



移动快捷支付接入与使用规则

移动快捷支付接口

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	3
4 接口使用规则	6
4.1 接入总流程	6
4.1.1 iOS	7
4.1.2 Android	8
4.2 通知规则	10
4.2.1 不可退款的移动快捷支付	10
4.2.2 可退款的移动快捷支付	10
5 测试流程规则	11
6 附录	11
6.1 如何获得 PID 与密钥	11
6.2 RSA 密钥生成与使用	14
6.2.1 生成商户密钥	14
6.2.2 RSA 密钥使用逻辑	18
6.3 业务数据传递	20

1 文档说明

1.1 文档说明

本文档是《移动快捷支付接口》的附录文档，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	手机客户端以字符串形式把需要传输的数据发送给支付宝的过程。
返回	支付宝以字符串形式直接把处理结果数据返回给手机客户端。
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。

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（移动快捷支付接口.pdf）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

类型	细则	原因
	必须保护合作者身份 ID、安全校验码 key、商户的公私钥、支付宝公钥的隐私性。	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	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。
组成	在 iOS 平台上使用移动快捷支付完成快捷支付。使用系统 UIApplication 的 openURL 方法来唤起移动快捷支付应用,并传递参数。	-(BOOL)openURL:(NSURL *)url
	在 Android 上使用移动快捷支付的 AlixPay 接口完成快捷支付。移动快捷支付实现为一个 Service,为其他应用提供支付服务。	String Pay(String info);
请求时签名逻辑	要请求的所有参数,需要根据参数名=参数值的格式,由字母 a 到 z 的顺序进行排序,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。	避免报异常错误,如:签名不正确。
	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。	避免报异常错误,如:签名不正确。
	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。	避免报异常错误,各种错误码需参考错误码列表。
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_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。而且只能设置其值为 utf-8,即本产品不支持 GBK 编码格式。	避免报异常错误,如:签名不正确。
	seller 是收款时的支付宝账号,需要与 partner 对应的支付宝账号为同一个,也就是说收款支付宝账号必须是签约时的支付宝账号。	避免签约支付宝账号出现资金受损的可能
	签名方式仅支持 RSA	避免签名不成功

类型	细则	原因
pkcs8 编码	移动快捷支付要求商户私钥需要做 pkcs8 编码以支持更高手机系统版本，php 服务器可不需要做。	php 不支持 pkcs8
接口结构	服务端：用于生成提交参数，以及处理支付宝的异步通知返回。	移动快捷支付由服务端和客户端构成，为了交互信息安全通常把所需参数放在服务端，当客户端有需要时去服务端获取。
	客户端：构建表单参数提交到支付宝。	
	支付参数提交时，需要组装订单信息 orderInfo，其中参数以 key="value" 形式呈现，参数之间以 "&" 分割，获取 Alipay 支付对象调用支付，此接口方法实现为同步调用，将阻塞商户应用 UI 线程，所以调用此接口需启动新线程，并使用 looper 为 main looper 的 Handler 对象与 UI 线程传递消息。	避免请求支付宝时报错，错误码为签名不正确。
数据传输	必须使用 https 协议	避免接口无法正常使用
通知返回验证	移动快捷支付接口的服务器异步通知中，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，先对这些参数根据“参数名=参数值”的格式，由字母 a 到 z 的顺序进行排序，再依照“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	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 7 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。
	在服务器异步通知页面文件中，需保证商户的所有业务全部运行完成，才能执行打印 success 的动作。	避免异步通知不正常，如收不到通知或业务处理没有完成却告诉支付宝系统已经处理完成。
	建议每一次业务操作需以日志形式记录到商户网站的日志操作数据库中，做好通知重复判断机制。	用来在必要时检查或跟踪业务处理情况
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

