■ 目录

1. 引言

1.1. 文档概述

支付宝对外接口分为两种，一种是接收外部请求的接口，我们统称为外部接入接口。一种是主动通知外部系统的接口，我们统称为外部通知接口。

外部服务接口的主要目的是让外部合作伙伴主动使用我们的服务，如：创建交易等。外部通知接口的主要目的是为外部合作伙伴提供数据同步服务(如：交易状态同步）以及异步处理结果返回服务（有些业务的处理是无法做到即时返回的）。

阅读对象：商户系统（商户）的技术开发人员。

1.2. 业务术语

名称	说明	
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）	

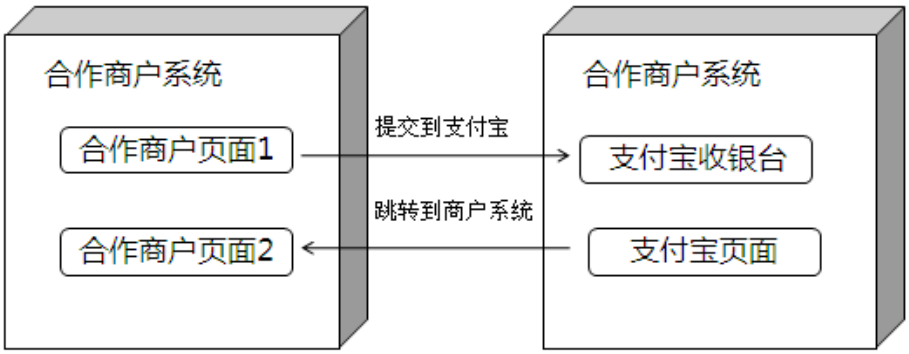
2. 交互模式

2.1. 请求/响应交互模式

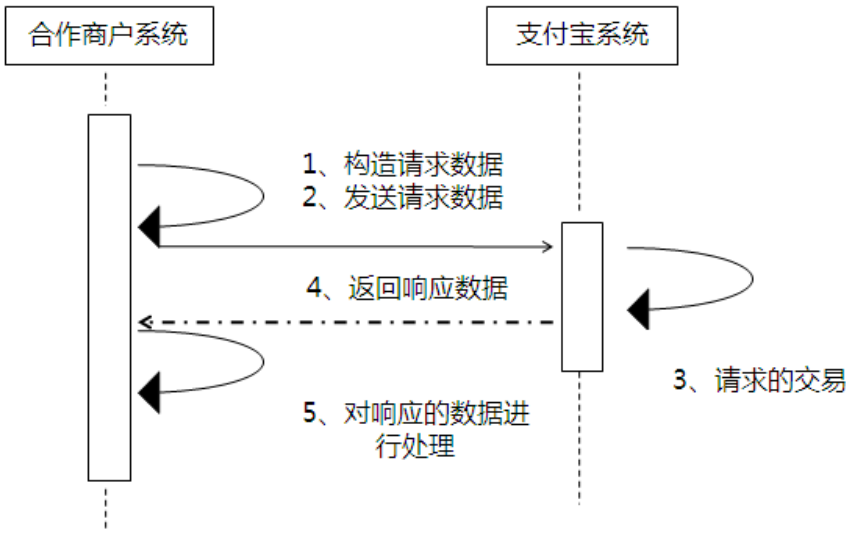
请求/响应模式是最常用的一种交互模式。在这种交互模式下，商户系统向支付宝系统发送请求数据，并同步等待支付宝系统处理完毕之后返回的响应数据。

