



# 交通银行 B2C 网上支付平台 商户端开发编程说明

## Java 环境

(版本号 v1.0)

# 1 开发包说明

交通银行提供的开发包中包含商户编程所需的 Jar 包 文件、商户证书、示例 Jsp 页面和说明文档。开发包的目录部署如下：

- cert 为测试提供商户端测试环境的商户测试证书和银行端测试环境测试公钥根证书；为生产提供商户端生产环境的商户证书和银行端生产环境公钥根证书；
- demo 存放交易演示 Jsp 页面文件，商户可参照 demo 中的页面进行编程开发；起始页面：Index.htm；
- doc 存放开发编程说明文档；
- icon 交通银行 logo 徽标；
- ini 商户端 API 配置文件，API 初始化需指定该配置文件，配置文件内容包含地址的指定、证书的指定及日志存放目录指定等；
- lib 提供商户编程 API 所需全部 Jar 包 文件。

jar 包说明：

- |                                   |                      |
|-----------------------------------|----------------------|
| ■ bocommapi1.0.jar                | 联机交易开发编程 API 包；      |
| ■ NetSignServer 1.8.jar           | 签名与验签 API 包          |
| ■ ISFJ_v2_0_119_2_BAISC_JDK13.jar | 辅助包，Java 环境 1.3 时使用； |
| ■ ISFJ_v2_0_119_2_BAISC_JDK14.jar | 辅助包，Java 环境 1.4 时使用； |

## 2 商户证书下载与更新

商户证书是交通银行 CA 发放的基于 PKI 认证体系的数字证书，主要用于网上支付交易及商户联机查询等过程中的签名和验签。

### 2.1 银行申请

对于申请商户证书，在网上开立特约商户时应注意：

- 1、 商户填写商户证书申请表，领取银行提供的商户证书密码信封。
- 2、 商户填写用户申请表，并指定特定用户，根据商户的申请，银行为其分配“ 商户配置 ” 权限。然后该指定用户登录企业网银，便可使用密码信封下载商户证书。

### 2.2 商户证书下载

商户在配置和调试 API 前，应先下载在我行申请的商户证书，并导出到应用服务器上。


✧ 证书下载

商户企业中拥有“商户证书下载”权限的用户，登录企业网银后，在菜单中选择“**商户配置**”->“**商户证书下载**”，进入证书下载页面：

账务查询	电子商务	企业付款	财务服务	现金管理	国际业务	客户服务	特色业务	English	帮助	退出
------	------	------	------	------	------	------	------	---------	----	----

B2C业务 | 划款业务 | **商户配置**

>> B2C参数设置 >> **商户证书下载** >> 商户证书更新

 企业商户证书下载：

请输入您的参考号和授权码：

参考号： -  -  -

授权码： -  -  -

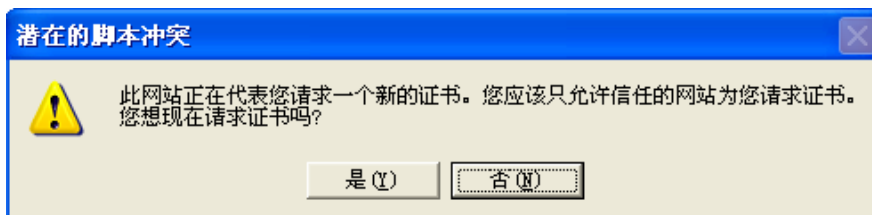
若证书下载失败，请不要关闭IE浏览器，进入“证书更新”页面重新下载证书。若证书下载仍然失败，请联系当地分行进行证书恢复。

下面的下拉框将列出您可选择的CSP（Cryptographic Service Provider，加密服务提供商）。为了使用智能卡读卡器或其它设备，请选择合适的CSP。

厂商：

CSP：

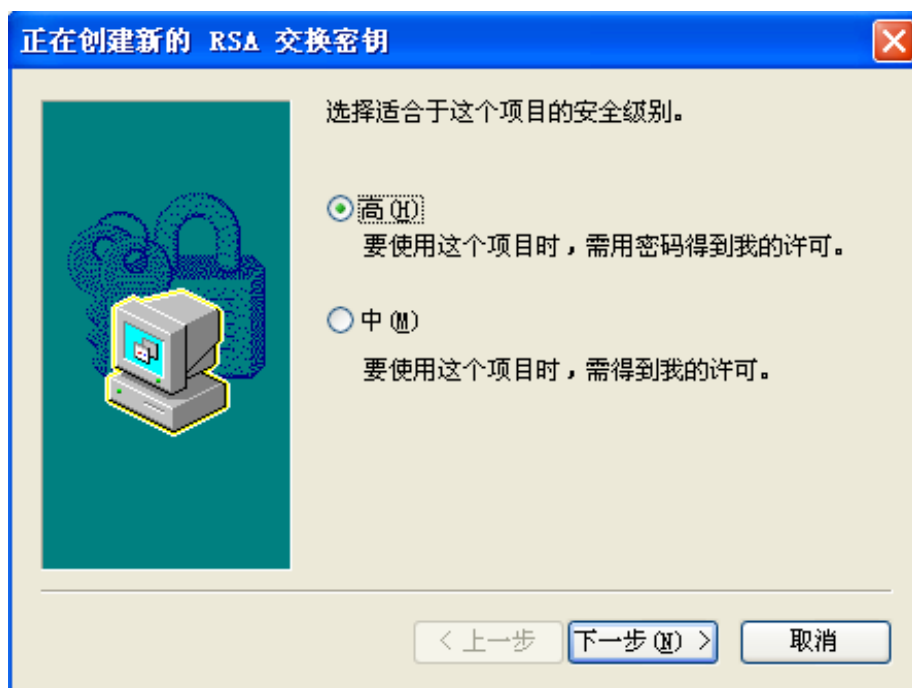
输入密码信封中的参考号和授权码，点击“**下载证书**”按钮，开始下载证书



选择“是(Y)”



