手机网站支付开发指南 JAVA

文件版本：2.1.0

2012-07-20

前言

1. 面向读者

本文档主要面向需要接入支付宝手机网站支付的商户的开发人员。

2. 读者所需技能

读者需有 JAVA EE 开发背景

3. 开发环境要求

OS: Windows XP 或以上

JDK:JDK1.6 或以上

目录

第一章 手机网站支付服务简介

当商户 WAP 网站具备收费场景时,通过手机网站支付服务完成支付过程。

1.1 WAP 支付服务

WAP 支付服务是指用户使用移动终端通过 WAP 方式访问支付宝手机网站完成支付的过程。WAP 支付支持 https/http 作为接口调用传输协议，以满足不同手机能正常访问，与此同时采用常见的请求/响应模式和第三方系统进行对接，降低了接入难度。可以进行广泛接入。

第二章 接入流程

2.1 接入前期准备

接入前期准备工作包括**商户签约**和**密钥配置**，已完成商户可略过。

2.1.1 商户签约

首先，商户需要在 https://ms.alipay.com 进行注册，并签约手机网站支付服务。签约成功后可获取支付宝分配的合作商户 ID(PartnerID)，账户 ID(SellerID)，如图：

图 2-1 商户 ID 获取示意图

签约过程中需要任何帮助请致电： 0571-88158090（支付宝商户服务专线）

2.1.2 密钥配置

签约成功后，商户可登陆 https://ms.alipay.com 获取商户账号对应的支付宝公钥和 MD5 key，具体获取步骤请见 3.2 RSA 密钥详解以及 3.4.2 MD5 key

接着，商户生成商户公钥和商户私钥（具体生成步骤请见 3.2 RSA 密钥详解），并登陆 https://ms.alipay.com，上传商户公钥（具体上传步骤请见 3.2 RSA 密钥详解）。

至此，接入前期准备工作完成，下一节将使用 demo 测试准备工作是否正确。

2.2 Demo

为了便于商户的接入，我们提供了 demo。通过本 demo，商户可测试 2.1 节的前期准备工作是否正确完成，同时还可参考 demo 的代码完成接入。本文档以 Java 版为例。

2.2.1 Demo 配置运行

步骤 1：

解压下载的压缩包 *WS_WAP_PAYWAP* ，进入目录“\WS_WAP_PAYWAP\WAP 支付 demo(JAVA 版)”，可以见到“*wapPayDemoMD5.rar*”和“*wapPayDemoRSA.rar*”两个压缩文件，两个 demo 的区别在于和支付宝通信时采用的加密和签名方法不同，以下以 RSA 版的为例。解压“*wapPayDemoRSA.rar*”，导入 eclipse，项目结构如图：

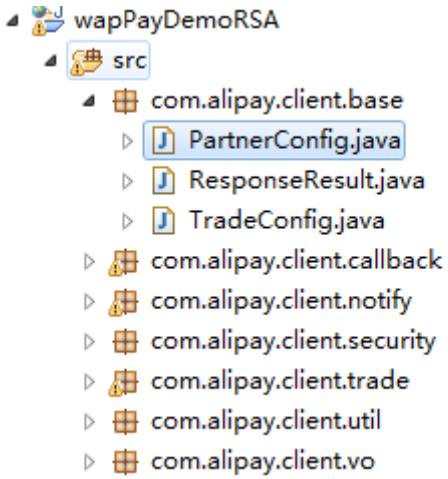

图 2- 2 Demo 项目结构图

步骤 2：

打开“*PartnerConfig.java*”文件，按照注释添加商户账号信息，具体包括：**合作商户 ID、支付宝账户、支付宝公钥（即服务器公钥）、商户私钥、商户 MD5 key**。如下图：

```
package com.alipay.client.base;

public class PartnerConfig {
    //合作商户ID。用签约支付宝账号登录ms.alipay.com后，在账户信息页面获取。
    public static final String PARTNER = "";
    //商户收款的支付宝账号
    public static final String SELLER = "";
    // 商户（RSA）私钥
    public static final String RSA_PRIVATE = "";
    // 支付宝（RSA）公钥 用签约支付宝账号登录ms.alipay.com后，在密钥管理页面获取。
    public static final String RSA_ALIPAY_PUBLIC ="";

    // 商户（MD5）KEY
    public static final String KEY="";
}
```

图 2-3 商户信息配置

步骤 3：

使用 tomcat 部署运行，浏览器访问 *http://localhost:8080/wapPayDemoRSA /*（注意，浏览器需要更改 User Agent 为移动设备，可以在 firefox 浏览器下使用插件 User Agent Switcher 更改）。如下图所示：