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：

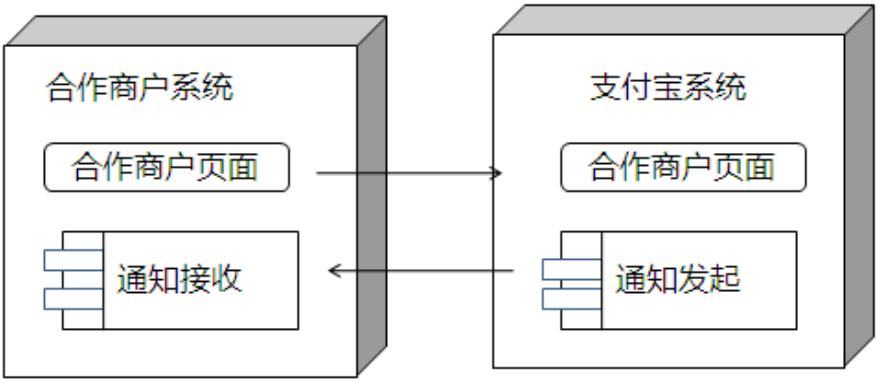
```
https://www.alipay.com/cooperate/gateway.do
```

2.2. 主动通知交互模式

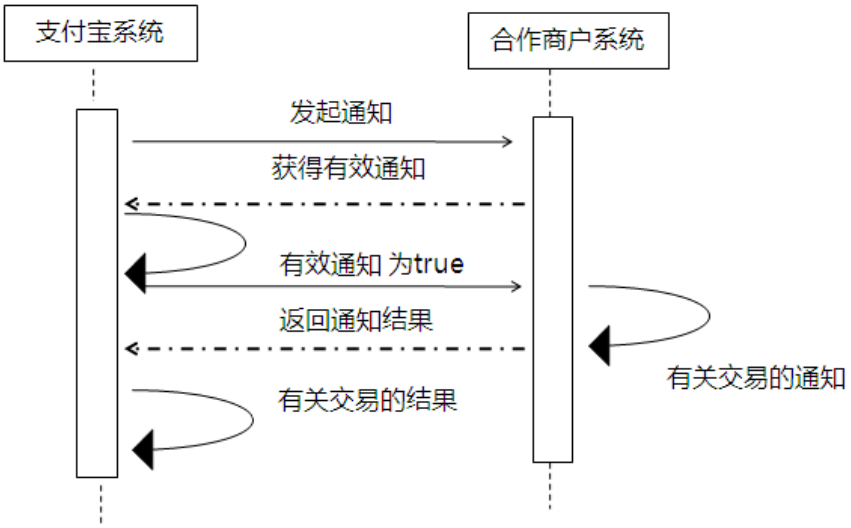
买家从商户网站跳转到支付宝网站，在支付宝网站完成最后操作，买家不用再回到商户网站。支付宝单系统会将商户关注的事件采用主动通知的方式提交给商户系统。

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= trade_create_by_buyer,partner=20880063000,email=test@msn.com