4.1 接入总流程

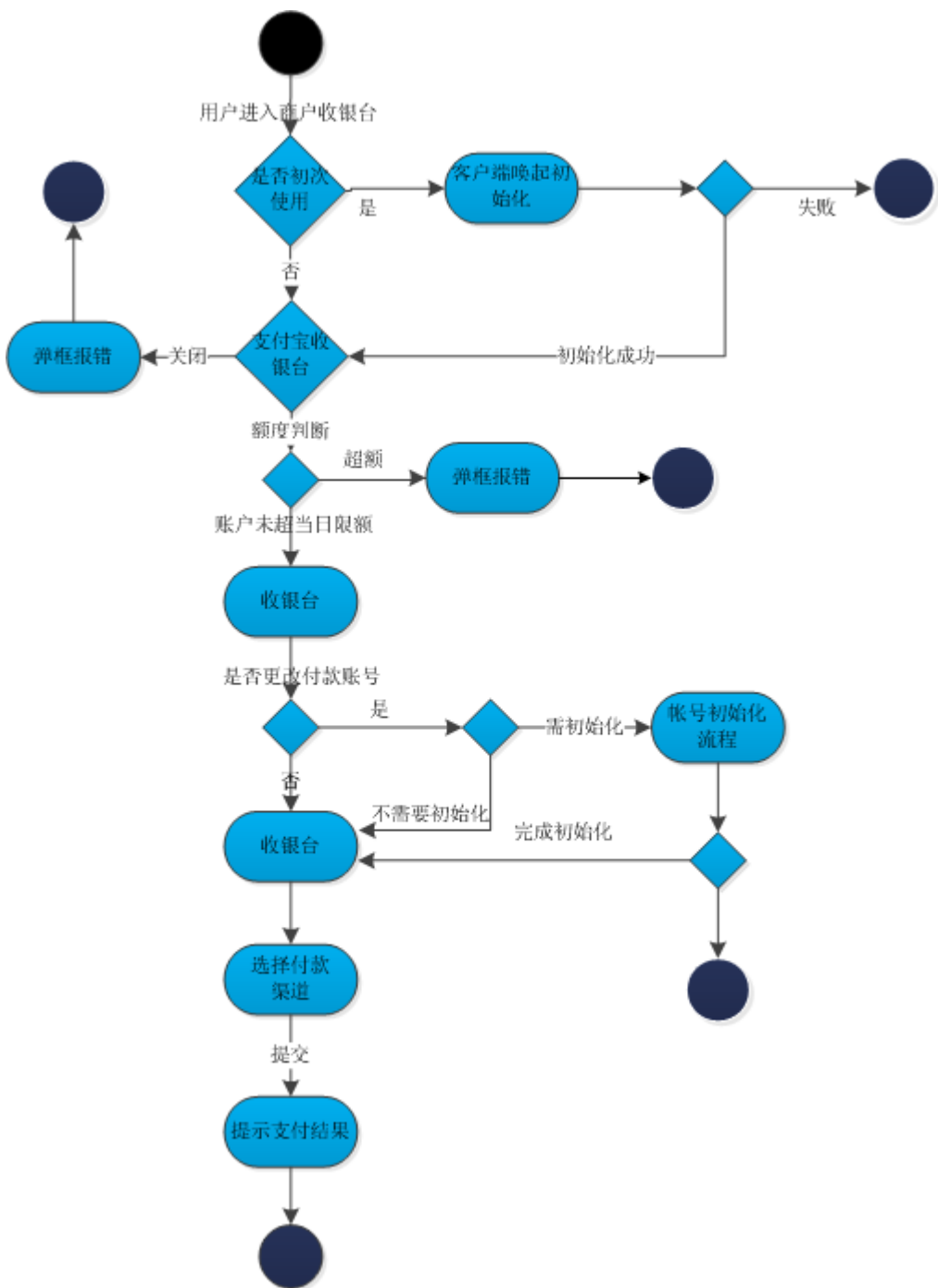


图4-1 移动快捷支付接入总流程

4.1.1 iOS

- 标准模式

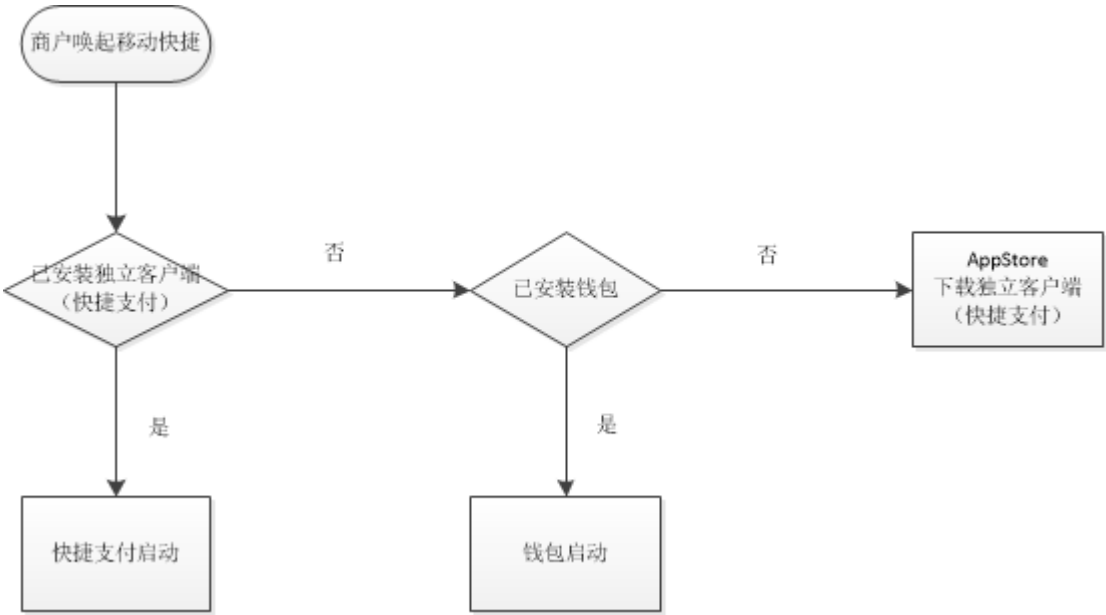


图4-2 标准模式流程

- 卡券接入类商户

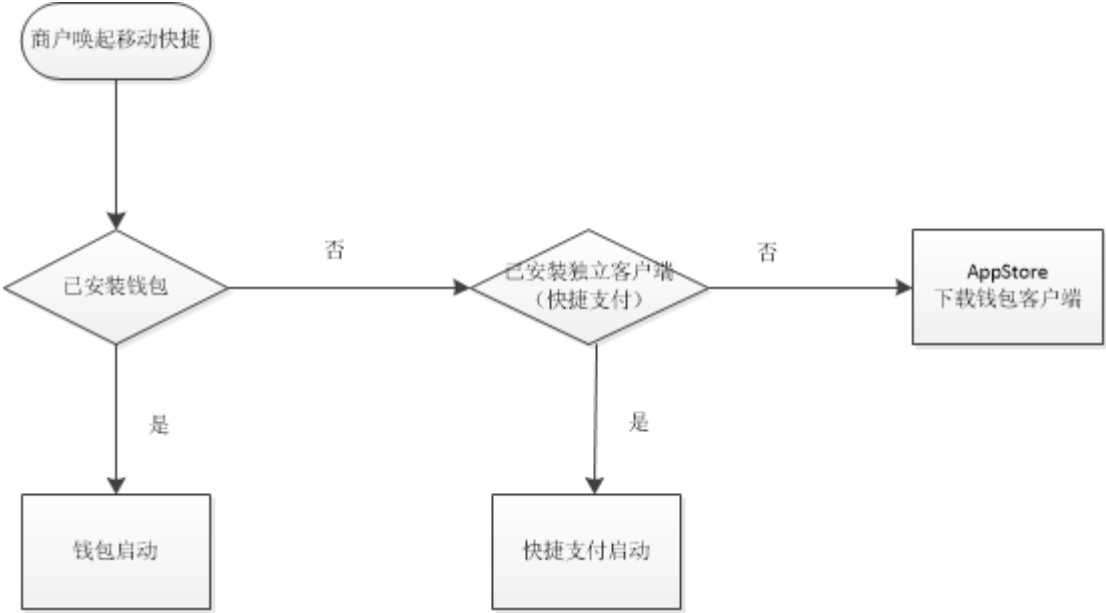


图4-3 卡券接入类商户流程

判断逻辑:

- 如果钱包和快捷插件都有，调用快捷支付（标准模式）。
优先调用移动快捷支付的独立客户端完成支付。

- 如果钱包和快捷插件都没有，推荐下载移动快捷支付的独立客户端。
是否有客户端：
 - 有：调用移动快捷支付的独立客户端完成支付；
 - 无：调用支付宝钱包完成支付。



注意：