图 2-4 选择商品

点击“购买”，页面将跳转到支付宝收银台，进入支付环节。如下图：

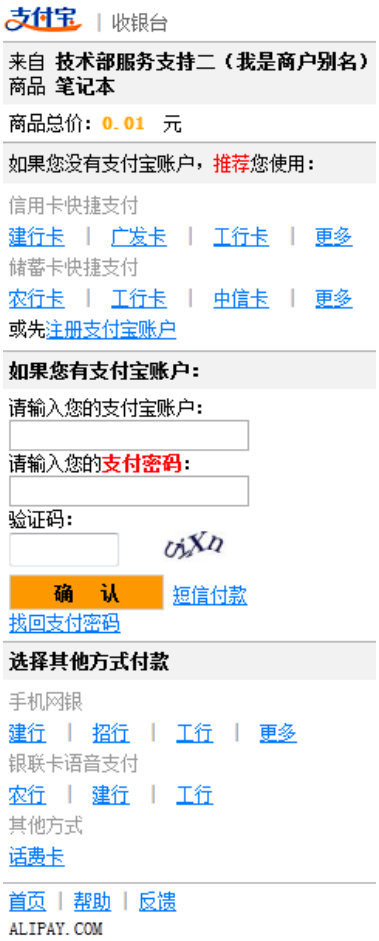

图 2-5 支付页面

注意：付款用的账户应为某一真实账户，且交易金额也是真实的！）

至此，说明 2.1 节中的准备工作没有错误，账户信息配置正确，可进行开发工作。

2.2.2 Demo 结构说明

以下为 Demo 中较为重要或参考价值较大的类的说明：

类或包名	说明
com.alipay.client.base.PartnerConfig	商户账户信息配置类，包括账号公私钥等信息
com.alipay.client.callback.CallBack	模拟对支付成功后支付宝的调用请求的处理
com.alipay.client.notify.NotifyReceiver	支付宝服务器通知的接收和处理
com.alipay.client.security	MD5 和 RSA 签名验签相关方法
com.alipay.client.trade.Trade	Demo 主流程
com.alipay.client.util	基础类，包含 base64 编码和 xml 解析等

2.3 开发

整个开发概括起来，即是用户发起购买后商户生成订单信息再顺序调用支付宝提供的接口的过程，本节将按顺序对所有接口的调用作详细介绍。

2.3.1 创建交易并获取 token

通过向支付宝的交易创建接口 *alipay.wap.trade.create.direct* 发送订单信息，商户可以创建交易并获取相应的 token。为了展示的方便，以下请求样例使用 *GET* 方式，但正式请求时务必使用 *POST*，否则可能由于请求信息过长而导致内容丢失。调用时的签名方式支持 RSA 和 MD5。以下为请求样例：

以下样例分为三类参数：(下划线参数为最外层参数，共 8 个)

蓝色参数---- 表示该参数为必传，值可自定义。（详见参数列表）

红色参数---- 表示该参数非必传。（详见参数列表）

绿色参数---- 表示该参数为必传，并且参数值须和示例一致。（详见参数列表）

样例

http://wappaygw.alipay.com/service/rest.htm?req_data=<direct_trade_create_req><subject>商户收银台测试</subject><out_trade_no>1282889603601</out_trade_no><total_fee>1</total_fee><seller_account_name>chenf003@yahoo.cn</seller_account_name><call_back_url>http://www.alipay.com/waptest0504/servlet/CallBack</call_back_url><notify_url>http://www.alipay.com/waptest0504/servlet/NotifyReceiver</notify_url><out_user>outID123</out_user><merchant_url>http://www.alipay.com</merchant_url><pay_expire>10</pay_expire></direct_trade_create_req>&service=alipay.wap.trade.create.direct&sec_id=0001&partner=2088101000137799&req_id=1282889689836&sign=VRVr7adPfsHblFjiBkGWryhKIKt+CaI4Cq2MA2wG1ENVuBAyFDlp3FbttndmID0USlfn22a9/6fQ+X+KPDE09hcTNz3gJ1edUiDWxHXY/ahTexCP79SDtoHx29uepXsHBe32DP0k9jZbfhpT8Ly0+ksuo5VJO0iymxQ87hQPjJw=&format=xml&v=2.0

如样例所示，8 个外层参数中，只有参数 *req_data* 包含了 XML 标签，XML 标签的具体含义请见详见参数列表。sign 值的生成请见 3.3.2 RSA 签名 或 3.4.3 MD5 签名和验签

1）成功返回样例：

请求结果判断：若返回的参数中包含 *res_data* 则说明成功，反之则失败。

当商户使用 RSA 签名方式时，实际返回的内容如下，其中 res_data 参数值为加密内容，商户需用商户 RSA 私钥先进行解密后再验签。具体请见 3.3 RSA 解密、签名和验签

样例

res_data=Cl2Mm1Z2YILG8oYe8%2FngEAvYSM9YYmcqUqLtUCZ10habqYb6poowofjzVG3nsUJY6qlgnRrq%2FxFTtdLdwBDGltV8rwpf1AFB01ydCanpQoFgQg%2Brt79JRQ%2B9CC3E%2Fg148C4F95eJ1FNf0L6taXaMFwxaxrvTAdDHzzvS

igy3%2BaKdFh8z2K1Zs4gm2bD39lR1CRXSipOyVFhCZZR9L9N8tQNZbDqnyBu%2FjLdLbvXvEuE4fImZPPbsALecVCvs HL4iKFrquPnhA4Zz%2FZEM%2FoJghXA6xlAO0a1d0h6Os%2Fd83mvDPfmhs3oVjPX3FsXCL18Dg4mdzj3gWllbqLnwa mM94g%3D%3D&service=alipay.wap.trade.create.direct&sec_id=0001&partner=2088201747196380&req_id=12 88337908547&sign=RiyyndPEei2QQc%2FHt1%2FIrmYyW6%2FFKNZFxPUiOcXndAOo3OifNRshRjaLlwEs3d2pBpbm yclfooF7tctFdXcrSM584wgsY%2Bj2o0Z6dXst9lmz%2F4OD%2BL2ubk1DXoLWau0f5NiteIuGqGDWUdXMKRLx1FJ0f %2FMN8GOCUZYN15%2FUE%2FE%3D&v=2.0

