <code>sign</code> 值做比较。	验证返回的签名
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 “ <code>success</code> ” 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 “ <code>success</code> ”, 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 <code>success</code> 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。
	如果交易付款完成时发送的交易状态是 <code>TRADE_SUCCESS</code> (可对交易做其他操作, 如退款、分润等), 则当超过签约合同指定的可退款时间段时, 支付宝会主动发送 <code>TRADE_FINISHED</code> (不能对该交易再做任何操作) 交易状态。此时, 需要根据商户自身业务情况, 来判断是否需对这次的交易完成通知进一步处理。	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、重复付款或订单数据错乱等。
	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



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	遇到特殊错误码（如： <code>UNKNOWN_EXCEPTION</code> 、 <code>GENERIC_FAILURE</code> 、 <code>SYSTEM_ERROR</code> ），须暂停后续一切操作，或者联系支付宝技术支持协助处理。	一般这种情况有可能请求已经提交到支付宝，支付宝系统失去响应但有可能由恢复机制自动回复并提交请求。
	如果遇到错误码“ <code>TRADE_BUYER_NOT_MATC</code> ”，表明支付使用的账户与原来的不一致，请使用原来的账户，或重新创建交易付款。	出现这种情况是买家账户不一致，或者买家使用多个账户，支付时账户必须与购买时账户（创建交易账户）一致。
	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。
	如果遇到错误码“ <code>TRADE_PRICE_NOT_MATCH</code> ”、“ <code>TRADE_TOTALFEE_NOT_MATCH</code> ”，表明商品价格或总价可能发生了变化，请重新创建交易付款。	出现这种情况，可能是商品价格发生了变更。（例如：过了促销期等）
	如果遇到错误码“ <code>FAIL_CREATE_CASHIER_PAY_ORDER</code> ”，可能是支付宝收银台系统出现异常，请稍候重试或联系支付宝技术支持协助处理。	一般是支付宝收银台系统出现异常
	错误码“ <code>BUYER_EMAIL_ID_MUST_NULL</code> ”表示使用纯网关产品时输入了买家账号	使用纯网关产品不需要输入买家账号
	错误码“ <code>DIRECTIONAL_PAY_FORBIDDEN</code> ”表示买家、平台商、卖家、分润方不符合定向支付条件。	可能原因： <ul style="list-style-type: none"> 买家未签署定向支付协议； 买家签署了定向支付协议，但买家、平台商、卖家不符合定向支付条件；或者买家、平台商、分润方不符合定向支付条件。
	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

类型	细则	原因
	如果交易状态出现“TRADE_PENDING”，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、超时时间等）在多次调用中保持一致，且交易尚未完成支付。	防止重复付款以及交易信息被篡改 📖 说明： <ul style="list-style-type: none"> 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； 如果超时时间不一致，则交易状态为等待买家支付，但买家支付时系统会报错。
	即时到账的优势：快捷，不存在卖家发货、买家确认收货的操作，直接将买家的资金转入卖家的支付宝账号。	即时到账交易主要用于虚拟物品和不存在卖家发货、买家确认收货的交易场景，比如手机话费交易、游戏币充值、网上订餐类网站等。
	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。
	即时到账交易不允许卖家在交易创建后修改价格	会影响同一笔交易的支付

类型	规范点	原因
	<p>支付宝每一次退款成功，都会向商户的即时到账异步通知地址 <code>notify_url</code> 发送一条交易成功的通知，其中状态有 <code>refund_status</code>、<code>refund_success</code>。</p> <p> 注意：</p> <ul style="list-style-type: none"> ● 商户要注意做防订单重复更新的代码处理。 ● 交易关闭的状态不会发送通知，因此当进行全额退款操作时，会出现收不到通知的情况，而在部分退款操作时，能收到即时到账接口的交易通知。 <p>解决方案是：</p> <p>商户使用退款接口，把退款的商户业务逻辑程序写到退款接口的通知页面里去，而即时到账接口中不对退款的通知做任何判断。</p>	<p>如果不处理或者处理错误会将商户自己的业务逻辑执行 2 次以上</p>
	<p>如果商户的一笔交易在退款期限内，没有进行退款操作，则支付宝系统会默认将交易的状态改为“交易完成”（即不可退款模式），并主动发送一条交易状态为 <code>trade_finished</code> 的异步通知（可以联系技术支持在后台关闭发送），商户需要根据通知，结合自身业务逻辑做交易不可退款的数据库状态变更。</p>	<p>处理错误会导致商户业务逻辑重复 2 次付款成功</p>
	<p>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。</p>	<p>便于做支付渠道区分</p>
	<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>	<p>保护买家账户安全</p>