对于特殊的需要同步卡券的行业，需要推荐下载钱包。

4.1.2 Android

先检测商户客户端中的移动快捷支付服务 **apk** 是否是最新的：

- 是：则直接安装支付；
- 否：则从约定的服务器下载安装包，安装并完成支付。

标准流程模式如下图所示。

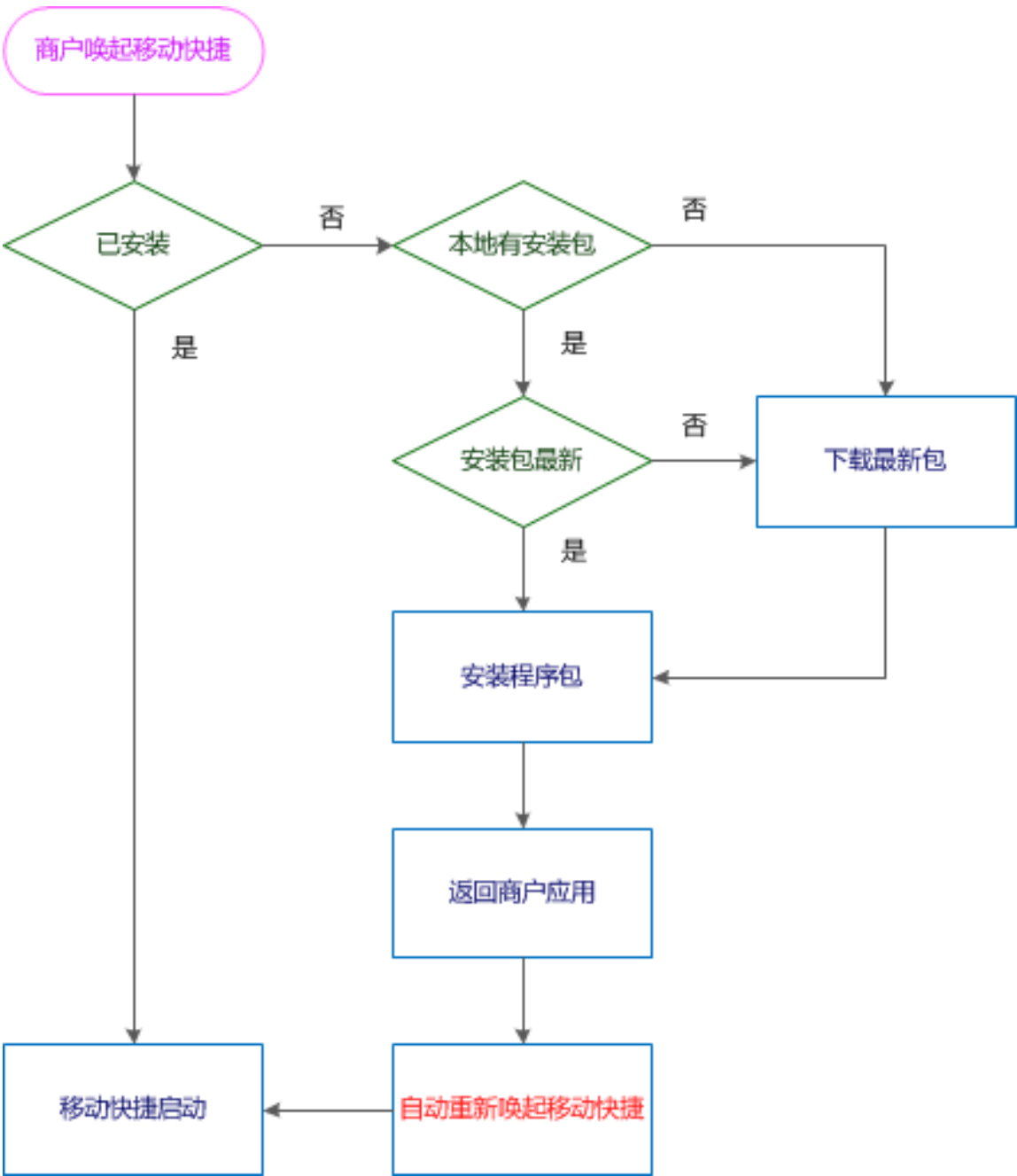


图4-4 Andriod 标准流程

支付宝为第三方应用客户端提供了 3 种集成移动快捷支付服务的方式：

- **A 方案：**将移动快捷支付服务安装包 apk 与第三方应用客户端整合在一起，根据不同的应用场景由第三方应用客户端释放并安装移动快捷支付安装包 apk；
- **B 方案：**动态下载安装，移动快捷支付服务安装包 apk 预先存放在预定的远程服务器中，第三方应用可以从此处下载 apk 并进行安装；