当商户使用MD5签名方式时，实际返回的内容如下（其中res_data参数值为明文内容，无需解密，直接验签即可）

样例

partner=2088101000137799&req_id=1283133204160&res_data=<?xml version="1.0" encoding="utf-8"?><direc t_trade_create_res><request_token>20100830e8085e3e0868a466b822350ede5886e8</request_token></direct _trade_create_res>&sec_id=MD5&service=alipay.wap.trade.create.direct&v=2.0&sign=72a64fb63f0b54f96b10ce fb69319e8a

由此，即可获得 token，如上例中：

<request_token>20100830e8085e3e0868a466b822350ede5886e8</request_token>

2）失败返回样例：

失败返回无论哪种签名方式，内容都是明文无需解密，也不必验签。

样例

partner=208810100013779&req_id=1283133132946&res_error=<?xml version="1.0" encoding="utf-8"?><err><code>0005</code><sub_code>0005</sub_code><msg>partner illegal</msg><detail>合作伙伴没有开通接口访问权限</detail></err>&sec_id=0001&service=alipay.wap.trade.create.direct&v=2.0

具体的错误信息，可以根据 res_error 的查阅错误代码表获取。

以下以 demo 源码为例说明主要步骤：

```
详细代码请参考类com.alipay.client.trade.Trade
Map<String, String> reqParams = prepareTradeRequestParamsMap(request);
        //签名类型
        String signAlgo = SEC_ID;
        String reqUrl = TradeConfig.REQ_URL;
        //签名
        String sign = sign(reqParams,signAlgo,PartnerConfig.RSA_PRIVATE);
        reqParams.put("sign", sign);
        ResponseResult resResult = new ResponseResult();
        String businessResult = "";
        try {
            resResult = send(reqParams,reqUrl,signAlgo);
        } catch (Exception e1) {
            e1.printStackTrace();
        }
```

```
        if (resResult.isSuccess()) {
            //对获取的结果进行解析
            businessResult = resResult.getBusinessResult();
        } else {
            return;
        }
                            .getBytes("UTF-8")));
        } catch (UnsupportedEncodingException e) {
        } catch (Exception e) {
        }
        // 开放平台返回的内容中取出request_token
        String requestToken = directTradeCreateRes.getRequestToken();
        Map<String, String> authParams = prepareAuthParamsMap(request,
                requestToken);
        //对调用授权请求数据签名
        String authSign =
sign(authParams,signAlgo,PartnerConfig.RSA_PRIVATE);
        authParams.put("sign", authSign);
        String redirectURL = "";
        try {
            redirectURL = getRedirectUrl(authParams,reqUrl);
        } catch (Exception e) {
            e.printStackTrace();
        }
        if (StringUtil.isNotBlank(redirectURL)) {
            response.sendRedirect(redirectURL);
            return;
        }
```

获取 token 后，可通过 WAP 支付（跳转到支付宝收银台）进入支付环节，具体步骤见下节。

2.3.2 授权并执行

通过调用向支付宝“授权并执行”接口，并向该接口传递 2.3.2 节中获取的 token，从而获取跳转到支付宝收银台的 URL 并使用户页面跳转。以下为 GET 形式的请求样例：

以下样例分为三类参数(下划线参数为最外层参数，共 8 个。只有 req_data 参数值中包含内层 xml 标

签参数。)：

蓝色参数---- 表示该参数为必传，值可自定义。（详见参数列表）

红色参数---- 表示该参数非必传。（详见参数列表）

绿色参数---- 表示该参数为必传，并且参数值须和示例一致。（详见参数列表）

样例（如用 GET 方式请求需要 URL Encode）：

http://wappaygw.alipay.com/service/rest.htm?req_data=<auth_and_execute_req><request_token>201008309e298cf01c58146274208eda1e4cdf2b</request_token></auth_and_execute_req>&service=alipay.wap.auth.authAndExecute&sec_id=0001&partner=2088101000137799&sign=LdXbwMLug8E4UjfJMuYv2KoD5X5F3vHGQsQbZ/rdEQ3eaN4FPal7rhsbZZ/+ZUL1kAKzTQSDdMk87MEWtWO1Yq6rhnt2Tv8Hh6Hb16211VXKgbBCpq861+LopRwegPbGStcwBuAyE4pi6fYlJ6gxzL4tMyeLe+T5XZ0RKRUk00U=&format=xml&v=2.0

请求成功或失败均会得到跳转到支付宝收银台的地址，以下为跳转后的页面

1）成功返回样例：

图 2-6 支付宝 WAP 收银台跳转成功

2）失败返回样例：

图 2-7 支付宝 WAP 收银台跳转失败

以下以 demo 源码为例说明主要步骤：

```
    详细代码请参考类com.alipay.client.trade.Trade
// 开放平台返回的内容中取出request_token
        String requestToken = directTradeCreateRes.getRequestToken();
        System.out.println("Token:"+requestToken);
        Map<String, String> authParams = prepareAuthParamsMap(request,
                requestToken);
        //对调用授权请求数据签名
        String authSign =
sign(authParams,signAlgo,PartnerConfig.RSA_PRIVATE);
        authParams.put("sign", authSign);
        String redirectURL = "";
        try {
            redirectURL = getRedirectUrl(authParams,reqUrl);
        } catch (Exception e) {
            e.printStackTrace();
        }
        if (StringUtil.isNotBlank(redirectURL)) {
            response.sendRedirect(redirectURL);
            return;
        }
```

2.4 处理支付宝系统通知

支付宝系统的通知包括同步和异步两种方式，同步是指在支付完成后支付宝直接调用商户指定的 call_back_url，并携带参数；异步是指支付宝在支付完成后发送通知到商户指定的 notify_url，以下为具体内容。