那么待签名数据就是：

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

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。（参见：签名及加密算法）

4. 接口

4.1. 外部接入接口

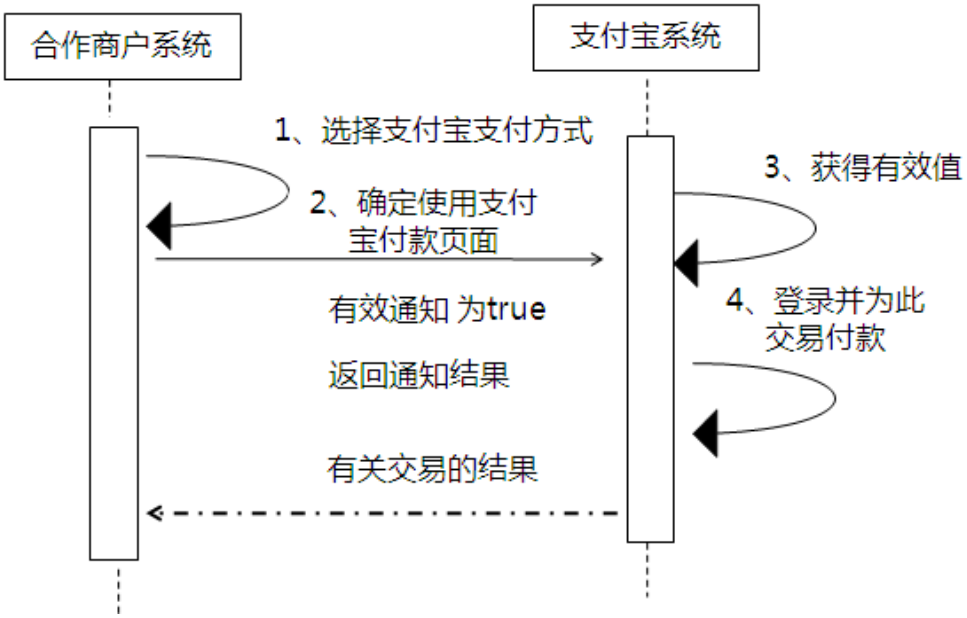
4.1.1.业务功能

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，页面跳转到支付宝收银台，支付宝会自动向买家显示的商品报价。

此接口包含两种支付入口，担保交易和及时到帐交易来创建交易。

4.1.2.交互模式

请求/响应交互模式，页面跳转



4.1.3. 请求参数列表

字段名	变量名	类型	说明	可空
协议参数				
接口名称	service	String	service= trade_create_by_buyer	N
商户	partner	String(16)	商户在支付宝的用户 ID	N
通知 URL	notify_url	URL	针对该交易支付成功之后的通知接收 URL。	N
返回 URL	return_url	URL	交易付款成功之后，则结果返回 URL，仅适用于立即返回处理结果的接口。	Y
签名	sign	String	参见：签名机制	N
签名方式	sign_type	String	参见：签名方式	N
参数编码字符集	_input_charset	默认为 GBK	商户系统与支付宝系统之间交互信息时使用的编码字符集。商户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
业务参数				

商品名称	subject	String(256)	商品标题	N
商品描述	body	String(400)	商品描述	Y
外部交易号	out_trade_no	String(64)	商户交易号（确保在商户系统中唯一）	N
商品单价	price	Number(13,2)	price：单位为 RMB Yuan 0.01～100000000.00 total_fee:单位为 RMB Yuan 0.01～100000000.00 quantity:0<quantity < 1000000 规则： 1.总价和单价不可以同时出现 2.如果输入的是总价，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,当数量为 1 时，设置单价为总价 3.如果如数的是单价，则商品数量必须输入	N
交易金额	total_fee	Number(13,2)		
折扣	discount	Number(13,2)	范围为-10000000.00～10000000.00	Y
商品展示网址	show_url	String(400)	详情的一个连接地址：譬如:http://www.alipay.com	Y
购买数量	quantity	Number(6,0)	1～1000000000	N
支付类型	payment_type	String	1	N
物流类型	logistics_type	String	参见：物流类型枚举表	N
物流费用	logistics_fee	Number(8,2)	0.00～10000.00,默认为 0（小数点后面最多两位）	N
物流支付类型	logistics_payment	String	参见：物流支付类型枚举表	N
收货人姓名	receive_name	String(128)	传递这个参数的值就不能为空	Y
收货人地址	receive_address	String(256)	至少包括姓名、联系地址和邮政编码	Y
收货人邮编	receive_zip	String(6)	邮编 5 位或 6 位数字组成	Y
收货人电话	receive_phone	String(30)	长度必须在 1-30 位以内	Y

收货人手机	receive_mobile	String(11)	必须是 11 位数字	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(30)		N
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)		Y
买家逾期不付款,自动关闭交易的期限	t_b_pay	如果商家未设置支持自定义超时,该参数应该为空,否则会报错		1d
卖家逾期不发货,建议买家退款的期限	t_s_send_1	如果商家未设置支持自定义超时,该参数应该为空,否则会报错		7d
买家逾期不确认收货,自动完成交易(平邮)的期限	t_s_send_2	如果商家未设置支持自定义超时,该参数应该为空,否则会报错		3d