- **C 方案：**也可以唤起钱包支付，商户可针对支付宝的钱包包名（com.eg.android.AlipayGphone.IAlipay）判断，如果 versioncode 大于等于 37，那么可以唤起钱包支付。

为避免捆绑安装旧版本的移动快捷支付服务后，接着又需要重新升级并安装最新版本，支付宝推荐使用 A 方案。在最新的 demo 中，会首先链接支付宝服务器，检查并判断捆绑在第三方应用客户端中的安装包是否为最新版本，如果是，则直接安装；如果捆绑在第三方应用客户端中的安装包不是最新版本，则从服务器中下载最新版本的安装包，然后进行安装。

4.2 通知规则

4.2.1 不可退款的移动快捷支付

移动快捷支付的异步通知存在两个通知交易状态（trade_status）。

- 第一个状态值是 WAIT_BUYER_PAY：表示等待付款，商户可根据自己的业务逻辑需求做相应操作，处理完业务逻辑后须返回 SUCCESS 字符串给支付宝；
- 第二个状态值是 TRADE_FINISHED：表示交易成功完成，此状态表示该笔订单支付宝端已经支付成功，商户根据此状态做相应的业务逻辑操作，最后同样需返回 SUCCESS 字符串给支付宝。

4.2.2 可退款的移动快捷支付

可退款的异步通知与不可退款的机制一致，第一个状态（WAIT_BUYER_PAY）相同，第二个状态为 TRADE_SUCCESS，在这个状态下商户可做相应业务逻辑操作，并返回 SUCCESS。第三个状态为 TRADE_FINISHED，表示订单完结不可再退款。是否退手续费通知判断：

- 退手续费：单笔交易完成退款操作支付宝异步通知发送 TRADE_CLOSE 状态（交易关闭），此交易状态需支付宝后台配置单独开启，默认不开；
- 不退手续费：单笔交易完成退款操作支付宝异步通知发送 TRADE_SUCCESS 状态，并在订单完成支付后的三个月发送 TRADE_FINISHED 状态。

5 测试流程规则

表5-1 测试流程规则

步骤	调试内容	备注
第一步: 在本机单独对这个接口进行调试	<ul style="list-style-type: none"> 正常获取授权令牌 正常唤起客户端支付 	仅仅把接口配置好, 不要放在商户的正式 app 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"> 正常获取授权令牌 正常唤起客户端支付 alipay 同步返回 服务器异步通知返回 	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 app 项目中	无	把调试好的接口与商户 app 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 app 项目进行调试	<ul style="list-style-type: none"> 整个业务操作流程 正常唤起客户端支付 alipay 同步返回 服务器异步通知返回 业务逻辑后续的执行 	在本机调试衔接到 app 项目, 并做好客户端和服务端的区分后的接口。



注意:

使用一个账号做测试的时候, 第一次走支付流程会出现授权页面, 在设置过免密额度以及短信校验后不会再出现授权页面。

6 附录

6.1 如何获得PID与密钥

步骤1: 使用签约支付宝账号登录“商家服务”平台中的“我的商家服务” (<https://b.alipay.com/order/serviceIndex.htm>)。



图6-1 我的商家服务

步骤2: 在地址栏中输入https://mobiless.alipay.com/myproduct/my_order.htm，即可查看到签约支付宝账号、合作者身份ID（PID）的信息。