2.4.1 call_back_url

用户在支付宝收银台完成支付后，会以 GET 方式跳转到 call_back_url（用户直接点击或自动跳转），同时会携带交易参数。商户在收到这一参数后，要先进行验签（具体点此）。样例如下：

样例

http://10.14.42.49:8080/paychannel/servlet/CallBack?out_trade_no=1320742949342&request_token=requestToken&result=success&trade_no=2011110823389231&sign=49a330fee069465c64e561a25bf31c78

商户可根据“result”参数判断交易状态。具体参数的含义请查询参数表

2.4.2 notify_url

在交易完成后，支付宝通过访问商户提供的地址的形式，将交易状态信息发送给商户服务器。商户通过支付宝的通知判断交易是否成功，具体如下：

商户地址：提供一个 http 的 URL(例:http://www.partnertest.com/servlet/NotifyReceiver)，支付宝将以 POST 方式调用该地址。

通知触发条件：交易状态发生改变，如交易从“创建”到“成功”或“关闭”。

商户返回信息：商户服务器收到通知后需返回纯字符串“success”，不能包含其他任何 HTML 等语言的文本。

通知重发：若支付宝没有收到商户返回的“success”，将对同一笔订单的通知进行周期性重发 (间隔时间为：2 分钟,10 分钟,10 分钟,1 小时,2 小时,6 小时,15 小时共 7 次)。

交易判断条件：收到 trade_status=TRADE_FINISHED（如果签有高级即时到帐协议则 trade_status=TRADE_SUCCESS）的请求后才可判定交易成功（其它 trade_status 状态请求可以不作处理）

以下为支付宝通知的样例：

1） RSA 方式签名时

当商户使用 RSA 签名方式时，商户实际收到支付宝通知请求如下：

样例：

http://www.partnertest.com/servlet/NotifyReceiver?service=alipay.wap.trade.create.direct&sign=Rw/y4ROnNicXhaj287Fiw5pvP6viSyg53H3iNiJ61D3YVi7zGniG2680pZv6rakMCeXX++q9XRLw8Rj6I1//qHrwMAHS1hViNW6hQYsh2TqemuL/xjXRCY3vjm1HCoZOUa5zF2jU09yG23MsMIUx2FAWCL/rgbcQcOjLe5FugTc=&sec_id=0001&v=1.0&notify_data=g3ivqicRwI9rI5jgmSHSU2osBXV1jcxohapSAPjx4f6qiqsoAzstaRWuPuutE0gxQwzMOtwL3npZqWO3Z89J4w4dXIY/fvOLoTNn8FjExAf7OozoptUS6suBhdMyo/YJyS3lVALfCeT3s27pYWihHgQgna6cTfgi67H2MbX40xtexIpUnjgxBkmOLai8DPOUI58y4UrVwoXQgdcwnXsfn2OthhUFiFPfpINgEphUAq1nC/EPymP6ciHdTCWRI6l1BgWuCzdFy0MxJLliPSnuLyZTou7f+Z5Mw24FgOacaISB+1/G+c4XIJVKJwshCDw9Emz+NAWsPvq34FEEQXVAeQRDOphJx8bDqLK75CGZX+6fx88m5ztq4ykuRUcrmoxZLJ+PiABvYFzi5Yx2uBMP/PmknRmj1HUKEhuVWsXR0t6EWpJFXlyQA4uxbShzncWDigndD7wbfNtkNLg5xMSFFIKay+4YzJK68H9deW4xqk4JYTKsv8eom9Eg9MrJZiIrFkFpVYPuaw0y/n61UEFYdzEQZz+garCmMYehEAQCGibYUQXBlf1iwTOZdqJIxdgCpSX21MIa9N9jicmFu8OXWZJkdN+UrSyvIcpzRori+U6522ovMz5Z8EzVTfcUENu+dWJRnhFlo6pvm0a3Fq2wBEyUV1/YYS3LaZiPj+wig5BCyJ92QXZnEUEtn87oX5kuzSRuLcVVi8OJIgyQWaWT9N0YFyH5AfV+VDNxu4UYy6KkGtcaVjSvbzDuzThMXs2HDwX3qujq25A/hzJKlgR9EjcumJeF/TM6eS7JS+FKXE1kUXnMnGbokaNemZn2yKlPC1VO4LU77G5v1nUs6MfYFq9HC4FYiQ6Y+hL8RgAMorty/RYT3cZ8SQCTO0bQ+qJuOnx79YEEmCUQc3iJBp0zFKYXIU6viqJYghEs6F3LiK8TvJRo8+ST5hKtnuU5b8R6f9yD8Uek1BruWvlYaA7I3Cc90CDhTyOghL2oCMOoKlxqgXdh3MGm128FOVyCjDLRw04b+kK83JGFMcjyVuhfhoVeETQicUCtFQ9ItlH3uFkB5su+r3399WGSXyGflrTbFhMq7mRztWotL2ATvf/enMBcGSCSCb47OzGxXhMDGZZu4Sq4pdF9fsZVBHgWsB/KS8bwxyvM068NoqnRmI72zgL7WFWumlm88j3K6KPxbB6soDSXRv6drbSv2t93llE5q4SP6GLztAw7UPWGTJLXOFyhyaszvhyZWxsX+C5PbXoCta1/cxt4Sp4WXDjaHn

6qHI/Vea28xx8fYV/xK5WTmvFwb0k9eRGCgB6/nzmGV1+lPJuK3pKy3L5LbUP0zJFh5gdPG7DecH+F0uBUC0QNMQ==