物流信息中可以传递多个 logistics_type、logistics_fee、logistics_payment 参数,后面添加 _index ,如:第一组物流信息参数名为 logistics_type_1、logistics_fee_1、logistics_payment_1,第二组物流信息参数为 logistics_type_2、logistics_fee_2、logistics_payment_2 ,依次类推。传递多种物流方式时 ,使用 logistics_type、logistics_fee、logistics_payment 表示缺省物流方式(特别注意:使用多种物流的时候,缺省物流不可少,缺省物流必须是多组物流中的一组)。通过这种方式合作伙伴可以传递多种物流选择方式供买家选择。如果只有一种物流方式的话,就是用 logistics_type、logistics_fee、logistics_payment 作为参数名就可以了。

例如:

某交易支持 EMS 与 平邮两种不同的物流,且 EMS 为默认物流,则传入的参数应该为:

logistics_type=EMS&logistics_fee=25.00&logistics_payment=BUYER_PAY&
logistics_type_1=EMS&logistics_fee_1=25.00&logistics_payment_1=BUYER_PAY
&logistics_type_2=POST&logistics_fee_2=5.00&logistics_payment_2= BUYER_PAY

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%401
26.com&discount=0&logistics_fee=0.01&notify_url=http%3A%2F%2Floca
lhost%3A8088%2Fjsp_shi_gbk%2Falipay_notify.jsp&payment_type=1&ser
vice=trade_create_by_buyer&partner=2088002123456782&_input_charse
t=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=h
ttp%3A%2F%2Flocalhost%3A8088%2Fjsp_shi_gbk%2Falipay_return.jsp&si
gn=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>
```

4.2. 外部通知接口

4.2.1.通知返回参数列表

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通知

1.合作伙伴通过“标准实物担保”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状态、交易金额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。

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	参见：签名机制	N
签名方式	sign_type	String	参见：签名方式	N
支付宝交易号	trade_no	String(64)	该交易在支付宝系统中的交易流水号	N
外部交易号	out_trade_no	String(64)	该交易在合作伙伴系统的流水号	N
商品名称	subject	String(256)		N
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	N
折扣	discount	Number(8,2)	-10000000.00～10000000.00	Y
购买数量	quantity	Number(6,0)	>0	N
交易金额	total_fee	Number(13,2)	单位为 RMB Yuan 0.01～1000000.00	N
支付类型	payment_type	String	1	Y
是否使用红包	use_coupon	String(1)	T/F	N
红包折扣	coupon_discount	Number(8,2)	-10000000.00～10000000.00	Y
金额是否修改过	is_total_fee_adjust	String(1)	T/F	N
交易状态	trade_status	String	参见：交易状态枚举表	N
退款状态	refund_status	String	参见：退款状态枚举表	Y
物流状态	logistics_status	String	参见：物流状态枚举表	Y
物流类型	logistics_type	String	参见：物流类型枚举表	Y

物流费用	logistics_fee	Number(8,2)	0.00～10000000.00	Y
物流支付类型	logistics_payment	String	参见：物流支付类型枚举表	Y
收货人姓名	receive_name	String(128)		Y
收货人地址	receive _address	String(256)		Y
