



接入与使用规则

即时到账批量退款无密接口

(refund_fastpay_by_platform_nopwd)

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	3
3 技术接入规则	4
4 接口使用规则	8
5 测试流程规则	11

1 文档说明

1.1 文档说明

本文档是《即时到账批量退款无密接口(refund_fastpay_by_platform_nopwd)》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
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	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。
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。
特殊字符	在 detail_data 的格式中用来作分割符，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（即时到账批量退款无密接口(refund_fastpay_by_platform_nopwd).pdf）、代码示例、本文档（即时到账批量退款无密接口(refund_fastpay_by_platform_nopwd)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。
安全	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
	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 sign、sign_type。	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 _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。	避免接口报错 ILLEGAL_SIGN
	待签名数据应该是参数原始值而不是 url 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。	避免接口报错 ILLEGAL_SIGN
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

类型	细则	原因
	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_input_charset（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
	退款理由中不能包含“^”、“ ”、“\$”、“#”等影响 detail_data 格式的特殊字符。	避免接口无法正常使用
	集成该接口时，退款批次号 batch_no 的格式必须是提交的当天日期+随机数。	业务规则要求
	退分润的备注信息允许为空，但是分割符“^”必须要写，格式要保持一致。	否则会报退款格式错误
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 sign、sign_type。	避免接口无法正常使用
	<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=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； • <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 name="参数名" type="hidden" 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 name="_input_charset" type="hidden" 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。

类型	细则	原因
数据传输	必须使用 https 协议，支持 get 或 post 方式提交。	避免接口无法正常使用
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 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： https://mapi.alipay.com/gateway.do?partner=合作者身份 ID&notify_id=通知 ID 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 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。	验证返回的签名
返回数据处理	在返回参数中，“退款成功总数（ success_num ）”参数值说明一个退款批次中退款成功的总笔数。	避免商户获取状态失败导致业务线的退款全部失败
	退款成功后，除了本接口的异步通知结果返回，还有即时到账接口也会被触发发送交易退款状态通知给商户。	商户要做好业务逻辑判断，以免造成多退款的资损。
	本接口直接返回的 XML 信息（ T/F ）仅代表提交成功与否，不代表退款成功或者失败，商户必须根据退款异步通知中的结果集来判断是否退款成功。	避免由于商户余额不足等问题导致接口提交退款失败使用户投诉
	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。
	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。
	必须判断退款操作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。
	建议每一次退款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
接入环境	不能把接口嵌入 iframe 框架中	避免接口无法正常使用

类型	细则	原因
错误码处理	遇到特殊错误码（如：UNKNOWN_EXCEPTION、GENERIC_FAILURE、SYSTEM_ERROR），须暂停后续操作，或者联系支付宝技术支持协助处理。 HTML 方式返回或异步通知方式返回。	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。
	出现“REFUND_DATE_ERROR”表示退款时间错误，请修改退款时间重新提交请求。 HTML 方式返回。	出现这种情况可能原因： <ul style="list-style-type: none">退款时间为空或格式不符合 yyyy-MM-dd HH:mm:ss;退款日期与退款批次号中的日期不一致。
	出现“TRADE_STATUS_ERROR”表示在该交易状态下不允许退款。 HTML 方式返回。	可能原因：因交易一方账号被锁定，导致交易处于挂起状态而不能退款。
	出现“REFUND_FAIL”表示退款失败，请联系支付宝技术支持协助处理。 异步通知方式返回。	该结果码只会出现在做意外数据恢复时，找不到结果码的情况。
	出现“RESULT_AMOUNT_NOT_VALID”表示退款金额不合理，请重新检查退款金额。 异步通知方式返回。	可能原因：一般是交易金额大于退款金额，如退款的票面价、机建费、燃油费、航意险大于交易时的相关金额。
	出现“REFUND_TRADE_FEE_ERROR”表示交易退款金额与付款金额不一致，请更新交易退款金额。 异步通知方式返回。	出现原因是退款金额与付款金额不一致
	出现“BATCH_REFUND_LOCK_ERROR”表示同一时间不允许进行多笔并发退款。 异步通知方式返回。	同一时间不允许进行多笔并发退款
	出现“PARTNER_NOT_SIGN_PROTOCOL”表示分润支付宝账号及收款账号没有和签约支付宝账号签署第三方协议“支付圈签约接口 (sign_protocol_with_partner)”	分润方（分润支付宝账号）及收款方（收款支付宝账号，即请求参数 seller_email，不是签约支付宝账号）没有和签约支付宝账号签署第三方协议“支付圈签约接口 (sign_protocol_with_partner)”
	遇到如下错误码时，系统会启动重试机制： <ul style="list-style-type: none">TRADE_PAYMENT_ERRORTRADE_PROCESS_ERRORQUERY_PARTICIPANT_ERROR 商户无需处理。	支付宝系统内部错误，可以由支付宝内部自行解决