图6-2 查看签约支付宝账号等信息

步骤3: 点击<密钥管理>按钮，输入“支付密码”，点击“确认”，即可查看到各类型的私钥。



图6-3 查看私钥

步骤4: 上传 RSA 类型的商户公钥，获得支付宝公钥。



图6-4 获取支付宝公钥

步骤5: 保存支付宝账号、合作者身份 ID（PID）、交易安全校验码（MD5）、交易安全校验码（RSA）中的支付宝公钥。

6.2 RSA密钥生成与使用

6.2.1 生成商户密钥

1. 打开openssl密钥生成软件

打开 openssl 文件夹下的 bin 文件夹，执行 openssl.exe 文件，如下图：

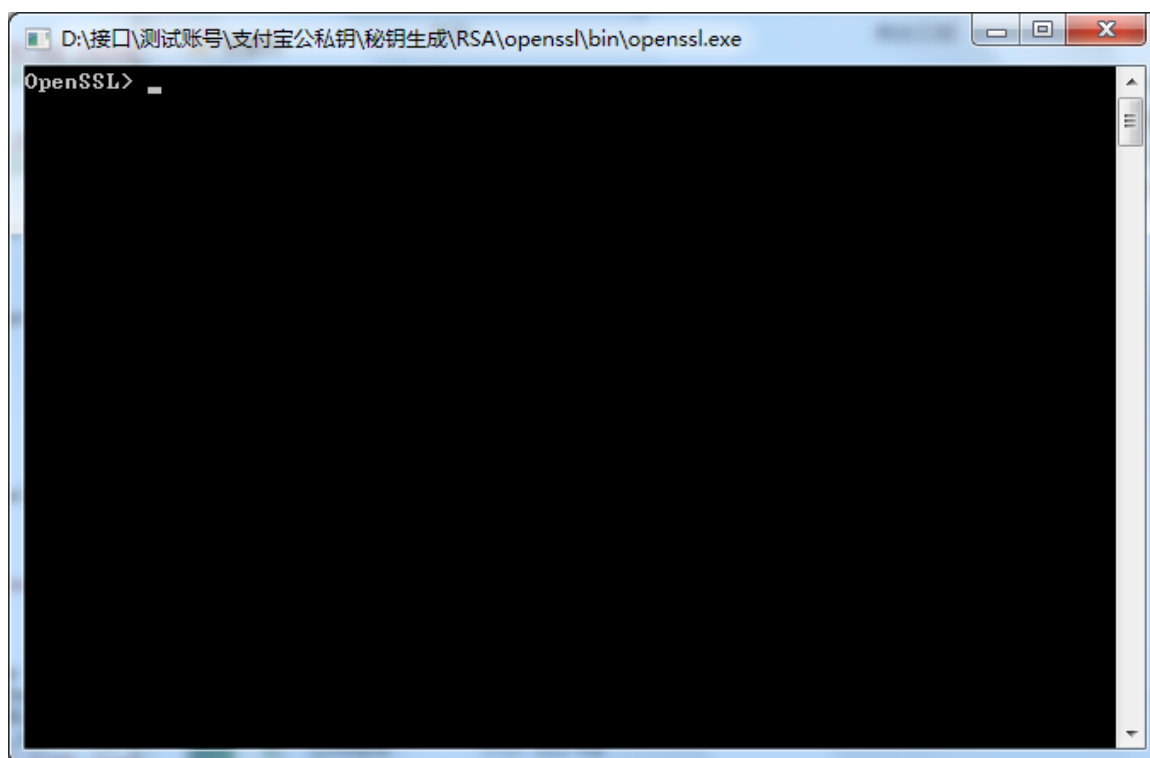


图6-5 执行 openssl.exe 文件

2. 生成RSA私钥

输入“**genrsa -out rsa_private_key.pem 1024**”命令，回车后，在当前 bin 文件目录中会新增一个 rsa_private_key.pem 文件，其文件为原始的商户私钥（**请妥善保管该文件**，PHP 开发语言中需要使用该文件），以下为命令正确执行截图：

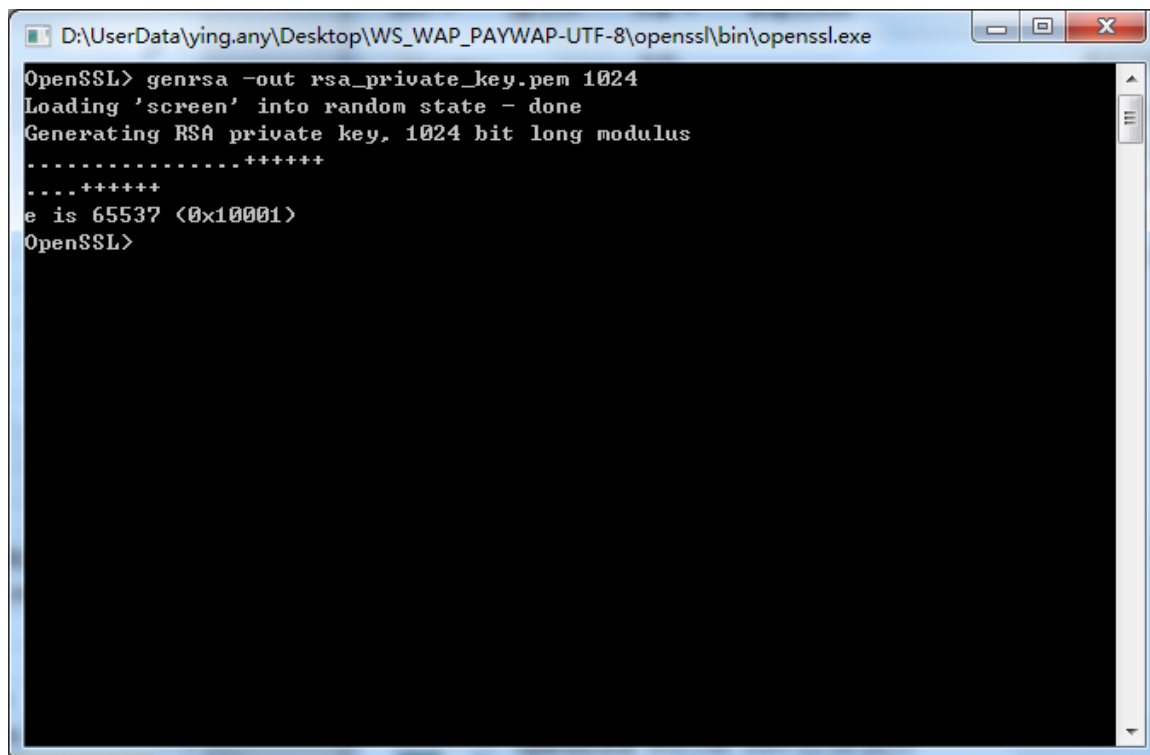
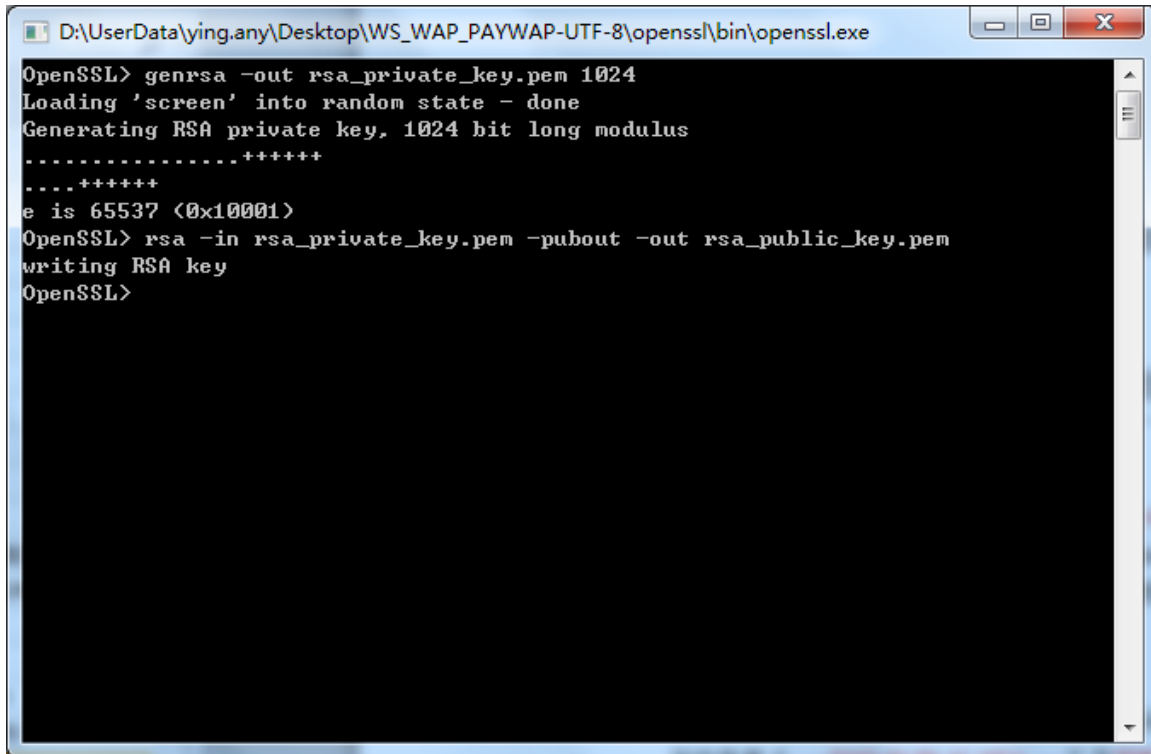