点击“设置安全级别”按钮



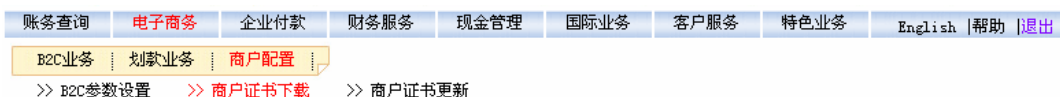
选择安全级别为“高”，点击“下一步”按钮



输入商户证书的密码，该密码可由商户自行定义。设置完成后，商户证书的密码作为配置参数放在证书配置文件 ini/B2CMerchant.xml 中（配置方式见后）。



点击“确定”，完成证书下载



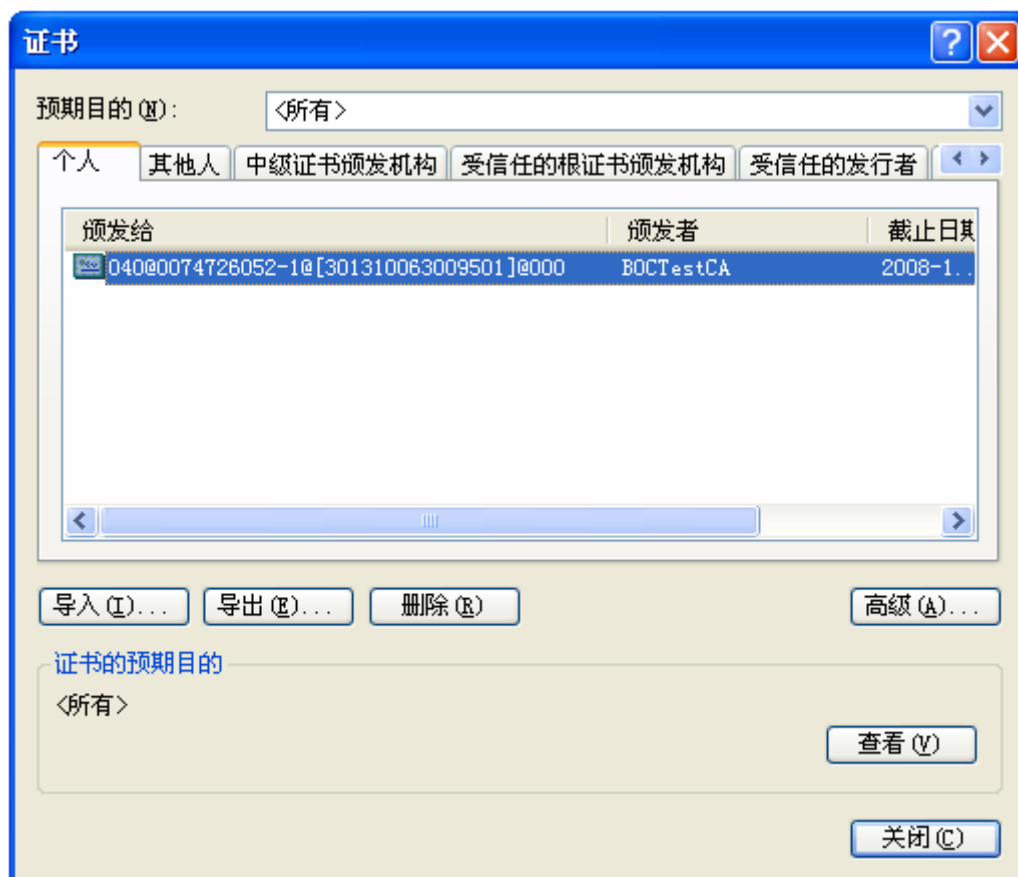
下载商户服务器证书：

证书下载成功, 请从浏览器中导出证书！

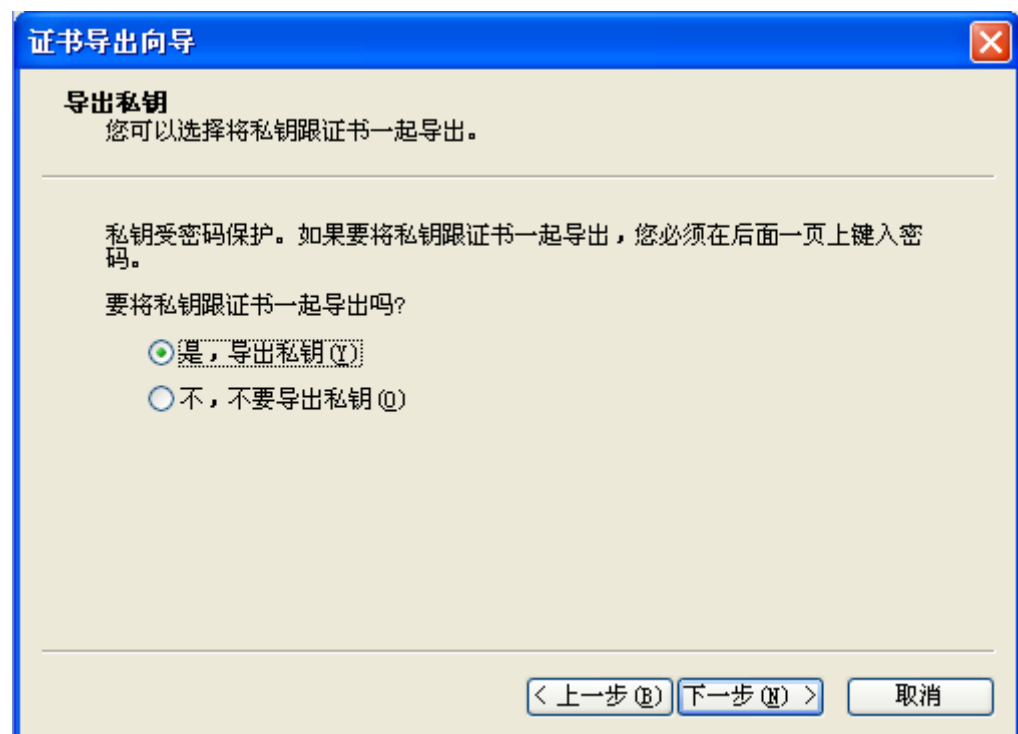
#### ◇ 导出

证书下载完成后，需将 IE 浏览器中的证书以 pfx 文件形式导出，以便 API 软件包调用。

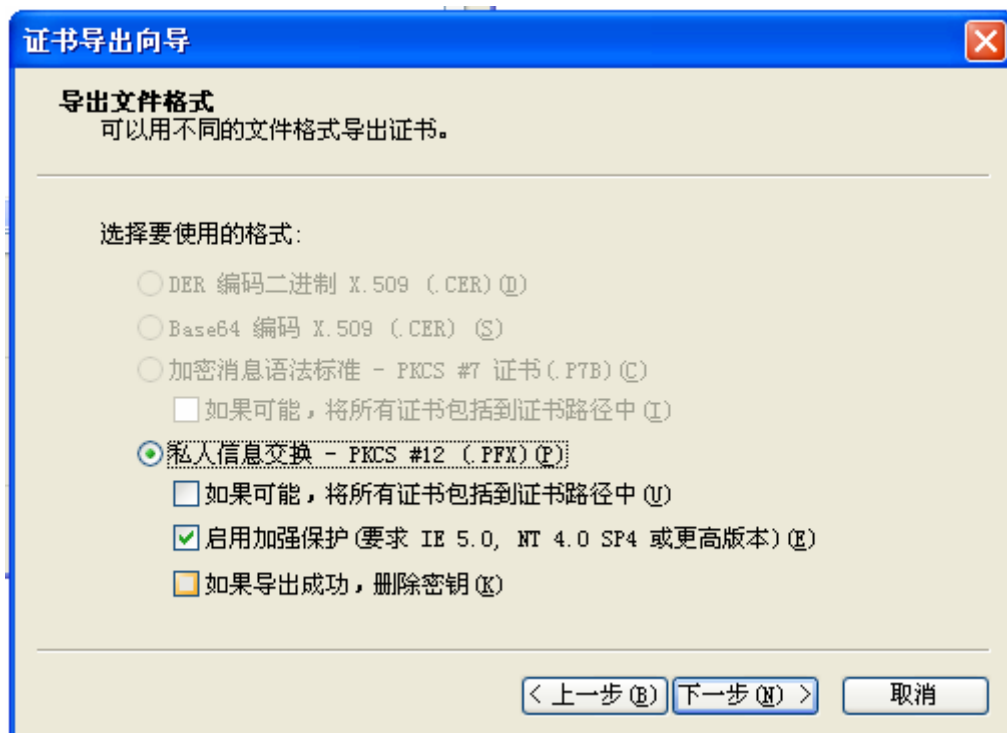
在 IE 浏览器中选择“工具”->“Internet 选项”->“内容”->“证书”->“个人”  
后便可看到商户证书：