收货人邮编	receive _zip	String(6)		Y
收货人电话	receive _phone	String(30)		Y
收货人手机	receive _mobile	String(11)		Y
卖家 Email	seller_email	String(100)		N
卖家 ID	seller_id	String(30)		N
买家 ID	buyer_id	String(30)		N
买家 Email	buyer_email	String(100)		N
交易创建时间	gmt_create	Timestamp		Y
买家付款时间	gmt_payment	Timestamp		Y
卖家发货时间	gmt_send_goods	Timestamp		Y
退款时间	gmt_refund	Timestamp	交易处于当前退款状态时的时间。（当发生退款时间有该条记录，否则没有）	Y
交易结束时间	gmt_close	Timestamp		Y
物流状态更新时间	gmt_logistics_modify	Timestamp		Y

4.2.2.物流支付类型枚举表

数值	类型	说明
SELLER_PAY	卖家支付	由卖家支付物流费用（费用不用计算到总价内）
BUYER_PAY	买家支付	买家支付物流费用（费用需要计算到总价内）

4.2.3.物流类型枚举表

物流参数类型列表值	说明
POST	平邮
EMS	EMS
EXPRESS	其他快递

4.2.4.物流状态枚举表

数值	说明
INITIAL_STATUS	初始状态
WAIT_LOGISTICS_FETCH_GOODS	等待物流取货
WAIT_LOGISTICS_SEND_GOODS	等待物流发货
LOGISTICS_SENDING	物流发货中

WAIT_RECEIVER_CONFIRM_GOODS	等待收货人确认收货
GOODS_RECEIVED	货物收到了
LOGISTICS_FAILURE	物流失败

4.2.5.交易状态枚举表

交易状态列表值	说明
WAIT_BUYER_PAY	交易创建
WAIT_SELLER_SEND_GOODS	买家付款成功
WAIT_BUYER_CONFIRM_GOODS	卖家发货成功
TRADE_FINISHED	交易成功结束
TRADE_CLOSED	交易关闭
modify.tradeBase.totalFee	修改交易价格

4.2.6.退款状态枚举表

状态代码	状态名称
WAIT_SELLER_AGREE	买家申请退款
REFUND_SUCCESS	退款成功
REFUND_CLOSED	退款关闭

4.2.7.错误代码列表

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
COMMISION_ID_NOT_EXIST	佣金收取帐户不存在
COMMISION_SELLER_DUPLICATE	收取佣金帐户和卖家是同一帐户
COMMISION_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
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
COMMISION_NOT_SUPPORT	不支持佣金
VIRTUAL_NOT_SUPPORT	不支持虚拟发货方式
ILLEGAL_CHARSET	字符集不合法

5. 应用场景

5.1. 场景描述

商户买家 NaNa 有一个支付宝账户，在商户网站上浏览商品，很快找到一款她非常喜欢的化妆品，赶紧放入购物车，准备结账，选择支付宝方式付款了。

NaNa 选择支付宝方式付款后，商户系统调用支付宝系统的支付接入接口（参见：支付接入接口），从商户网站跳转到支付宝的页面，NaNa 输入支付宝账户和支付密码开始付款，如果她的支付宝帐户有足够这笔交易的余额，那么她可以迅速完成交易付款；如果他的支付宝账户没有足够这笔交易的余额，她可以选择网银付款或者支付宝的卡通产品付款。

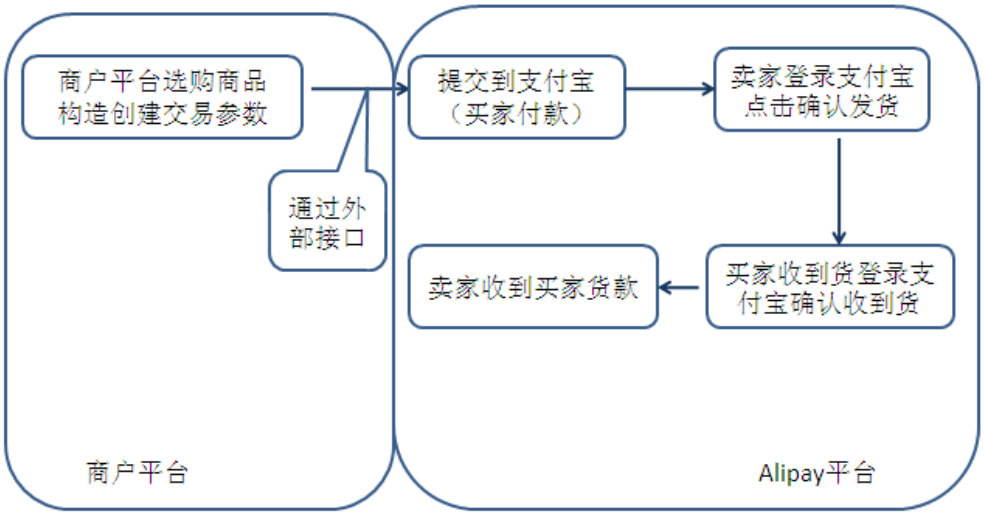
NaNa 支付成功后，支付宝系统向商户系统发送交易信息的通知，商户系统在接收到通知之后（参见：外部通知接口，向支付宝系统发送通知验证请求（参见：通知验证），判断该通知的合法性和真实性。支付宝系统验证通知，并返回

验证结果。如果验证结果为 true，商户系统根据该通知做相应的处理，比如给买家 NaNa 发货。处理完毕，返回结果给支付宝系统(输出 success)。

商户还可以根据 NaNa 本次交易的外部交易号（out_trade_no）查询商户网站内的相关交易信息，判断 NaNa 是否是此订单，确认正确，商户就可以开始给 NaNa 发货（在支付宝网站上点击发货）。

NaNa 收到期待很久的化妆品，却发现这不是她想要的牌子，经过协商之后，商户答应 NaNa 将会委托支付宝把款项退还给她。商户在支付宝系统中发起退款。如果 NaNa 表示满意，那么 NaNa 就可以登录支付网站系统，点击确认收到货，那么 NaNa 支付给商户的款项就会自动划入商户的签约支付宝帐号中。

5.1.1. 交易流程图例



5.2. 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：