类型	规范点	原因
平级分润规则	<p>平级分润格式： 收款方 Email_1^金额 1^备注 1 收款方 Email_2^金额 2^备注 2</p> <p>功能效果： 买家付出了交易金额 100 元，同时，金额 1 给了收款方 Email_1，金额 2 给了收款方 Email_2，seller_email 获得剩下的金额（剩下的金额为：100-金额 1-金额 2-支付宝手续费）。</p> <p>金额计算规则：</p> <ul style="list-style-type: none"> ● 买家交易金额=金额 1+金额 2+seller_email 收款金额+支付宝手续费 ● 收款方 Email_1 实际获得金额=金额 1 ● 收款方 Email_2 实际获得金额=金额 2 ● seller_email 实际获得金额=买家交易金额-金额 1-金额 2-支付宝手续费 	分润规则，避免出现分润错误。
多级分润规则	<p>多级分润格式： 收款方 Email_1^金额 1^备注 1 收款方 Email_1^收款方 Email_2^金额 2^备注 2</p> <p>功能效果： 买家付出了交易金额 100 元，同时，金额 1 给了收款方 Email_1，收款方 Email_1 把获得的金额中的金额 2 给了收款方 Email_2，seller_email 获得剩下的金额（剩下的金额可以为：100-金额 1-支付宝手续费）。</p> <p>金额计算规则：</p> <ul style="list-style-type: none"> ● 买家交易金额=金额 1+seller_email 收款金额+支付宝手续费 ● 收款方 Email_1 实际获得金额=金额 1-金额 2（金额 1 必须大于等于金额 2） ● 收款方 Email_2 实际获得金额=金额 2 ● seller_email 实际获得金额=买家交易金额-金额 1-支付宝手续费 	分润规则，避免出现分润错误。

类型	规范点	原因
限额	<p>使用借记卡快捷支付时，系统会先根据后台配置的借记卡快捷额度档次来进行限额；如果后台未配置额度档次，则会根据商户签约大快捷产品借记卡快捷渠道信息中的额度控制档次来进行限额；如果后台以及签约信息中也未设置额度档次，则默认使用 A 档限额；如果商户后台未设置额度档次且商户未签约大快捷产品借记卡快捷渠道，可使用借记卡快捷体验 B 档额度。</p> <p>借记卡快捷额度档次共分为 4 个档次，其中：</p> <ul style="list-style-type: none"> ● A 档：单笔和单日 2 万限额，单月 5 万限额； ● B 档：单笔和单日 2 千限额，单月 1 万限额； ● C 档：单笔和单日 500 限额，单月 1 千限额； ● O 档：单笔、单日和单月无限额。 	控制借记卡支付风险
	<p>使用信用卡快捷支付时，系统会先根据商户签约大快捷产品信用卡快捷渠道信息中的额度控制档次来进行限额；如果签约了大快捷产品信用卡快捷渠道且未设置额度档次，或签约了独立的信用卡快捷产品，则默认使用 A 档限额；如果商户未签约大快捷产品信用卡快捷渠道且未签约信用卡快捷产品，可使用信用卡快捷体验 C 档额度。</p> <p>信用卡快捷额度档次共分为 3 个档次，其中：</p> <ul style="list-style-type: none"> ● A 档：单笔、单日和单月无限额； ● B 档：单笔和单日 2 千限额，单月 1 万限额； ● C 档：单笔和单日 500 限额，单月 1 千限额。 	控制信用卡支付风险
	<p>使用余额支付时，系统会先根据后台配置的余额额度档次来进行限额；如果后台未配置额度档次，则会根据商户签约信息中的额度控制档次来进行限额；如果后台及签约信息中也未设置额度档次或者商户未签约大快捷产品，则默认使用 A 档限额。</p> <p>余额额度档次共分为 4 个档次，其中：</p> <ul style="list-style-type: none"> ● A 档：单笔和单日 2 万限额，单月 5 万限额； ● B 档：单笔和单日 2 千限额，单月 1 万限额； ● C 档：单笔和单日 500 限额，单月 1 千限额； ● O 档：单笔、单日和单月无限额。 	控制余额支付风险
业务应用注意事项	<p>如果商户签约的是纯网关，那么必须配置请求参数 <code>paymethod</code> 为 <code>bankPay</code>，<code>defaultbank</code> 为某家银行简码。</p>	<p>如果没有设置这两个请求参数，那么支付宝就不会按照纯网关的手续费来收取，而是按照即时到账接口收取费用，导致商户多支出费用。</p>

类型	规范点	原因
	站内不能关闭交易，只能通过站外接口关闭交易。可通过自定义超时参数实现控制交易关闭时间。	交易订单自定义超时关闭功能
	当商户开通了纯网关、大额信用卡、信用卡分期、快捷支付前置、快捷支付网关之中的一个功能时，使用请求参数 defaultbank （默认网银）后，用户付款时支付宝会自动略过网银选择界面。 各功能都有自己的网银简码列表，具体请参考接口技术文档的附录部分。	不同功能使用不同的网银简码才能达到网银前置的效果
	<ul style="list-style-type: none"> 在快捷支付前置接口中，请求参数 default_login 可设置也可不设置； 在快捷支付网关接口中，请求参数 default_login 必须设置。 	在快捷支付前置接口中，不设置该参数不会影响快捷支付前置功能；但在快捷支付网关接口中，则会出现不能正常付款的情况。
	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。	即时到账异步通知与同步通知时间
	支付宝支付手续费扣取模式分为两种： <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。
	如果商户请求时传递了 extra_common_param 参数，支付宝会在同步和异步通知中返回该参数。	支持商户自定义公用回传参数（ extra_common_param ）
	建议商户不要使用类似 js 的 window.open 、 location.replace 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 form 表单提交数据。	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）
	在集成接口的过程中，如果商户使用 seller_email 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 seller_id 做为商户收款账户。	seller_id 值是支付宝分配给商户的唯一身份 ID，不会改变，使用 seller_id 作为收款账户可以避免 seller_email 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 seller_id 值。）
	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。	买家通过何种渠道付款，则款项原路退回。

5 测试流程规则

表5-1 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。
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。