

## 接口：外部（纯）担保交易服务

Service: create\_partner\_trade\_by\_buyer

### 修订历史

版本号	作者	修订章节	修订原因	修订日期	审阅人
2.0					

## 术语

术语	解释
外部（纯）担保交易接口	提供给外部商家创建标准支付宝交易的接口

## 功能描述：

此接口,用户只能选择担保交易来创建交易

## 访问形式:

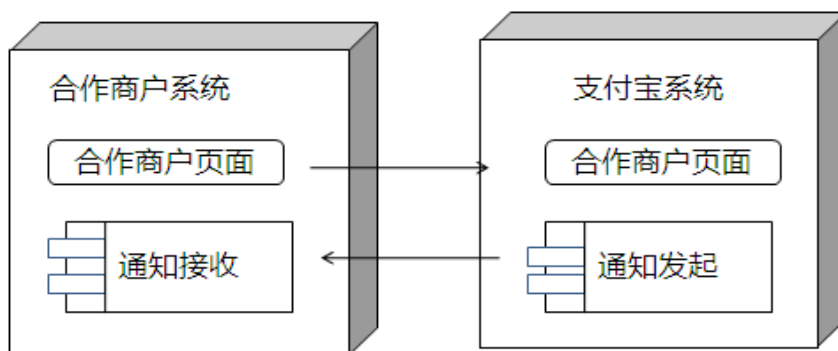
从外部商户页面直接跳到如下页面:



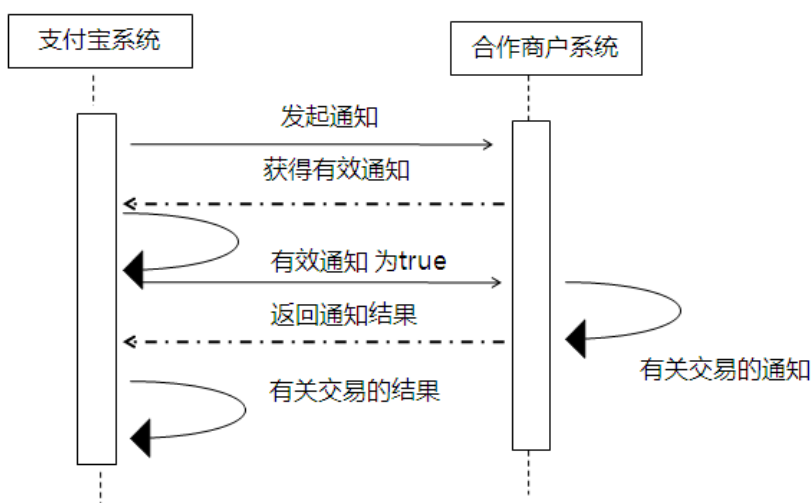
买家从商户网站跳转到支付宝网站, 在支付宝网站完成最后操作, 买家不用再回到商户网站。支付宝单系统会将商户关注的事件采用主动通知的方式提交给商户系统。

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。



#### 1.1.1. 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。
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。

### 1.1.2. 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=abcdefghijklmnpqrst
```

#### ✧ 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=abcdefghijklmnpqrst
```

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

## 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。

### 1.1.3. 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\_partner\_trade\_by\_buyer,partner=20880063000,email=test@msn.com

那么待签名数据就是：

email=test@msn.com&partner=20880063000&service= create\_partner\_trade\_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。
- ✧ 按照 sign\_type 参数指定的签名算法对待签名数据进行签名。

## 1.22 请求输入参数表

特别注意：

a. 您必需使用 HTTPS 协议，详见[请求 url 样例](#)。

b. 特别注意：请按照支付宝外部请求接口概述中要求的签名方式，对输入参数进行签名，该接口请求才能够被支付宝系统接收。

c. 此接口支持重复调用，传入的参数和前一次调用的情况要一致，进入后显示交易信息。

字段名	变量名	类型	说明	可空	样例
基本参数					
接口名称	service	String	create_partner_trade_buyer	N	create_partner_trade_by_buyer
合作伙伴 ID	partner	String(16)	合作伙伴在支付宝的用户 ID	N	654851222
通知 URL	notify_url	URL	针对该交易的交易状态同步通知接收 URL。	N	http://test.com
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	http://test.com
代理商 ID	agent	String(16)	如果一些交易网站的交易，有一定的“代理”所属关系，代理商可以在交易中传递该参数，来表明代理的身份。这里传送的值，请使用代理商所属支付宝账户的 PartnerID。	N	21465665
签名	sign	String	见签名机制	N	dergdjv1gh0894q2gcjq447tcyn4dr3i1
签名方式	sign_type	String	见签名方式	N	DSA
参数编码字符集	_input_charset	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	N	utf-8