```
<form
name="alipaysubmit"method="post"action="https://www.alipay.com/cooper
ate/gateway.do?_input_charset=utf-8">
<input type=hidden name="body" value="a">
<input type=hidden name="logistics_type" value="EMS">
```

```
<input type=hidden name="logistics_fee" value="10">
<input type=hidden name="logistics_payment" value="BUYER_PAY">
<input type=hidden name="out_trade_no" value="2008111517451234">
<input type=hidden name="partner" value="2088002123456782">
<input type=hidden name="payment_type" value="1">
<input type=hidden name="seller_email" value="test@msn.com">
<input type=hidden name="service" value=" trade_create_by_buyer">
<input type=hidden name="sign" value="abc123">
<input type=hidden name="sign_type" value="MD5">
<input type=hidden name="subject" value="商品名称">
<input type=hidden name="price" value="0.01">
<input type=hidden name="quantity" value="1">
<input type=hidden name="discount" value="0">
<input type=hidden name="show_url" value="http://www.alipay.com">
<input type=hidden name="return_url" value="http://www.alipay.com ">
```

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。

5.2.1. 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=utf-8
body=a
discount=0
logistics_fee=10
  ……
seller_email=test@msn.com
service=trade_create_by_buyer
show_url=www.sina.com.cn
subject=商品名称
```

使用“&”符号把参数串联起来，产生待签名数据：

```
_input_charset=utf-8&body=a&discount=0&logistics_fee=
10&logistics_payment=SELLER_PAY&logistics_type=EMS&ou
t_trade_no=2008111517451234&partner=2088002123456782&
payment_type=1&price=0.01&quantity=1&return_url=http:
//www.alipay.com&seller_email=test@msn.com&service=tr
```

```
ade_create_by_buyer&show_url=www.sina.com.cn&subject=
商品名称
```

5.2.2. 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=utf-8&body=a&discount=0&logistics_fee=1
0&logistics_payment=SELLER_PAY&logistics_type=EMS&out_
trade_no=2008111517451234&partner=2088002123456782&pay
ment_type=1&price=0.01&quantity=1&return_url=http://ww
w.alipay.com&seller_email=test@msn.com&service=trade_c
reate_by_buyer&show_url=www.sina.com.cn&subject=商品名
称abc123
```

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=4b04730e2e8a0a034fa66c509030f8af

5.2.3. 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?sel
ler_email=test%40msn.com&discount=0&logistics_f
ee=0.01&notify_url=http%3A%2F%2Flocalhost%3A808
8%2Fjsp_shi_gbk%2Falipay_notify.jsp&payment_typ
e=1&service=trade_create_by_buyer&partner=20880
02509209142&_input_charset=utf-8&logistics_type
=EMS&price=0.01&out_trade_no=20081118082127&sub
ject=%E5%95%86%E5%93%81%E5%90%8D%E7%A7%B0&logis
tics_payment=SELLER_PAY&quantity=1&body=a&retur
n_url=http%3A%2F%2Flocalhost%3A8088%2Fjsp_shi_g
bk%2Falipay_return.jsp&show_url=www.sina.com&sig
n=4b04730e2e8a0a034fa66c509030f8af&sign_type=MD
5
```