类型	细则	原因
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
网络	确保网络顺畅	防止接口调用失败
业务操作规则	接口支持重复调用,但必须确保每个合作伙伴每个退款批次号(batch_no)的唯一性。	防止重复退款
	同一批明细里(detail_data)不允许包含两条交易号相同的退款明细	防止重复退款
	必须保证要退款的账号余额充足	避免退款不成功
	如果退分润与退款同时进行,那么必须保证分润方的账户有充足的余额。	避免退款不成功
	要退款的交易号必须存在且已经是付款成功状态,即交易状态(TRADE_STATUS)是 TRADE_SUCCESS 的状态。	避免退款不成功
	退分润的分润支付宝账号必须与平台或者退款接口使用商户签约支付圈接口	否则退款时会报没有签约的错误,即“PARTNER_NOT_SIGN_PROTOCOL”。
	收款账号不是签约账号,即参数 seller_email 填写的支付宝账号对应的合作者身份 ID 不是参数 partner 对应的合作者身份 ID,该 seller_email 填写的支付宝账号必须与平台或者退款接口使用商户签约支付圈接口。	否则退款时会报没有签约的错误,即“PARTNER_NOT_SIGN_PROTOCOL”。

类型	规范点	原因
	如果请求支付宝以后 IS_SUCCESS 返回 T ，那么当该批次中有一个或多个交易退款失败，此时不影响整个批次中其他交易的退款。	<ul style="list-style-type: none"> 如果请求支付宝后 IS_SUCCESS 返回为 F，那么该笔批次的所有交易退款都失败； 如果请求支付宝后 IS_SUCCESS 返回为 T，该笔批次中的各个交易彼此独立，所以失败的交易退款不会影响成功的交易退款。
	如果接口中设置了 notify_url ，当请求支付宝以后 IS_SUCCESS 返回为 F 时，支付宝系统不会发送服务器异步通知。	对请求的校验未通过
	余额支付功能关闭的情况下，接口退款不受影响。	保证商户顺利完成交易退款
	该接口支持单笔和批量退款，批量退款最多一次支持 1000 笔，50 笔是最高效的退款笔数。数据请求以表单 POST 方式发送到支付宝。	无法传递大量数据
	需要对退款成功后的异步通知返回结果做所有返回参数的解析分类，判断交易状态即可。	防止未来业务变更，有更好的兼容性。
	签约自动退款协议时需要安装数字证书	牵扯到支付密码，所以必须使用更为安全的签约模式。
	现有的退款模式有 2 种： 1、平台响应退款数据集格式 2、供应商响应退款数据集格式	<ul style="list-style-type: none"> 第一种模式的优点：和支付宝交互少； 缺点：分润方资金不足，平台资金充足，平台会垫付退款资金。 第二种模式的优点：可以保证分润方必须将资金退给平台； 缺点：交互次数多。
	两个平台的退款协议，一个平台（A）和另一个平台（B）签约，A 平台的供票商签约不会自动默认已经和 B 平台签约。	防止跨平台无密退款签约报错
	一笔交易可以多次退款，只需要遵守多次退款的总金额不超过该笔交易付款金额的原则。	防止重复退款
	一笔交易的退款次数上限为 99 次	超过该次数则支付宝系统不会处理该笔交易的退款，如果接口中设置了 notify_url ，那么支付宝系统也不会发送服务器异步通知。
单笔数据集	退款笔数总和要等于参数 batch_num 的值	数据一致，防止发生退款金额错误。