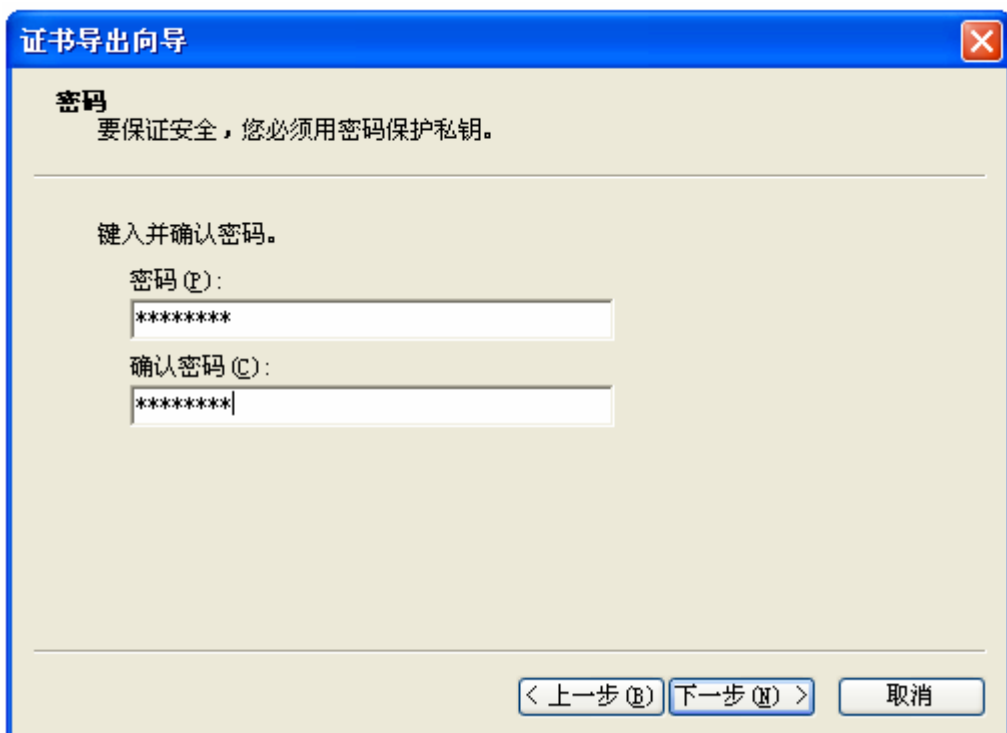
选择商户证书，点击“导出”按钮，可将该证书导出



注意导出时应选择“是，导出私钥 (Y)”



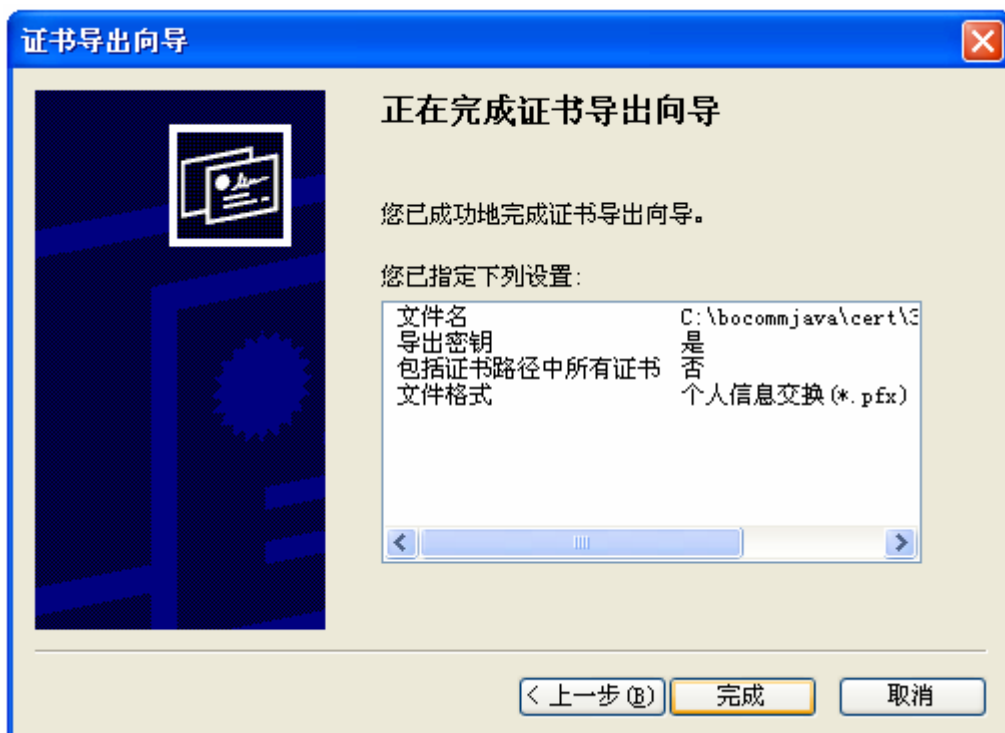
点击“下一步”按钮



输入商户证书密码 (此密码为下载时所设定的密码), 点击“下一步”按钮



选择证书安放路径，点击“下一步”按钮。



点击“完成”，完成证书导出。

## 2.3 商户证书更新

商户证书更新主要为因故遗失、损坏证书的商户提供证书自行更新服务。商户下载新商



户证书后原证书作废。

商户中拥有“商户证书更新”权限的用户，登录企业网银，在菜单中选择“**商户配置**”->“**商户证书更新**”，进入商户证书更新页面：

账务查询	<b>电子商务</b>	企业付款	财务服务	现金管理	客户服务	特色业务	English   帮助   退出
B2C业务   划款业务   <b>商户配置</b>							
>> B2C参数设置    >> 商户证书下载    >> <b>商户证书更新</b>							

商户证书更新：

商户号：301310053110018

证书编号	证书类型	证书状态	生效时间	证书有效期	证书DN
000	客户证书	正常	2006-07-08 15:24:21	24个月	CN=04000131000010033488[301310053110018]@000, OU=Merchant s, OU=BANKCOMM, O=boccomm.com, C=CN

©2005 交通银行版权所有  
 E-mail: 95559@bankcomm.com    二十四小时服务热线: 95559

选择所要下载的商户证书，点击“**证书更新**”按钮，下载证书



下载过程同前 1.2 商户证书下载，结束后证书更新完成。

## 3 证书与开发包安装

### 3.1 环境搭设

商户需要保证开发环境完整而正确的搭设(如 JAVA 开发工具 Eclipse ,JSP WEB 服务器 TOMCAT , WINDOWS 的 WEB 服务器 IIS 等的安装)。

### 3.2 安装前检查

首先,请查看开发包待安装环境的 [ClassPath](#) (类路径)所指路径。然后对应 J2EE 的应用服务器,查找并确认 WEB-INF/lib 和 WEB-INF/classes 目录。最后检查安装环境中使用 Jre 的版本,允许版本为:1.3、1.4 或更高版本。

### 3.3 开发包安装

将开发包中 Lib 目录下的所有 Jar 文件复制到 [ClassPath](#) 或 WEB-INF/lib 目录下,注意: ISFJ\_v2\_0\_119\_2\_BAISC\_JDK13.jar 和 ISFJ\_v2\_0\_119\_2\_BAISC\_JDK14.jar 不能同时使用。即 Jre1.3 请使用 ISFJ\_v2\_0\_119\_2\_BAISC\_JDK13.jar; Jre1.4 或更高版本请使用 ISFJ\_v2\_0\_119\_2\_BAISC\_JDK14.jar。