其中 notify_data 参数值为加密内容，商户需用商户 RSA 私钥先进行解密后再验签，验签具体请见 3.3 RSA 解密、签名和验签　注意：支付宝系统通知待签名数据构造规则比较特殊，为固定顺序，具体见此

2） MD5 方式签名时

当商户使用 MD5 签名方式时，商户实际收到支付宝通知请求如下

样例：

http://www.partnertest.com/servlet/NotifyReceiver?service=alipay.wap.trade.create.direct&sign=Rw/y4ROnNicXhaj287Fiw5pvP6viSyg53H3iNiJ61D3YVi7zGniG2680pZv6rakMCeXX++q9XRLw8Rj6I1//qHrwMAHS1hViNW6hQYsh2TqemuL/xjXRCY3vjm1HCoZOUa5zF2jU09yG23MsMIUx2FAWCL/rgbcQcOjLe5FugTc=&v=1.0&sec_id=MD5&notify_data=<notify><payment_type>1</payment_type><subject>收银台【1283134629741】</subject><trade_no>2010083000136835</trade_no><buyer_email>dinglang@a.com</buyer_email><gmt_create>2010-08-30 10:17:24</gmt_create><notify_type>trade_status_sync</notify_type><quantity>1</quantity><out_trade_no>1283134629741</out_trade_no><notify_time>2010-08-30 10:18:15</notify_time><seller_id>2088101000137799</seller_id><trade_status>TRADE_FINISHED</trade_status><is_total_fee_adjust>N</is_total_fee_adjust><total_fee>1.00</total_fee><gmt_payment>2010-08-30 10:18:26</gmt_payment><seller_email>chenf003@yahoo.cn</seller_email><gmt_close>2010-08-30 10:18:26</gmt_close><price>1.00</price><buyer_id>2088102001172352</buyer_id><notify_id>509ad84678759176212c247c46bec05303</notify_id><use_coupon>N</use_coupon></notify>

其中 notify_data 参数值为明文内容，无需解密。

通知中其他参数意义详见参数列表

第三章 签名详解

3.1 RSA 和 OpenSSL 介绍

3.1.1 什么是 RSA

RSA 是一种非对称的签名算法，即签名密钥（私钥）与验签密钥（公钥）是不一样的，私钥用于签名，公钥用于验签。

在与支付宝交易中，会有 2 对公私钥，即商户公私钥，支付宝公钥。

商户公私钥：由商户生成，商户私钥用于对商户发往支付宝的数据签名；商户公钥需要上传至支付宝，当支付宝收到商户发来的数据时用该公钥验证签名。

支付宝公钥：支付宝提供给商户，当商户收到支付宝发来的数据时，用该公钥验签。

3.1.2 为什么要用 RSA

使用这种算法可以起到防止数据被篡改的功能，保证支付订单和支付结果不可抵赖(商户私钥只有商户知道)。

3.1.3 什么是 OpenSSL

一句话概括：OpenSSL 是基于众多的密码算法、公钥基础设施标准以及 SSL 协议安全开发包。

3.1.4 为什么要用 OpenSSL

通过 OpenSSL 生成的签名和内置的算法可以做到跨平台，这样在不同的开发语言中均可以签名和验签。

3.2 RSA 密钥详解*

3.2.1 找到生成 RSA 密钥工具

(1)下载开发指南和集成资料，如下图，您能看到此文档说明指南和集成包已经下载了。

图 3-1 文档下载

(2)解压下载的压缩包(WS_SECURE_PAY)，找到并解压 openssl-0.9.8k_WIN32(RSA 密钥生成工具).zip 工具包

图 3- 2 openssl

3.2.2 生成商户密钥并获取支付宝公钥

(1) 生成原始 RSA 商户私钥文件

假设解压后的目录为 c:\alipay，命令行进入目录 C:\alipay\bin，执行“*openssl genrsa -out rsa_private_key.pem 1024*”，在 C:\alipay\bin 下会生成文件 rsa_private_key.pem，其内容为原始的商户私钥（请妥善保存该文件），以下为命令正确执行截图：

```
c:\alipay\bin>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)

c:\alipay\bin>
```

图 3- 3 生成原始 RSA 商户私钥文件

(2) 将原始 RSA 商户私钥转换为 pkcs8 格式

命令行执行“ *openssl pkcs8 -topk8 -inform PEM -in rsa_private_key.pem -outform PEM -nocrypt*”得到转换为 pkcs8 格式的私钥。复制下图红框内的内容至新建 txt 文档，去掉换行，最后另存为“private_key.txt”（请妥善保存，签名时使用）。