类型	规范点	原因
规则	不能包含两条交易号相同的退款明细，同一条退款明细（第 N 笔交易）中不能存在两条转出人和转入人都相同的退分润信息。	防止重复退款
收费退款数据集规则	被收费人支付宝账号与收费人支付宝账号对应用户 ID 只能填写一项；如果被收费人支付宝账号与收费人支付宝账号对应用户 ID 填写了一项，那么另一项必须置空。如：被收费人支付宝账号^^退款金额^退费理由	数据完整一致，避免出现退款账户信息错误。
	如果被收费人的支付宝账号与用户 ID（2088 开头的 16 位纯数字）不一致，系统以用户 ID 为准。	数据完整一致，避免出现退款账户信息错误。
	调用接口退款，支付宝自动计算退支付手续费金额。	避免商户计算退手续费不一致导致的资损问题
分润退款数据集规则	转入方是收到退回金额的支付宝账号，转出方是付出退回金额的支付宝账号。	数据完整一致，避免出现退款账户信息错误。
	支付宝账号必须为邮箱格式，不能是手机格式，或者使用支付宝账号对应的用户 ID。	数据完整一致，避免出现退款账户信息错误。
	被收费人支付宝账号与收费人支付宝账号对应用户 ID 只能填写一项；如果被收费人支付宝账号与收费人支付宝账号对应用户 ID 填写了一项，那么另一项必须置空。如：转出方支付宝账号^^转入方支付宝账号^^分润金额^退款理由	数据完整一致，避免出现退款账户信息错误。
	如果被收费人的支付宝账号与用户 ID（2088 开头 16 位纯数字）不一致，系统以用户 ID 为准。	数据完整一致，避免出现退款账户信息错误。
	无密退款在保证资金平的基础上可以做到先分润给 2 方（A 和 B），后期 A 方退款给平台后，平台再次分润给 B 方。	支付宝支持这种资金流设置，业务要求后才开通。
退款资金流向	客户银行卡挂失或者注销将无法退回到客户账户中。需要商户向客户收集新的银行卡号并且提供声明对新的银行卡号正确性负责，委托支付宝退款。	退回到卡的情况：信用卡、信用卡卡通、Q 账号、支付宝中间账号（例如：0571-88158090@alipay.com）支付的订单；其他的实时到账退到支付宝余额账户。
	一定要遵循“入大于出，先入后出”的规则。即退款总金额不能大于交易总金额，先收回分润出去的钱再退款。	防止发生退款金额错误

类型	规范点	原因
接口异常	异步处理如果响应时间过长直接导致退款返回失败，或者退款通知超过 24 小时未发出，则可以人工进行重新提交请求。	可能是支付宝的消息积压导致异步调度任务失败

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"> 根据接口要求配置批量退款请求参数，发送请求数据 返回批量退款处理结果数据 	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"> 根据接口要求配置批量退款请求参数，发送请求数据 返回批量退款处理结果数据 	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	<ul style="list-style-type: none"> 根据接口要求配置批量退款请求参数，发送请求数据 返回批量退款处理结果数据 对获取的返回结果数据进行处理 	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 根据接口要求配置批量退款请求参数，发送请求数据 返回批量退款处理结果数据 对获取的返回结果数据进行处理 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 根据接口要求配置批量退款请求参数，发送请求数据 返回批量退款处理结果数据 对获取的返回结果数据进行处理 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。