### 3.4 配置参数文件

商户端的 API 在初始化时,需要从配置文件中获得必须的配置参数。如果初始化失败,API 的所有接口将报错。

编辑 ini/B2CMerchant.xml 文件,文件内容示例如下:

```
<?xml version="1.0" encoding="gb2312"?>
<BOCOMB2C>
  <ApiURL>https://ebanktest.95559.com.cn/corporbank/NsTrans</ApiURL>
  <!--测试环境 API 请求 URL,注:[请在 hosts 文件中增加对应关系 61.172.242.229
ebanktest.95559.com.cn],生产系统改为 https://ebank.95559.com.cn/corporbank/NsTrans -->
  <OrderURL>https://61.172.242.228/netpay/MerPayB2C</OrderURL>
  <!-- 测试环境 订单支付请求 URL,生产系统改为
https://pbank.95559.com.cn/netpay/MerPayB2C-->
  <!--商户可修改参数-->
  <EnableLog>true</EnableLog><!--日志开关(true:表示写日志,false:表示不写日志)-->
  <LogPath>C:\bocommjava\log</LogPath> <!--日志文件存放目录-->
```

```
<SettlementFilePath>C:\bocommjava\settlement\</SettlementFilePath>
<!-- 下载结算明细文件存放目录-->
<MerchantCertFile>C:\bocommjava\cert\301310063009501.PFX</MerchantCertFile>
<!-- 测试环境商户证书文件（绝对路径），商户上线时该证书换为商户生产证书-->
<MerchantCertPassword>12345678</MerchantCertPassword>
<!-- 测试环境商户证书私钥加密密码，商户上线时该密码换为商户生产证书密码-->
<RootCertFile>C:\bocommjava\cert\test_root.cer</RootCertFile>
<!-- 测试环境根证书文件，商户上线时该根证书换为交通银行生产根证书 root.cer-->
</BOCOMB2C>
```

注意：在测试联调过程中，请在 hosts 文件中增加对应关系 61.172.242.229  
ebanktest.95559.com.cn

## 3.5 证书相关放置与安装

### ✧ 放置商户证书

建立交通银行商户证书安放目录，如：

Unix 系统：/bocommjava/cert/

Windows 系统：C:\bocommjava\cert\（要求与配置文件中设定的目录保持一致）

将交通银行提供的 root.cer 和网银上下载的商户证书（PFX 格式文件）移动到上述目录下。

### ✧ 安装银行根证书

商户端的运行环境需要将银行根证书导入 JVM。导入根证书的方法如下：

1. 确认使用 JRE 的 keystore 文件的全路径；一般情况路径为：

\$JAVA\_HOME\jre\lib\security\cacerts ；

2. 在命令行环境下执行以下命令：

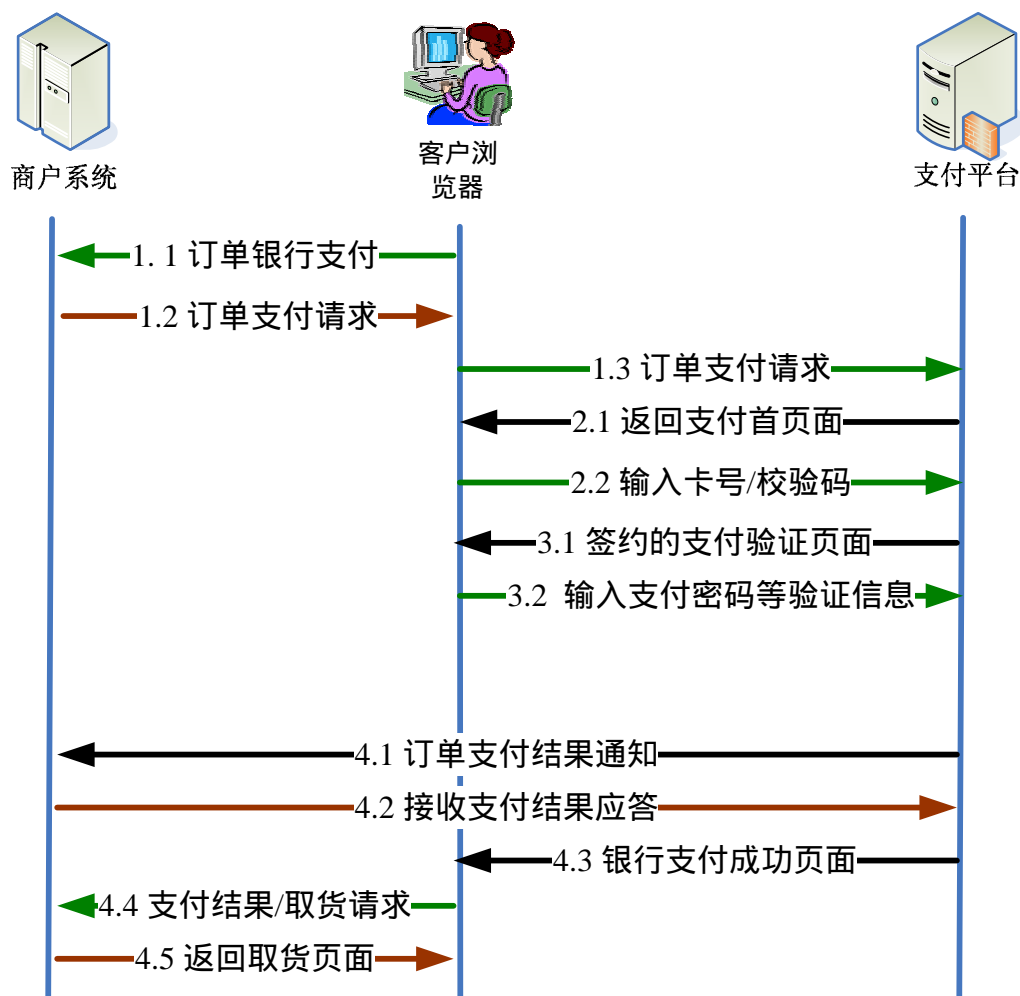
```
keytool -import -keystore $JAVA_HOME\jre\lib\security\cacerts -storepass changeit -keypass  
changeit -alias bocommca -file root.cer 完成导入。
```

注：keytool 是 JRE 的密钥和证书管理工具

## 4 商户 B2C 订单支付编程

### 4.1 支付交易流程

B2C 订单支付交易流程图：



#### 4.1.1 商户订单接收处理

1.1 客户（消费者）在商户网站选购完商品，选择通过交通银行进行网上支付；

1.2 商户生成订单信息，调用交行 API 签名模块对订单信息进行签名，附入客户浏览器重定向至交行支付平台；

1.3 支付平台接收到客户浏览器请求后，对订单信息进行验证，对于商户证书无效、签名不正确、订单信息不合法、该订单在当日已有成功支付、商户状态不正常等情况则为客户返

回相应错误页面；

## 4.1.2 网上支付首页页面处理

- 2.1 若订单合法，支付平台则返回给客户网上支付的首页面，由客户输入支付用的银行卡号或支付卡号和校验码；
- 2.2 支付平台根据个人客户输入的银行卡号或支付卡号判断客户身份及相关签约版本。