图6-6 生成 RSA 私钥

3. 生成RSA公钥

输入“*rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem*”命令回车后，在当前 bin 文件目录中会新增一个 *rsa_public_key.pem* 文件，其文件为原始的商户公钥（请妥善保存该文件，PHP 开发语言中需要使用该文件），以下为命令正确执行截图：

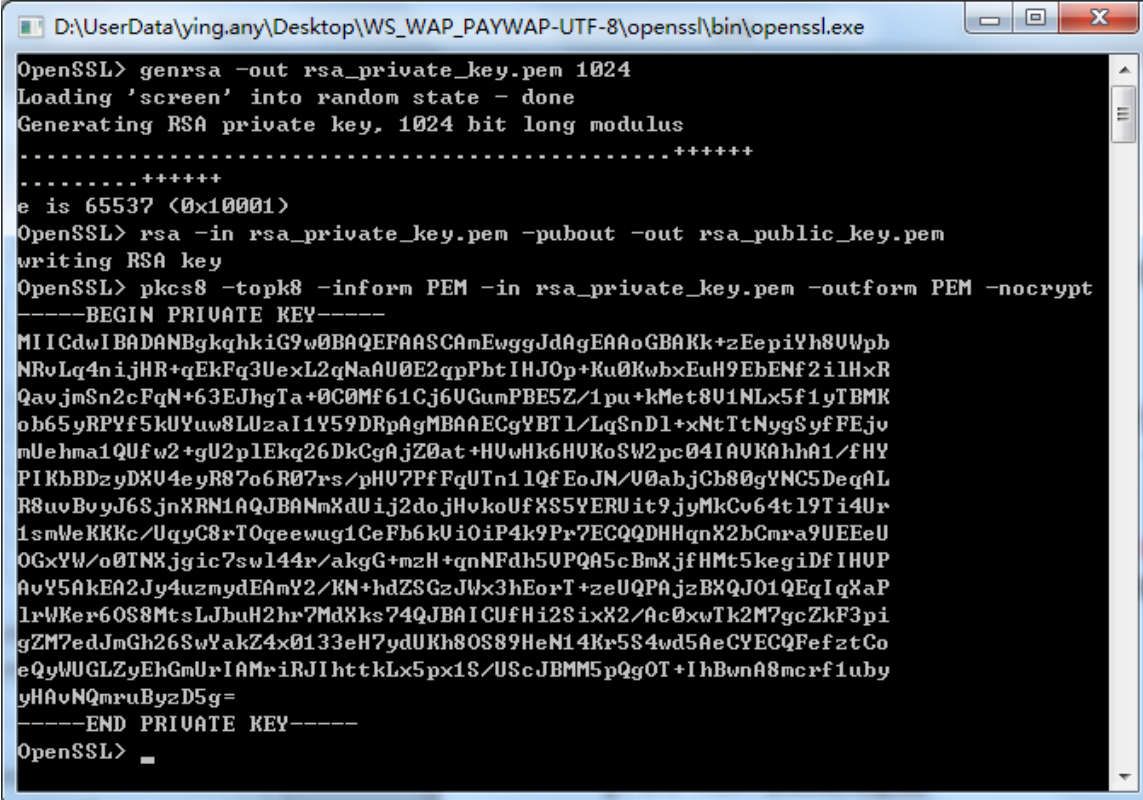


```
D:\UserData\ying.any\Desktop\WS_WAP_PAYWAP-UTF-8\openssl\bin\openssl.exe
OpenSSL> genrsa -out rsa_private_key.pem 1024
Loading 'screen' into random state - done
Generating RSA private key, 1024 bit long modulus
.....+++++
....+++++
e is 65537 (0x10001)
OpenSSL> rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem
writing RSA key
OpenSSL>
```