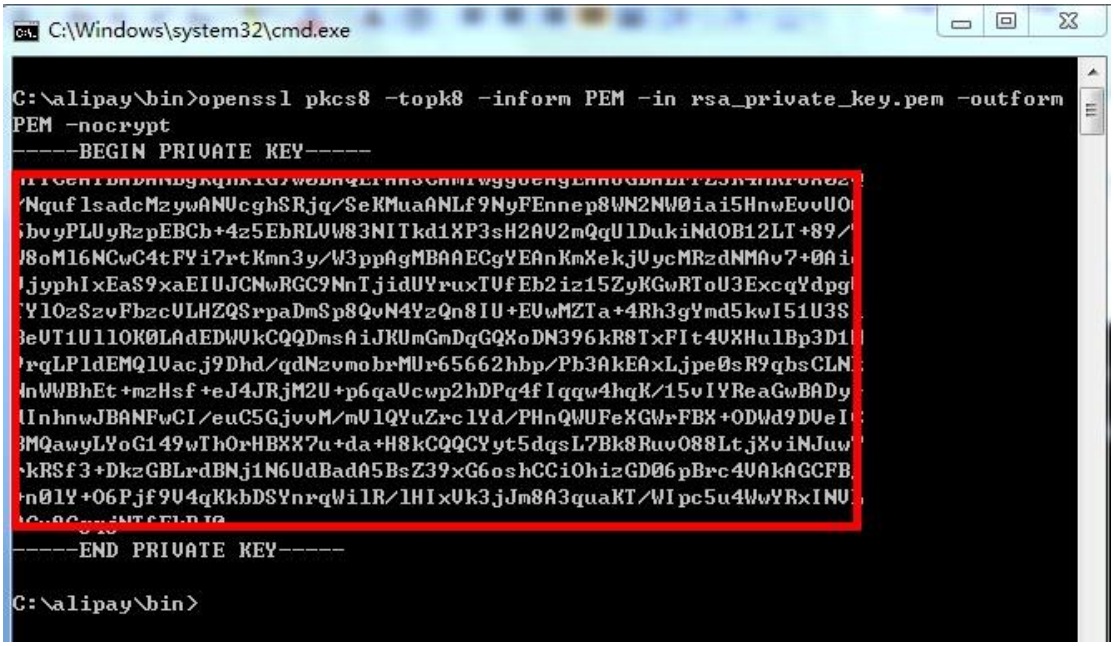

图 3-4 转换私钥格式

(3) 生成 RSA 商户公钥

命令行执行“ *openssl rsa -in rsa_private_key.pem -pubout -out rsa_public_key.pem*”，在 C:\alipay\bin 文件夹下生成文件 rsa_public_key.pem。接着用记事本打开 rsa_public_key.pem，复制全部内容至新建的 txt 文档，删除文件头“-----BEGIN PUBLIC KEY-----”与文件尾“-----END PUBLIC KEY-----”及空格、换行，如下图。最后得到一行字符串并保存该 txt 文件为“public_key.txt”。

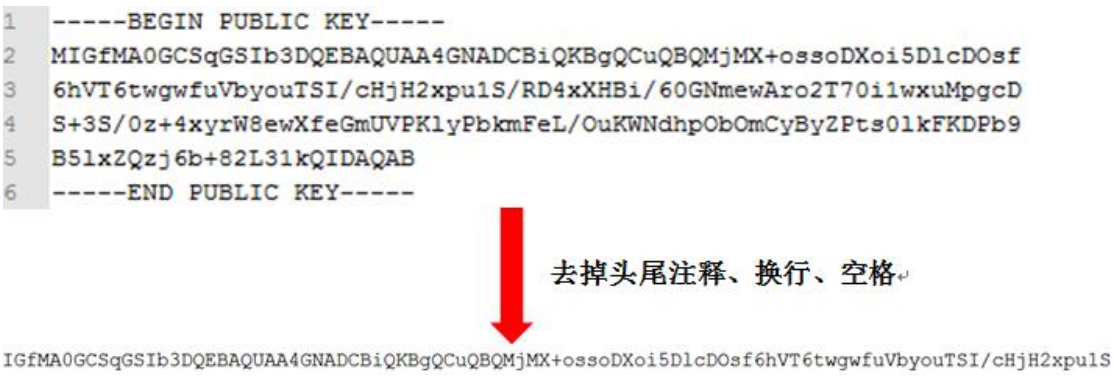

图 3-5 生成公钥

(4) 上传商户公钥至支付宝

浏览器访问 https://ms.alipay.com/index.htm 并用签约帐号登录，点击菜单栏“我的产品”，右侧点击“密钥管理”，见下图红色框内

图 3- 6 商户公钥上传

点击“上传”，选择步骤(3)生成的“public_key.txt”并完成上传。

(5) 获取 RSA 支付宝公钥

成功上传公钥至支付宝后，页面显示如下：

图 3- 7 支付宝公钥获取

其中红色框内部分即支付宝公钥，请复制至新建 txt 文档，去掉换行和空格，妥善保存（用于验签收到的支付宝通知）。

3.3 RSA 解密、签名和验签*

建议：签名和验签尽量在商户服务器端进行，同时一些敏感数据（如公私钥等）也应存储在服务器端，避免可能的安全隐患。

3.3.1 RSA 解密

如果采用了 RSA 签名，则在请求支付宝接口后返回的数据以及支付宝发送给商户的通知中有部分数据会被加密，此时需要使用商户私钥先进行解密，再用支付宝公钥验签。Demo 中提供了 RSA 解密的方法。

类： *com.alipay.client.security.RSASignature*

方法名： *decrypt*

参数： *String content* 需解密的密文

String key 商户私钥

返回值： 解密后的内容

3.3.2 RSA 签名

1）生成待签名数据

将要传递所有参数(**除 sign、sign_type 以外**)按照参数名称字符升序的顺序串联起来（如：p1=v1&p2=v2&p3=v3），构成待签名数据。

注意： a.没有值的参数无需传递，也无需包含到待签名数据中。

b. 如果传递了_input_charset 参数，那么这个参数也应该包含在待签名数据中.