## 4.1.3 支付验证处理

- 3.1 对于个人客户，交行网上支付分为多个版本。根据安全级别，可分为手机注册版和证书认证版；根据银行卡种类，可分为借记卡/准贷记卡和贷记卡。因不同版本的额度及校验方式有所不同，所以支付平台根据客户的签约版本显示不同的支付验证页面。
- 3.2 客户输入密码等验证信息，支付平台向银行后台发起扣款，若因密码输错、余额不足等原因导致的支付失败，则允许客户返回支付首页输入卡号、密码，重新提交，直到客户支付成功或关闭浏览器为止。

## 4.1.4 通知商户支付结果

- 4.1 若银行后台扣款成功，则依据商户提交的订单接口（见后 2.2）中的“通知方式”内容对商户进行通知。

“通知方式”内容：

- a. “不通知”：则直接进入 4.3（获取支付结果）处理流程；
- b. “通知”：若商户订单信息中有明确的“主动通知 URL”，则支付平台将向该 URL 发送支付结果，得到 http 200 应答则认为通知完成。若未收到应答，则支付平台将在最大次数内定时轮循重新发送通知。

注：商户若需要银行提供主动通知功能，则商户自身需要具备能够处理重复通知情况的能力。在支付平台已经成功完成支付结果通知的情况下，允许商户向支付平台发起再次进行支付结果通知的请求。

- c. “取页面”：若商户订单信息中有明确的“主动通知 URL”，则支付平台向该 URL 发送 http 通知，获取商户返回的结果页面并将其显示在 4.3（获取支付结果）中。

若超时未完整获得商户页面，则在 4.3（获取支付结果）中直接显示银行默认的支付结果页面。

#### 4.2 支付平台显示给个人客户支付结果

a. 若订单接口中的“通知方式”为“取页面”，且成功获得商户返回的结果页面，则支付平台将商户结果页面显示给个人客户。注：商户结果页面不宜过大和复杂，应简洁为宜。

b. 如果未获取商户结果页面，支付平台则给个人客户显示银行指定支付结果页面。

若订单接口中指定了“取货 URL”，则支付结果页面将包含“取货”按钮，当客户点击“取货”按钮或响应时间超过订单接口中的“自动跳转时间”，客户浏览器将被重定向至商户取货页面，同时附带支付结果及银行签名（商户处理支付结果及银行签名方式见 4.3）；

注：若订单接口中指定了“取货 URL”，且商户在订单接口中将“自动跳转时间”设为 0，则银行端不显示支付结果，客户也不需要点击“取货”按钮，客户浏览器就被直接重定向至商户取货页面。

4.3 商户系统返回给客户取货信息页面。

## 4.2 提交订单

### 4.2.1 订单接口：

	中文名称	必输	英文名称	长度	备注
1.	消息版本号	Y	interfaceVersion	7	固定为 1.0.0.0
2.	商户客户号（商户编号）	Y	merID	15	商户客户号是银行生成的 15 位编号
3.	订单号	Y	orderid	20	商户应保证定单号+定单日期的唯一性
4.	商户订单日期	Y	orderDate	8	YyyyMMdd
5.	商户订单时间	N	orderTime	8	Hhmmss
6.	交易类别	Y	tranType	1	0 B2C
7.	订单金额	Y	amount	17	单位:元 并带两位小数

					15 位整数+2 位小数
8.	订单币种	Y	curType	3	人民币 CNY
9.	订单内容	N	orderContent	100	商家填写的其他订单信息，在个人客户页面显示
10.	商家备注	N	orderMono	100	不在个人客户页面显示的备注，但可在商户管理页面上显示
11.	物流配送标志	N	phdFlag	1	0 非物流 1 物流配送
12.	通知方式	Y	notifyType	1	0 不通知 1 通知 2 转页面
13.	主动通知 URL	N	merURL	100	为空则不发通知
14.	取货 URL	N	goodsURL	100	为空则不显示按钮，不自动跳转
15.	自动跳转时间	N	jumpSeconds	2	等待 n 秒后自动跳转取货 URL；若不填写则表示不自动跳转
16.	商户批次号	N	payBatchNo	10	商家可填入自己的批次号，对账使用
17.	代理商名称	N	proxyMerName	30	二级商户编号/或证件号码
18.	代理商证件类型	N	proxyMerType	4	代理商证件类型
19.	代理商证件号码	N	proxyMerCredentials	20	代理商证件号码
20.	渠道编号	Y	netType	1	固定填 0: (html 渠道)
21.	商家签名	Y	merSignMsg	2000	商家签名 detect 方式签名

在订单支付请求提交到交行支付平台前，商户系统需要调用交行软件包对订单数据进行签名，并附带入客户浏览器。

商城需给交行网上支付平台提供具有如下 form 表单的页面，详细可参照 demo/merchant.htm

```
<html>

<head>

<title>商户订单测试</title>

<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

</head>
```

```
<body bgcolor="#FFFFFF" text="#000000">

<form name="form1" method="post" action="merchant.jsp">

  <p>

    <input type="hidden" name="merCert" value="">

    交通银行商户订单测试 </p>

    <table width="75%" border="0" cellpadding="0" cellspacing="0">

      <tr>

        <td width="23%">消息版本号*</td>

        <td width="77%">

          <input type="text" name="interfaceVersion" value="1.0.0.0">

          <input type="submit" id="Submit" value="提交" name="submit">

        </td>

      </tr>

      <tr>

        <td width="23%">订单号*</td>

        <td width="77%">

          <input type="text" name="orderid" maxlength="20" value="">

        </td>

      </tr>

      <tr>

        <td width="23%">商户订单日期*</td>

        <td width="77%">

          <input type="text" name="orderDate" value="">

          yyyyMMdd</td>

        </tr>

      <tr>

        <td width="23%">商户订单时间*</td>

        <td width="77%">
```



```

        <input type="text" name="orderTime" value="">
        HHmmss</td>
    </tr>
    <tr>
        <td width="23%">交易类别*</td>
        <td width="77%">
            <input type="text" name="tranType" value="0" readOnly>
            0:B2C </td>
        </tr>
    <tr>
        <td width="23%">订单金额*</td>
        <td width="77%">
            <input type="text" name="amount" value="">
        </td>
    </tr>
    <tr>
        <td width="23%">交易币种*</td>
        <td width="77%">
            <input type="text" name="curType" value="CNY">
        </td>
    </tr>
    <tr>
        <td width="23%">订单内容</td>
        <td width="77%">
            <input type="text" name="orderContent" value="">
        </td>
    </tr>
    <tr>
        <td width="23%">商家备注</td>

```

```

        <td width="77%">

            <input type="text" name="orderMono" value="">

        </td>

    </tr>

    <tr>

        <td width="23%">物流配送标志</td>

        <td width="77%">

            <input type="text" name="phdFlag" value="" readOnly>

        </td>

    </tr>

    <tr>

        <td width="23%">通知方式*</td>

        <td width="77%">

            <input type="text" name="notifyType" value="0">

            0 不通知 1 通知 2 抓取页面 </td>

    </tr>

    <tr>

        <td width="23%">主动通知 URL</td>

        <td width="77%">

            <input type="text" name="merURL" value="" size="40">

        </td>

    </tr>

    <tr>

        <td width="23%">取货 URL </td>

        <td width="77%">

            <input type="text" name="goodsURL" value="" size="40">

        </td>

    </tr>

    <tr>