图6-7 生成 RSA 公钥

4. 生成PKCS8 编码的私钥

输入命令 “*pkcs8 -topk8 -inform PEM -in rsa_private_key.pem -outform PEM -nocrypt*” 并回车，当前界面中会直接显示出生成结果：



```
D:\UserData\ying.any\Desktop\WS_WAP_PAYWAP-UTF-8\openssl\bin\openssl.exe
OpenSSL> genrsa -out rsa_private_key.pem 1024
Loading 'screen' into random state - done
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
OpenSSL> rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem
writing RSA key
OpenSSL> pkcs8 -topk8 -inform PEM -in rsa_private_key.pem -outform PEM -nocrypt
-----BEGIN PRIVATE KEY-----
MIICGwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBAKk+zEepiYh8UWpb
NRvLq4n iJHR+qEkFq3UexL2qNaAU0E2qpPbtIHJO p+Ku0KwbxEuH9EbENf2ilHxR
QavjmSn2cFqN+63EJhgTa+0C0MF61Cj6UGumPBE5Z/1pu+kMet8U1NLx5f1yTBMK
ob65yRPyf5kUYuw8LUzaI1Y59DRpAgMBAACgYBT1/LqSnD1+xNtTtNygsyfFEjv
mUehma1QUfw2+gU2p1Ekq26DkCgA jZ0at+HUwHk6HUkoSW2pc04IAUkAhha1/fHY
PIKbBDzyDXU4eyR87o6R07rs/pHU7PfFqUTn1lQfEoJN/U0abjCb80gYNC5DeqAL
R8uvBoyJ6SjnXRN1AQJBANmXdUij2dojHvkoUfXS5YERUit9jyMkCv64t19Ti4Ur
ismWeKKKc/UqyC8rT0qeewug1CeFb6kViOiP4k9Pr7ECQQDHHqnX2bCmra9UEEeU
OGxYW/o0TNXjgic7swl44r/akgG+mzH+qnNFDh5UPQA5cBmXjfhMt5kegidfIHUP
AvY5AkEA2Jy4uzmydEAmY2/KN+hdZSGzJWx3hEorT+zeUQPAjzBXQJO1QEgIqXaP
lrWker60S8MtsLJbuH2hr7MdXks74QJBAlCUfHi2S ixX2/Ac0xwTk2M7gcZkF3pi
gZM7edJmGh26SwYakZ4x0133eH7ydUKh80S89HeN14Kr5S4wd5AeCYECQFefztCo
eQyWUGLZyEhGmUrIAMr iRJlhttkLx5px1S/UScJBMM5pQgOT+IhBwnA8mcrf1uby
yHA0vNQmruByzD5g=
-----END PRIVATE KEY-----
OpenSSL>
```

图6-8 生成 PKCS8 编码的私钥

右键点击 openssl 窗口上边边缘，选择“编辑→标记”，选中要复制的文字：

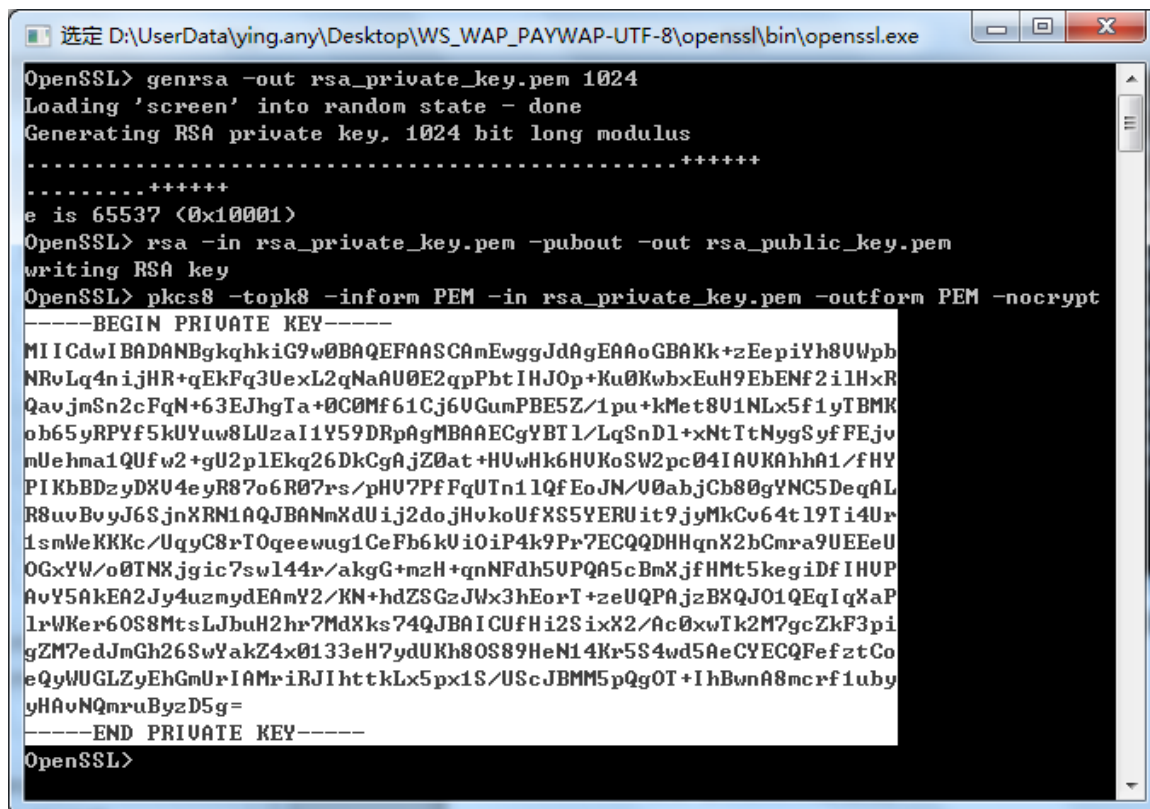


图6-9 选中要复制的文字

此时继续右键点击 openssl 窗口上边边缘，选择“编辑→复制”，把复制的内容粘贴进一个新的记事本中，可随意命名，只要知道这个是 PKCS8 格式的私钥即可（[请妥善保存该文件](#)）。