例如：调用某接口需要以下参数：

service=cae_charge_agent

partner=2088006300000000

email=test@msn.com

那么待签名数据就是：

email=test@msn.com&partner=2088006300000000&service= cae_charge_agent

签名参数的构造，可以参考 Demo 中类 *com.alipay.client.security.ParameterUtil*

中的 *getSignData* 方法。

2）签名

使用商户私钥对待签名数据进行签名，获得签名 sign 值。

可以参考 Demo 中类 *com.alipay.client.security.RSASignature* 的 *sign* 方法。

方法名： *sign*

参数： *String content* 需签名的内容

String key 商户私钥

String encode 编码方式，注意务必与_input_charset 参数的值保持一致

返回值： 解密后的内容

3）验签

使用支付宝公钥对支付宝发送至商户的内容进行验签。注意，对于加密的参数值需要先进行解密，对于 encode 过的值要先 decode 才可验签。

可以参考 Demo 中类 *com.alipay.client.security.RSASignature* 的 doCheck 方法。

方法名： *doCheck*

参数： *String content* 待验签的内容

String sign 签名值

String publicKey 支付宝公钥

String encode 编码方式，注意务必与_input_charset 参数的值保持一致

返回值： boolean true 为验签通过，false 为验签不过

以下为各接口返回值需验签参数：

a. ***alipay.wap.trade.create.direct* 交易创建接口**

需要验签的参数包括：*res_data*、*req_id*、*sec_id*、*partner*、*service*、*v*

将这 6 个参数按照参数名称字符升序的顺序串联起来组成待验签字符串，可参考待签名数据的生成

b. **call_back_url**

需要验签的参数包括：*out_trade_no*、*request_token*、*result*、*trade_no*

将这 4 个参数按照参数名称字符升序的顺序串联起来组成待验签字符串，可参考待签名数据的生成

c. **支付宝系统通知 notify**

支付宝系统通知待签名数据构造规则比较特殊，为固定顺序

商户收到如下请求数据:

```
http://www.partnertest.com/servlet/NotifyReceiver?service=alipay.wap.trade.create.direct&sign=Rw/y4ROnNicXhaj287Fiw5pvP6viSyg53H3iNiJ61D3YVi7zGniG2680pZv6rakMCeXX++q9XRLw8Rj6I1//qHrwMAHS1hViNW6hQYsh2TqemuL/xjXRCY3vjm1HCoZOUa5zF2jU09yG23MsMIUx2FAWCL/rgbcQcOjLe5FugTc=&v=1.0&sec_id=0001&notify_data=<notify><payment_type>1</payment_type></notify>
```

则只需对以下数据验签:

```
service=alipay.wap.trade.create.direct&v=1.0&sec_id=0001&notify_data=<notify><payment_type>1</payment_type></notify>
```

3.4 MD5

3.4.1 MD5 简介

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、接收双方事先规定好的，这里我们称其为签名密钥），就可以用于签名了。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。

3.4.2 MD5 Key

当商户使用 MD5 加密方式生成签名之前，需要将待签名参数加上 MD5 Key 参数。

获取 Key：登录 https://ms.alipay.com 我的产品->密钥管理，然后复制出来 MD5 密钥字符串

图 3-3 获取 MD5 密钥

3.4.3 MD5 签名和验签

MD5 待签名数据和待验签数据的构造请见 RSA 签名数据获取

MD5 签名方法可参考：

类： *com.alipay.client.security.MD5Signature*

方法名： *sign*

参数： *String content* 待签名数据

String key MD5 key

返回值： 签名值 sign

MD5 验签方法请参考：

类： *com.alipay.client.security.MD5Signature*

方法名： *verify*

参数： *String content* 待验签的内容

String sign 签名值

String key MD5 key

String charset 编码方式，注意务必与_input_charset 参数的值保持一致

返回值： boolean true 为验签通过，false 为验签不过

第四章 常见问答

1、支付完成之后 Notify_url 接收不到支付宝异步请求的通知？

有以下几种可能：

a. Notify_url 不是一个能够公开访问的地址

b. 防火墙、白名单的问题（建议暂时关闭防火墙试试，或者配置下白名单）

2、验签，报“订单信息被篡改”是什么问题？

可能有以下 2 种情况

a) 有可能数据在传输过程中被黑客截取和篡改

b) 检查待签名字符串中的参数值是否有以下四个符号，如果参数当中包含了这四个字符也会报“订单信息被篡改”：

+加号

&连接符

“双引号

=等号

3、 上传商户公钥报格式错误怎么办？

首先确认上传的位置是否是RSA的下面，注意不要是DSA，无线目前不支持DSA加密；

另外请检查上传的文件中是否去除注释、空格、换行等，必须是一行的字符串

4、当输入付款账号和支付密码后，支付宝收银台报“请求出错!”的提示？

a) 请把demo中所有的参数都加上(Notify_url、Call_back_url、Out_user等都请填上)

b) seller_account_name请不要填2088开头的商户号，请填写支付宝账号(邮件或者手机号格式)

附录 A 错误代码列表

错误代码	说明
0000	系统异常
0001	缺少必要的参数，检查非空参数是否已经传递
0002	签名错误，检查签名的参数是否符合支付宝签

	名规范
0003	服务接口错误，检查 service 是否传递正确
0004	req_data 格式不正确
0005	合作伙伴没有开通接口访问权限，合同是否有效
0006	sec_id 不正确，支持 0001，MD5
0007	缺少了非空的业务参数
ILLEGAL_SIGN	签名错误，检查签名的数据是否符合支付宝签名规范
ILLEGAL_SERVICE	接口不存在，检查 service 是否传递正确
ILLEGAL_PARTNER	无效商户,检查传入的 PARTNER 值是否正确
ILLEGAL_PARTNER_EXTERFACE	商户接口不存在，该商户没有开通该接口
HAS_NO_PRIVILEGE	无权访问该接口
SYSTEM_ERROR	系统错误