```

```
<td width="23%">自动跳转时间 </td>

<td width="77%">

    <input type="text" name="jumpSeconds" value="">

    ( 秒 ) </td>

</tr>

<tr>

    <td width="23%">商户批次号</td>

    <td width="77%">

        <input type="text" name="payBatchNo" value="">

        商家对账使用</td>

</tr>

<tr>

    <td width="23%">代理商家名称 </td>

    <td width="77%">

        <input type="text" name="proxyMerName" value="">

    </td>

</tr>

<tr>

    <td width="23%">代理商家类型 </td>

    <td width="77%">

        <input type="text" name="proxyMerType" value="">

    </td>

</tr>

<tr>

    <td width="23%">代理商家证件号码 </td>

    <td width="77%">

        <input type="text" name="proxyMerCredentials" value="">

    </td>

</tr>
```

```

<tr>

    <td width="23%">渠道编号* </td>

    <td width="77%">

        <input type="text" name="netType" value="0" readOnly>

        0:html 渠道

        <input type="hidden" name="merSignMsg" >

    </td>

</tr>

<tr>

    <td width="23%">&nbsp;</td>

    <td width="77%">&nbsp;</td>

</tr>

</table>

<p>&nbsp;</p>

</form>

</body>

</html>

```

### 4.2.2 签名方式：

将 1 至 20 号域用 | 分隔，进行签名，如

```

interfaceVersion|merID|orderid|orderDate|orderTime|tranType|amount|curType|orderContent|order
Mono|phdFlag|notifyType|merURL|goodsURL|jumpSeconds|payBatchNo|proxyMerName|proxy
MerType|proxyMerCredentials|netType

```

然后通过交通银行签名组件的 Detached 方式进行签名

签名信息不带原文

### 4.2.3 程序示例：

将上述 1 至 20 号域以 | 分隔打包后，形成 |分隔的字符串，然后进行签名，详细情况可参照 demo/merchant.jsp,示例如下

```
<html>
<head>
<title>商户订单测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

</head>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<%
    BOCOMB2CClient client = new BOCOMB2CClient();
    int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");//该代码只需调用一次
    if (ret != 0){ //初始化失败
        out.print("初始化失败,错误信息 : "+client.getLastErr());
        return;
    }
    String interfaceVersion;
    String merID ;
    String orderid;
    String orderDate;
    String orderTime;
    String tranType;
    String amount;
    String curType;
    String orderContent;
    String orderMono;
    String phdFlag;
    String notifyType;
    String merURL;
    String goodsURL;
    String jumpSeconds;
    String payBatchNo;
    String proxyMerName;
    String proxyMerType;
    String proxyMerCredentials;
    String netType;
    String sourceMsg;

    interfaceVersion      = request.getParameter("interfaceVersion");
    merID                  = BOCOMSetting.MerchantID;
    orderid                = request.getParameter("orderid");
    orderDate              = request.getParameter("orderDate");
    orderTime              = request.getParameter("orderTime");
    tranType               = request.getParameter("tranType");
```

```

amount                = request.getParameter("amount");
curType               = request.getParameter("curType");
orderContent          = request.getParameter("orderContent");
orderMono             = request.getParameter("orderMono");
phdFlag               = request.getParameter("phdFlag");
notifyType            = request.getParameter("notifyType");
merURL                = request.getParameter("merURL");
goodsURL              = request.getParameter("goodsURL");
jumpSeconds           = request.getParameter("jumpSeconds");
payBatchNo            = request.getParameter("payBatchNo");
proxyMerName          = request.getParameter("proxyMerName");
proxyMerType          = request.getParameter("proxyMerType");
proxyMerCredentials   = request.getParameter("proxyMerCredentials");
netType               = request.getParameter("netType");
sourceMsg              = interfaceVersion + "|" + merID + "|" + orderid + "|" +
orderDate + "|" + orderTime + "|" + tranType + "|" + amount + "|" + curType + "|" + orderContent
+ "|" + orderMono + "|" + phdFlag + "|" + notifyType + "|" + merURL + "|" + goodsURL + "|" +
jumpSeconds + "|" + payBatchNo + "|" + proxyMerName + "|" + proxyMerType + "|" +
proxyMerCredentials + "|" + netType;

com.bocom.netpay.b2cAPI.NetSignServer nss = new
com.bocom.netpay.b2cAPI.NetSignServer(); //初始化签名库
String merchantDN = BOCOMSetting.MerchantCertDN;
nss.NSSetPlainText(sourceMsg.getBytes("GBK"));
byte bSignMsg[] = nss.NSDetachedSign(merchantDN); //对订单进行签名
if(nss.getLastErrnum() < 0){
    out.print("ERROR:商户端签名失败");
    return;
}
String signMsg = new String(bSignMsg,"GBK");
%>
<body bgcolor="#FFFFFF" text="#000000" onload="javascript:form1.submit()">

<form name="form1" method="post" action="<%=BOCOMSetting.OrderURL%>">

    <input                type="hidden"                name="interfaceVersion"
value="<%out.print(interfaceVersion);%>">

    <input type="hidden" name="merID" value="<%out.print(merID);%>">

    <input type="hidden" name="orderid" value="<%out.print(orderid);%>">

    <input type="hidden" name="orderDate" value="<%out.print(orderDate);%>">

```

```
<input type="hidden" name="orderTime" value="<% out.print(orderTime);%>">

<input type="hidden" name="tranType" value="<% out.print(tranType);%>" >

<input type="hidden" name="amount" value="<% out.print(amount);%>">

<input type="hidden" name="curType" value="<% out.print(curType);%>">

<input type="hidden" name="orderContent" value="<% out.print(orderContent);%>">

<input type="hidden" name="orderMono" value="<% out.print(orderMono);%>">

<input type="hidden" name="phdFlag" value="<% out.print(phdFlag);%>" >

<input type="hidden" name="notifyType" value="<% out.print(notifyType);%>">

<input type="hidden" name="merURL" value="<% out.print(merURL);%>">

<input type="hidden" name="goodsURL" value="<% out.print(goodsURL);%>">

<input type="hidden" name="jumpSeconds" value="<% out.print(jumpSeconds);%>">

<input type="hidden" name="payBatchNo" value="<% out.print(payBatchNo);%>">

<input type="hidden" name="proxyMerName" value="<% out.print(proxyMerName);%>">

<input type="hidden" name="proxyMerType" value="<% out.print(proxyMerType);%>">

<input type="hidden" name="proxyMerCredentials"
value="<% out.print(proxyMerCredentials);%>">

<input type="hidden" name="netType" value="<% out.print(netType);%>" >

<input type="hidden" name="merSignMsg" value="<% out.print(signMsg);%>">

</form>
</body>
</html>
```