6.2.2 RSA密钥使用逻辑

RSA 密钥使用逻辑：

商户在使用 RSA 签名方式的支付宝接口时，真正会用到的密钥是商户私钥与支付宝公钥。商户上传公钥给支付宝，支付宝把公钥给商户，是公钥互换的操作。这就使得商户使用自己的私钥做签名时，支付宝端会根据商户上传的公钥做验证签名。商户使用支付宝公钥做验证签名时，同理，也是因为支付宝用支付宝私钥做了签名。

1. PHP开发语言使用方法

key 文件夹里面须存放.pem 后缀名的商户私钥、支付宝公钥两个文件。

- 商户的私钥
 - 必须保证只有一行文字，即：没有回车、换行、空格等；
 - 不需要对刚生成的（原始的）私钥做 pkcs8 编码；

- 不需要去掉去掉“-----BEGIN PUBLIC KEY-----”、“-----END PUBLIC KEY-----”；
 - 简言之，只要维持刚生成出来的私钥的内容即可。
- 支付宝公钥
 - (1) 先用签约支付宝账号登录支付宝网站，再浏览器访问“<https://ms.alipay.com/index.htm>”，点击菜单栏“我的商家服务”，右侧点击“密钥管理”进入密钥管理页面，复制“交易安全校验码（RSA）”→“支付宝公钥”栏目下的整串信息到一个新的记事本中，命名为“alipay_public_key.txt”；
 - (2) 去掉这串字符串中的回车、换行、空格，变成只有一行文字；
 - (3) 在这串支付宝公钥字符串的头尾部分，分别增加“-----BEGIN PUBLIC KEY-----”、“-----END PUBLIC KEY-----”这两条文字；
 - (4) 切割这串支付宝公钥字符串，切割后的格式与商户刚生成的公钥格式一致即可，如下图：

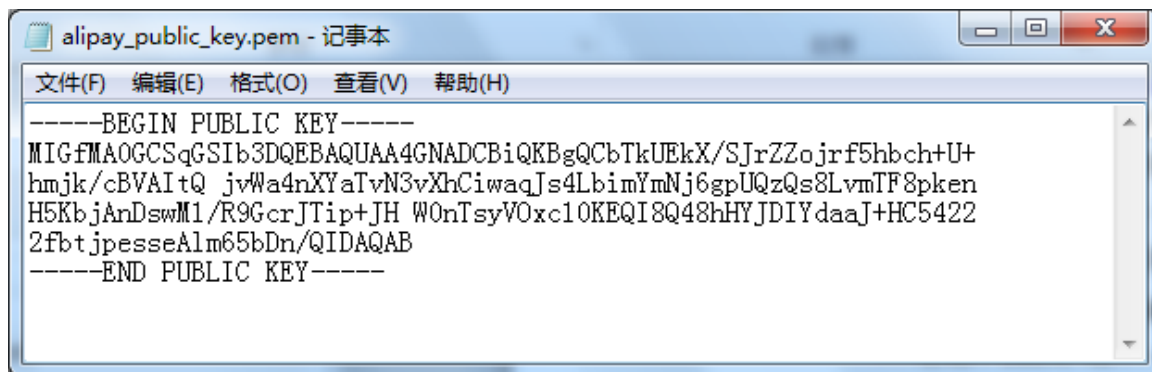


图6-10 支付宝公钥示意图

- (5) 保存该记事本，并改变后缀名为.pem。

2. JAVA和ASP.NET(C#)开发语言使用方法

- 商户的私钥
 - 必须保证只有一行文字，即：没有回车、换行、空格等；
 - 需对刚生成的（原始的）私钥做 pkcs8 编码；
 - 编码完成后，复制该段私钥，并去掉该段里面的回车、换行、空格、“-----BEGIN RSA PRIVATE KEY-----”、“-----END RSA PRIVATE KEY-----”。
- 支付宝公钥

- (1) 先用签约支付宝账号登录支付宝网站，再浏览器访问“<https://ms.alipay.com/index.htm>”，点击菜单栏“我的商家服务”，右侧点击“密钥管理”进入密钥管理页面，复制“交易安全校验码（RSA）”→“支付宝公钥”栏目下的整串信息；
- (2) 去掉这串字符串中的回车、换行、空格，必须保证只有一行文字。

6.3 业务数据传递

支付宝提供的业务参数为支付宝需要商户传递过来的数据要求。商户只需要根据自己的业务需求，在业务逻辑代码运行时把这些动态数据以赋值给变量的形式，再通过支付宝接口本身的接口逻辑，传递给支付宝系统，让支付宝系统可识别。

举例说明，商户要把某笔订单的数据传递给支付宝。那么商户需要先根据支付宝的参数要求，从自己的下单系统中拿到付款总金额（total_fee）、商户的订单号（out_trade_no）、订单名称（subject）等数据，再把这些数据一个一个以值的形式赋给对应的变量。再通过代码逻辑，把变量组合及加工成一次可以发送给支付宝的请求。