5.2.4. 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

5.3. 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

5.4. 签名及加密算法

5.4.1. 签名算法对比

算法 / 功能	MD5	DSA	RSA
防篡改	√	√	√
防抵赖	×	√	√
加密	×	×	√
电子签名法是否承认	×	√	√

5.4.2. MD5 签名算法

MD5 是一种摘要生成算法，本来是不能用于签名的。但是，通过在待签名数据之后加上一串私密内容（指令发送、接收双发事先规定好的，这里我们称其为签名密钥），就可以用于签名了。

例如：

假设签名密钥是 32#af*dsf，那么商户系统调用某接口的预签名数据就是：

email=test@msn.com&service=trade_create_by_buyer32#af*dsf

使用这种算法签名只能起到防数据篡改的功能，不能起到签名防抵赖的功能，因为双方都知道签名密钥。

5.4.3. DSA 签名算法

DSA 是一种非对称的签名算法，即签名密钥（私钥）与验签名密钥（公钥）是不一样的，私钥用于签名，公钥用于验签名。使用这种算法签名在起到防数据篡改功能的同时，还可以起到防抵赖的作用，因为私用只有签名者知道。

商户系统发送请求时，使用自己的密钥对待签名数据进行 DSA 签名，支付宝系统使用商户的公钥进行校验；支付宝系统返回数据时，使用支付宝的密钥对待签名数据进行 DSA 签名，商户使用支付宝的公钥进行校验。

5.4.4. RSA 签名算法

RSA 也是一种非对称算法，同时，它还是一种加密算法，使用方法跟 DSA 签名算法类似。

5.5. OpenSSL 命令

5.5.1. DSA 密钥生成命令

1. 生成 DSA 参数

```
openssl dsaparam -out dsa_param.pem 1024
```

2. 生成 DSA 私钥

```
openssl gendsa -out dsa_private_key.pem dsa_param.pem
```

3. 生成 DSA 公钥

```
openssl dsa -in dsa_private_key.pem -pubout -out dsa_public_key.pem
```

4. 将 DSA 私钥转换成 PKCS8 格式

```
openssl pkcs8 -topk8 -inform PEM -in dsa_private_key.pem -outform PEM
-nocrypt
```

5.5.2. RSA 密钥生成命令

1. 生成 RSA 私钥

```
openssl genrsa -out rsa_private_key.pem 1024
```

2. 生成 RSA 公钥

```
openssl rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem
```

3. 将 RSA 私钥转换成 PKCS8 格式

```
openssl pkcs8 -topk8 -inform PEM -in rsa_private_key.pem -outform PEM
-nocrypt
```

5.5.3. 签名/验签名命令

✧ RSA 签名

```
openssl sha1 -sign rsa_private_key.pem -out rsasign.bin plaintext.txt
```

✧ RSA 验签名

```
openssl sha1 -verify rsa_public_key.pem -signature rsasign.bin
plaintext.txt
```

✧ DSA 签名

```
openssl dgst -dss1 -sign dsa_private_key.pem -out dsasign.bin
plaintext.txt
```

✧ DSA 验签名

```
openssl dgst -dss1 -verify dsa_public_key.pem -signature dsasign.bin
plaintext.txt
```

对二进制签名做 Base64 编码

```
openssl base64 -in rsasign.bin -out base64.txt
```

对 base64 编码过的签名做 base64 解码

```
openssl base64 -d -in base64.txt -out rsasign.bin
```