## 4.3 获取支付结果

### 4.3.1 获取支付结果接口：

银行通知商户的方式分为客户浏览器重定向返回和主动通知两种，无论哪种方式，支付平台返回报文的接口都是一致的，并附带银行签名。

返回结果接口：

	中文名称	长度	英文名称	备注
1.	商户客户号	15	MERCHNO	
2.	订单编号	20	ORDERNO	
3.	交易金额	17	TRANAMOUNT	1 位符号+13 位整数+1 位小数点+2 位小数
4.	交易币种	3	TRANCURRTYPE	CNY 人民币
5.	平台批次号	10	PAYBATNO	银行支付平台的批次号
6.	商户批次号	10	MERCHBATCHNO	为商家预留的对账字段
7.	交易日期	8	TRANDATE	yyyymmdd
8.	交易时间	8	TRANTIME	Hhmmss
9.	交易流水号	15	SERIALNO	
10.	交易结果	1	TRANRST	1 成功
11.	手续费总额	17	FEESUM	
12.	银行卡类型	1	CARDTYPE	0 借记卡 1 准贷记卡 2 贷记卡 3 他行银联卡
13.	银行备注	100	BankMoNo	
14.	错误信息描述	100	ErrDis	
15.	银行签名	2000	SignMsg	

商户系统将收到银行支付平台如下的支付通知信息：



上表 1 至 15 号域以 | 分隔的字符串

例如：

```
301310053110018|99990001|100.00|CNY|14124|412444|20060729|120902|0000052|1|10.21|
2||QWR+QWRQWEQD+G\RQWRFVBNJKFSFAF
```

其中

```
301310053110018|99990001|100.00|CNY|14124|412444|20060729|120902|0000052|1|10.21|
2|| 是源文
```

QWR+QWRQWEQD+G\RQWRFVBNJKFSFAF 是签名信息

商户系统在接收到通知结果后,需调用交通银行验签控件对返回数据的签名信息进行验签。

### 4.3.2 验签方式：

通过 request.getParameter("notifyMsg")方式,可得到交通银行支付平台发送到商户的支付结果,各个字段以"|"分隔,其中签名信息包含在最后一个字段中。得到源文,截取支付结果中的签名信息,通过交通银行签名组件的 Detached 方式进行验签。

### 4.3.3 程序示例：

详细情况可参照 demo/merchant\_result.jsp

```
<html>
<head>
<title>交通银行商户测试结果页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<body bgcolor="#FFFFFF" text="#000000">
<%
out.print("商户结果页面");
out.print("<br>");
out.print("-----");
out.print("<br>");

String notifyMsg = request.getParameter("notifyMsg");//获取银行通知结果
int lastIndex = notifyMsg.lastIndexOf("|");
```

```
String signMsg = notifyMsg.substring(lastIndex+1, notifyMsg.length());//获取签名信息
String srcMsg = notifyMsg.substring(0, lastIndex+1);
int veriyCode = -1;
com.infosec.NetSignServer nss = new com.infosec.NetSignServer();    //初始化签名库
nss.NSDetachedVerify( signMsg.getBytes("GBK"),srcMsg.getBytes("GBK")); //对通知结果进行验签
veriyCode = nss.getLastErrnum();
if (veriyCode<0){//验签出错
    out.print("商户端验证签名失败：return code:"+veriyCode);
    return;
}
java.util.StringTokenizer stName = new java.util.StringTokenizer(srcMsg, "|");//拆解通知结果到Vector
java.util.Vector vc = new java.util.Vector();
int i =0;
while (stName.hasMoreTokens()) {
    String value = (String) stName.nextElement();
    if (value.equals("")) value = "&nbsp;";
    vc.add(i++,value);
}

%>
<table width="75%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td width="14%">商户客户号</td>
        <td width="86%"><%out.print(vc.get(0));%></td>
    </tr>
    <tr>
        <td width="14%">订单编号</td>
        <td width="86%"><%out.print(vc.get(1));%></td>
    </tr>
    <tr>
        <td width="14%">交易金额</td>
        <td width="86%"><%out.print(vc.get(2));%></td>
    </tr>
    <tr>
        <td width="14%">交易币种</td>
        <td width="86%"><%out.print(vc.get(3));%></td>
    </tr>
    <tr>
        <td width="14%">平台批次号</td>
        <td width="86%"><%out.print(vc.get(4));%></td>
    </tr>
    <tr>
```

```

        <td width="14%">商户批次号</td>
        <td width="86%"><%out.print(vc.get(5));%></td>
    </tr>
    <tr>
        <td width="14%">交易日期</td>
        <td width="86%"><%out.print(vc.get(6));%></td>
    </tr>
    <tr>
        <td width="14%">交易时间</td>
        <td width="86%"><%out.print(vc.get(7));%></td>
    </tr>
    <tr>
        <td width="14%">交易流水号</td>
        <td width="86%"><%out.print(vc.get(8));%></td>
    </tr>
    <tr>
        <td width="14%">交易结果</td>
        <td width="86%"><%out.print(vc.get(9));%>&nbsp;&nbsp;&nbsp;[1:成功]</td>
    </tr>
    <tr>
        <td width="14%">手续费总额</td>
        <td width="86%"><%out.print(vc.get(10));%></td>
    </tr>
    <tr>
        <td width="14%">银行卡类型</td>
        <td width="86%"><%out.print(vc.get(11));%>&nbsp;&nbsp;&nbsp;[0:借记卡 1:准贷记卡 2:贷记卡]</td>
    </tr>
    <tr>
        <td width="14%">银行备注</td>
        <td width="86%"><%out.print(vc.get(12));%></td>
    </tr>
    <tr>
        <td width="14%">错误信息描述</td>
        <td width="86%"><%out.print(vc.get(13));%></td>
    </tr>
</table>
<p>&nbsp;&nbsp;&nbsp;</p>
</body>
</html>

```

## 5 商户联机交易 API 编程

商户除了可以登录到企业网银,进行 B2C 的支付管理外,我行支付平台还提供联机 API 服务。具有一定开发能力的第三方支付平台和商户,可通过其财务软件调用我行 API 接口实现实时、自动的订单查询和对账等业务。

商户与银行支付系统间的通讯接口均采用统一的 Https 协议,商户需打开相关的网络和端口限制。

### 5.1 API 调用接口

商户 API 接口提供了 BOCOMB2CClient 类,它包含了可调用的全部交易请求接口的方法,每个交易请求接口都返回 BOCOMB2COPReply 类型的对象,该对象包含了交易返回的全部数据,通过调用该方法可得到所有交易返回数据。

#### 5.1.1 BOCOMB2CClient 类

Java 实现类:com.BOCOM.netpay.b2cAPI.BOCOMB2CClient

该类的成员方法用于提供全部 B2C 商户联机交易接口的实现。交易接口方法的返回数据都为 BOCOMB2COPReply 类型,每个联机交易的接口方法可实现:请求数据的打包、签名、数据向交行支付平台的发送、响应的接收、解包、签名的校验等。最后再将响应数据生成一个 BOCOMB2COPReply 的实例并将该实例返回。

#### 1 初始化方法-1

调用该方法完成交通银行 API 组件的初始化,传入参数为配置文件 B2CMerchant.xml 的绝对路径,该方法只需要调用一次,再次调用方法直接返回。

```
int initialize(String fileName); //0:成功; 其他:出错
```

#### 2 初始化方法-2

调用该方法完成交通银行 API 组件的初始化,传入参数为 XML 配置文件的输入流,该方法只需要调用一次,再次调用方法直接返回。

```
int initialize(InputStream in); //0:成功; 其他:出错
```

### 3 强制初始化方法-1

系统在运行过程中，若需要重新初始化配置文件，可调用该方法。

```
int force_initialize(String fileName)    //0:成功; 其他:出错
```

### 4 强制初始化方法-2

系统在运行过程中，若需要重新初始化配置文件，可调用该方法。

```
int force_initialize(InputStream in)    //0:成功; 其他:出错
```