			也会使用该字符集对返回参数或通知参数进行编码。 <b>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</b>		
商品名称	subject	String(256)		N	水果
商品描述	body	String(400)		Y	很好吃的
外部交易号	out_trade_no	String(64)	合作伙伴交易号（确保在合作伙伴系统中唯一）	N	205485121225
商品单价	price	Number(13, 2)	单位为 RMB Yuan 0.01~100000000.00（小数点后面最多两位）	N	8412121222214
折扣	discount	Number(13, 2)	-10000000.00 ~ 10000000.00（小数点后面最多两位）	Y	10.00
商品展示网址	show_url	String(400)		Y	http://www.display.com.cn
购买数量	quantity	Number(6, 0)	大于 0 小于 1000000000	N	5
支付类型	payment_type	String	见公用枚举表-支付类型表	N	1
物流类型	logistics_type	String	见公用枚举表-物流类型枚举表	N	VIRTUAL
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	10.00
物流支付类型	logistics_payment	String	见公用枚举表-物流支付类型枚举表	N	SELLER_PAY
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)	卖家在支付宝的注册 Email 或注册 ID, 两者任何一个。	N	maaimin@yahoo.com.cn
卖家 ID	seller_id	String(30)			25845544488
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	maaimin@yahoo.com.cn
买家 ID	buyer_id	String(30)			25845544488
买家留言	buyer_msg	String(200)		Y	尽快给我发货哦
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

### 1.23 请求 URL 样例:

[https://www.alipay.com/cooperate/gateway.do?logistics\\_fee\\_1=23&logistics\\_type\\_1=EMS&buyer\\_email=tianc\\_alipay%40alipay.com&service=create\\_partner\\_trade\\_by\\_buyer&partner=2088002007018916&logistics\\_type=EMS&price=34.89&logistics\\_payment\\_2=BUYER\\_PAY&t\\_b\\_rec\\_post=7d&quantity=3&body=testABC&return\\_url=http%3A%2F%2Fwww.blive.cn%2Falipay%2Findex.php&logistics\\_fee\\_2=23&seller\\_email=tianc\\_alipay%40alipay.com&t\\_s\\_send\\_2=7d&discount=-5.8&t\\_s\\_send\\_1=7d&logistics\\_fee=23&payment\\_type=1&logistics\\_type\\_2=POST&agent=2088002007018916&logistics\\_payment\\_1=BUYER\\_PAY&out\\_trade\\_no=20080303-8&logistics\\_payment=BUYER\\_PAY&subject=testABC&t\\_b\\_pay=7d&sign=gnmn0Gkob30XY1UQxgsqVFtpXz3f0Vaa2oeIFy2kXy%2FgLcwg%3D%3D&sign\\_type=DSA](https://www.alipay.com/cooperate/gateway.do?logistics_fee_1=23&logistics_type_1=EMS&buyer_email=tianc_alipay%40alipay.com&service=create_partner_trade_by_buyer&partner=2088002007018916&logistics_type=EMS&price=34.89&logistics_payment_2=BUYER_PAY&t_b_rec_post=7d&quantity=3&body=testABC&return_url=http%3A%2F%2Fwww.blive.cn%2Falipay%2Findex.php&logistics_fee_2=23&seller_email=tianc_alipay%40alipay.com&t_s_send_2=7d&discount=-5.8&t_s_send_1=7d&logistics_fee=23&payment_type=1&logistics_type_2=POST&agent=2088002007018916&logistics_payment_1=BUYER_PAY&out_trade_no=20080303-8&logistics_payment=BUYER_PAY&subject=testABC&t_b_pay=7d&sign=gnmn0Gkob30XY1UQxgsqVFtpXz3f0Vaa2oeIFy2kXy%2FgLcwg%3D%3D&sign_type=DSA)

## 1.24 输出参数

注意：接口以通知方式进行返回

A. 页面跳转返回的参数列表(内部重定向到 createPartnerTradeByBuyer.vm)

B. 通知返回的参数列表(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)

基本参数					
字段名	变量名	类型	说明	可空	样例
通知类型	notify_type	String	trade_status_sync	N	trade_status_sync
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	123584545
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	20005-05-05 12:21:21
签名	sign	String	见 HTTP 参数签名机制	N	dergdjvlg0 894q2gcjq44 7tcyn4dr3i1
签名方式	sign_type	String	见签名方式	N	DSA
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	54115411
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	5415125121
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	11.00
折扣	discount	Number(8, 2)	-10000000.00 ~ 10000000.00	Y	1.00
购买数量	quantity	Number(6, 0)	>0	N	5
交易金额	total_fee	Number(13, 2)	单位为 RMB Yuan 0.01~1000000.00	N	50
支付类型	payment_type	String	见公用枚举表-支付类型枚举表	Y	1