附录 B 手机网站支付接口参数表

参数名	中文描述	类型(精度)	说　明	商户必传	参数值样例
service	接口名称	String	注意：交易创建、授权并执行两次请求的值不同。	Y	alipay.wap.trade.create.direct/alipay.wap.auth.authAndExecute
partner	合作伙伴 id	String(16)	合作伙伴在支付宝的用户 ID，与支付宝签约后自动生成	Y	2088002007015955
sec_id	签名算法	String(4)	签名算法。目前只支持 MD5 和 RSA(用 0001 表示)	Y	0001　或　MD5
req_id	请求号	String(32)	请求号用于关联请求与响应，并且防止请求重播。支付宝 wap 限制来自一个 partner 的请求号必须唯一。	Y	1e925b9b4b115961660130f9281e3898
sign	签名	String	签名，对 request/response 中参数签名后的值	Y	72020eb70e0fdcfbf404edcbb83bfd81
format	请求参数格式	String	参数值必须和样例保持一致	Y	xml
v	接口版本号	String	参数值必须和样例保持一致	Y	2.0
req_data	请求业务参数	String	参数值内容为 xml 格式,包含内层标签参数	Y	<?xml version="1.0" encoding="UTF-8"?><dire

					ct_trade_create_req> <subject>彩票</subject><out_trade_no>20080801-1</out_trade_no><total_fee>50</total_fee><seller_account_name>tbbusi003@126.com</seller_account_name><out_user>xxxxx</out_user><notify_url>http://www.yoursite.com/notifyurl.htm</notify_url></direct_trade_create_req>
direct_trade_create_req	固定标签	String	req_data 参数值 xml 内容中必须包含的固定标签。	Y	<subject>彩票</subject><out_trade_no>20080801-1</out_trade_no><total_fee>50</total_fee><seller_account_name>tbbusi003@126.com</seller_account_name><out_user>xxxxx</out_user><notify_url>http://www.yoursite.com/notifyurl.htm</notify_url>
subject	商品名称	String(256)	订单商品名称	Y	彩票
out_trade_no	外部交易号	String(64)	合作伙伴系统的交易号，传递给支付宝系统做外部交易号（不能重复）	Y	2008080101
total_fee	订单价格	String(15)	用户购买的商品或服务的价格（必须是金额的格式,单位：元）	Y	1.01
pay_expire	交易自动关闭时间	Int	买家如未能在该设定值范围内支付成功，交易将被关闭。单位：“分钟”，值区间 0<pay_expire，默认值 21600（15 天）。最终关闭时间点误差 1-2 分钟。	N	10
seller_account_name	卖家帐号	String(100)	交易卖方的支付宝帐号，交易成功后该笔交易的资金会转入这个支付宝帐号中	Y	tbbusi003@126.com
out_user	商户系统用户唯一标识	String(32)	买家在商户系统的唯一标识，当该 out_user 支付成功一次后再来支付时，30 元内无需密码。	N	21321211111

notify_url	商户接受通知的url	String(200)	商户接受通知的 url	Y	http://www.yoursite.com/notifyurl.htm
merchant_url	返回商户	String	用户在支付宝页面可返回商户的链接	N	http://www.yoursite.com/partnerurl.htm
call_back_url	支付成功跳转链接	String(200)	由商户提供，只有当交易支付成功之后，才会跳转到该 url。	Y	http://www.yoursite.com/callbackurl.htm
request_token	token	String(40)	前面调用交易创建接口成功返回后获得的**（注当此参数为页面返回时，为固定值）**		20081113f9d49c20e8e5c8e40b6107ec42259e41
trade_no	交易号	String(64)	交易号，该笔交易在支付宝系统的交易号		2009092904171521
notify_data	通知业务参数	String	通知的业务参数，包含交易号、外部交易号、交易状态等信息。		见例子
payment_type	支付方式	String	用户的支付方式(商户可不关心该参数)		1
buyer_email	买家账号	String(100)	买家的支付宝账号		chenf002@yahoo.cn
gmt_create	创建时间	String	交易创建时间		2009-09-29 19:59:24
notify_type	通知类型	String	该通知的类型，暂时只有交易状态同步(商户可不关心该参数)		trade_status_sync
quantity	数量	String	购买商品数量		1
notify_time	通知时间	String	发送通知的时间		2009-09-29 19:59:25
seller_id	卖家 id	String	卖家的支付宝账号 id		2088102001058148
trade_status	交易状态	String	交易的状态。TRADE_FINISHED（支付成功），WAIT_BUYER_PAY(等待买家付款)		TRADE_FINISHED/ WAIT_BUYER_PAY
is_total_fee_adjust	总价是否被修改	String	交易价格是否被修改，Y 或 N(本接口创建的交易不会被修改)		N
total_fee	交易总价	String	即订单金额。单位：元		2.21
gmt_payment	付款时间	String	交易的付款时间，如果交易未付款，没有该属性		2009-09-29 19:59:25
seller_email	卖家账号	String(100)	卖家的支付宝账号		youngbeckham@gmail.com

gmt_close	交易结束时间	String	交易结束的时间		2009-09-29 19:59:25
price	单个商品价格	String	目前和 total_fee 值相同。单位：元		2.21
buyer_id	买家 id	String	买家的支付宝账号 id		2088101000137393
notify_id	通知 id	String	唯一识别通知内容，重发相同内容的通知 notify_id 值不变。		2311b764be6fba98f593ba98f7eb7470
use_coupon	是否使用红包	String	交易时是否使用红包，Y 或 N		N
_input_charset	参数编码字符集	String	见签名机制	N	GBK