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	1.00
交易金额是否修改过	is_total_fee_adjust	String(1)	T/F	N	T
交易状态	trade_status	String	见公用枚举表-交易状态枚举表	N	WAIT_BUYER_PAY
退款状态	refund_status	String	见公用枚举表-退款状态枚举表	Y	WAIT_SELLER_AGREE
物流状态	logistics_status	String	见公用枚举表-物流状态枚举表	Y	INITIAL_STATUS
物流类型	logistics_type	String	见公用枚举表-物流类型枚举表	Y	VIRTUAL
物流费用	logistics_fee	Number(8, 2)	0.00~10000000.00	Y	10.00
物流支付类型	logistics_payment	String	见公用枚举表-物流支付类型枚举表	Y	SELLER_PAY
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	naaimin@yahoo.com.cn
卖家 ID	seller_id	String(30)		N	15464636
买家 ID	buyer_id	String(30)		N	123548874
买家 Email	buyer_email	String(100)		N	gaaimin0577@yahoo.com
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	2005-8-9 12:21:41

交易结束时间	gmt_close	Timestamp		Y	
物流状态更新时间	gmt_logistics_modify	Timestamp		Y	

#### 通知输入样例:

[http://10.2.5.100/api/apireceive/returnSuccess.php?notify\\_id=a6599156f863e49dcd74d7096ffd41c&seller\\_email=zxjl584@alibaba-inc.com&notify\\_type=trade\\_status\\_sync&payment\\_type=1&buyer\\_email=maaamin0577@yahoo.com.cn&sign=042b76d56f88edd34805dfc865b7cc28&trade\\_no=2008040202679041&quantity=3&total\\_fee=35.20&price=6.00&body=goodbody&out\\_trade\\_no=3145158185028077&notify\\_time=2008-04-0211:26:11&subject=good subject&receive\\_name=要要敏&trade\\_status=WAIT\\_SELLER\\_SEND\\_GOODS&sign\\_type=MD5](http://10.2.5.100/api/apireceive/returnSuccess.php?notify_id=a6599156f863e49dcd74d7096ffd41c&seller_email=zxjl584@alibaba-inc.com&notify_type=trade_status_sync&payment_type=1&buyer_email=maaamin0577@yahoo.com.cn&sign=042b76d56f88edd34805dfc865b7cc28&trade_no=2008040202679041&quantity=3&total_fee=35.20&price=6.00&body=goodbody&out_trade_no=3145158185028077&notify_time=2008-04-0211:26:11&subject=good subject&receive_name=要要敏&trade_status=WAIT_SELLER_SEND_GOODS&sign_type=MD5)

#### 通知触发条件

##### 1、交易状态改变时

##### 交易状态列表

状态代码	状态名称
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
TRADE_CLOSED	交易关闭

##### 2、退款状态改变时

##### 退款状态列表

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

##### 3、交易金额改变时

状态代码	状态名称
modify.tradeBase.totalFee	修改交易价格

#### 错误代码列表

错误代码	说明
USER_NOT_EXIST	用户不存在
ILLEGAL_SIGN	签名不正确
ILLEGAL_ARGUMENT	输入参数有错误

TRADE_SELLER_NOT_MATCH	卖家账户与交易中不一致
EXPARTNER_INFO_UNCORRECT	传入外部商户接口信息不正确
SELLER_NOT_IN_SPECIFIED_SELLERS	卖家不在指定的商户限制卖家中
TRADE_BUYER_NOT_MATCH	买家不匹配

## 1.25 注意事项

1. 合作伙伴通过“外部(纯)担保交易”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。**

2. 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

3. 该接口可以配置属性有:

属性编号	说明	默认值
support_self_timeout	是否支持自定义超时	false
support_commission	是否支持佣金	False
limit_seller_set	限制卖家集合	空

## 附录

### 1.2.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
------	-----------

## 签名及加密算法

### 1.2.1. 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=create_direct_pay_by_user32#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。

### 1.2.2. 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。

### 1.2.3. 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。

## 1.3.OpenSSL 命令

### 1.3.1. 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
```

## 1.3.2. 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
```

## 1.3.3. 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
```