### 5 错误查询

若调用交易接口的函数返回为 null，可通过此函数获得错误信息。

```
String getLastErr()    //返回错误信息
```

### 6 结算账户余额查询接口

根据商户号查询商户结算帐户的余额。返回数据参看 [5.2](#)（交易接口返回 XML 包）。

```
BOCOMB2COPReply QueryAccountBalance();
```

### 7 批量订单查询接口

查询指定多笔订单号的支付结果，返回数据参看 [5.2](#)（交易接口返回 XML 包）。

```
BOCOMB2COPReply queryOrder(String orders)    //orders 是订单信息，多笔订单不超过 20 笔，之间用 ‘|’ 分隔；如 “10001|10002|10003|10004”
```

### 8 当批订单查询接口

查询当日多笔订单的支付结果，每次最多返回 20 条记录，返回数据参看 [5.2](#)（交易接口返回 XML 包）。

```
BOCOMB2COPReply QueryCurOrder(  
    String begTime,    //开始时间[yyyyMMddHHmmss]  
    String endTime,    //结束时间[yyyyMMddHHmmss]  
    int flag,    //订单状态 0 所有 1 已支付 2 已撤销  
    int begIndex,    //起始索引号，可为 NULL,默认填 0  
    String begOrder,    //起始订单号，可为 NULL，若输入需和截止定单号关  
    //联使用  
    String endOrder,    //截至订单号，可为 NULL
```

```
int sortField,           //排序字段 1, 订单号; 2, 金额; 3, 时间
int sortOrder)          // 1: 降序; 2, 升序
```

## 9 历史订单查询接口

查询多笔历史订单的支付结果，每次最多返回 20 条记录，返回数据参看 [5.2](#) (交易接口返回 XML 包)。

```
BOCOMB2COPReply QueryHistoryOrder(
    String begDate,       //开始时间[yyyyMMddHHmmss]
    String endDate,       //结束时间[yyyyMMddHHmmss]
    int flag              //订单状态 0 所有 1 已支付 2 已撤销 3 部分退货 4 退
                          //货处理中 5 全部退货
    int beginindex,       //起始索引号，可为 NULL，默认填 0
    String begOrder,      //起始订单号，可为 NULL，若输入，需和截止定单号关联
                          //使用
    String endOrder,      //截至订单号，可为 NULL
    int sortField,        //排序字段 1, 订单号; 2, 金额; 3, 时间
    int sortOrder)        // 1: 降序; 2, 升序
```

## 10 退款明细查询接口

商户查询退款信息，每次最多返回 20 条记录，返回数据参看 [5.2](#) (交易接口返回 XML 包)。

```
BOCOMB2COPReply QueryRefund(
    String begDate,       //(录入)开始日期[yyyyMMdd]
    String endDate,       //(录入)结束日期[yyyyMMdd]
    int refundtype        //退款类型：99 所有 1 支付撤销、2 退款
    String order,         //订单号，可为 NULL
    int flag              //交易状态：99 - 所有; 1 - 成功; 2 - 失败
    int begIndex          //起始索引号，默认填 0
)
```

## 11 退款录入接口

商户录入退款信息，返回数据参看 [5.2](#) (交易接口返回 XML 包)。

```
BOCOMB2COPReply Refund(
    String operator       //商户操作员号
    String order          //订单号
    String date           //订单日期[yyyyMMdd]
    String amount         //退款金额，可为 NULL，若部分退款需填
```

```
String comment //退款备注
)
```

## 12 对账单下载接口

商户系统调用该交易接口，输入日期，下载结算明细文件到本地目录，返回数据参看 [5.2](#) (交易接口返回 XML 包)。

```
BOCOMB2COPReply DownLoadSettlement(
    String date) //指定清算日期[yyyyMMdd]
```

## 13 持卡人身份验证 (VIP 商户)

商户系统调用该交易接口，输入银行卡号、客户名称 (非必输)、证件类型 (非必输)、证件号 (非必输)；该功能专为 VIP 商户设置；客户名称和证件类型、号码至少二选一输入，也可同时输入，返回数据参看 [5.2](#) (交易接口返回 XML 包)。

```
BOCOMB2COPReply verifyCustID (
    String cardNo, //银行卡号
    String custName, //客户名称
    String certType, //证件类型, 可以为空
    String certNo) //证件号, 可以为空
```

其中，证件类型有：

代码	名称
15	居民身份证
16	临时身份证
17	军人身份证件
18	武警身份证件
19	港澳台居民通行证
20	护照
21	其他
22	临时户口
23	户口簿
24	边境证

## 14 当批订单完整查询接口

查询当日多笔订单的支付结果，一次返回全部记录，返回数据参看 [5.2](#) (交易接口返回 XML 包)。

```
BOCOMB2COPReply QueryCurOrderFull(
    String begTime, //开始时间[yyyyMMddHHmmss]
    String endTime, //结束时间[yyyyMMddHHmmss]
```

int flag	//订单状态 0 所有 1 已支付 2 已撤销
String begOrder,	//起始订单号,可为 NULL,若输入,需和截止定单号
<b>关联使用</b>	
String endOrder,	//截至订单号,可为 NULL
int sortField,	//排序字段 1, 订单号; 2, 金额; 3, 时间
int sortOrder)	// 1: 降序; 2, 升序

## 5.1.2 BOCOMB2COPReply 类

Java 实现类：com.BOCOM.netpay.b2cAPI.BOCOMB2COPReply

类 BOCOMB2COPReply 用来处理交易返回的 XML 数据包,BOCOMB2COPReply 解析 XML 数据包,并保存响应数据项,包括错误码、错误信息、响应结果和多笔结果明细数据。以“订单查证”的返回明细 XML 包为例：

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
    <opResult>
      <merchantID>商户号</ merchantID >
      <signData>签名数据</signData> <!--对 opResultSe 中全部数据进行签名-->
      <cert>证书</cert>
    </opResult>
    <opResultSet>
      <opResult>
        <order>订单号</order>
        <state>订单状态</state>
        <paySerial>支付流水号</paySerial>
        <currType>币种</currType>
        <amount>金额</amount>
        <date>订单日期</date>
        <time>订单时间</time>
      </opResult>
    </opResultSet>
  </opRep>
</BOCOMB2C>
```

其中，<opResultSet>和</opResultSet>之间的内容称为交易结果集，<opResult>和</opResult>之间的内容称为交易结果。一个交易结果集中可以有一个或多个交易结果。交易结果可以在交易结果集之外，它是所有交易结果的公共信息，如本例。交易结果中的一个标签只与一个值对应，例如“<order>订单号</order>”，“order”就是数据名称，“订单号”就是值。



## 1. 获得交易返回码

String getRetCode() //若返回是"0"：正确；其他：有错误

## 2. 获得交易错误信息

String getErrorMessage() //在错误代码不为"0"时，得到错误信息

## 3. 获得交易结果

OpResultSet getResult()

## 4. 获得交易结果集

OpResult getResultSet()

### 5.1.3 其他 Java 类接口

#### 1. OpResultSet 类

响应数据结果集，方法：

int getOpresultNum(); //获得响应结果集中的结果个数

OpResult getOpresult(int i); //根据索引得到相应结果

String getResultValueByName(int index,String name); //根据结果集中的结果索引和数据名称得到相应的数据值

#### 2. OpResult 类

响应数据结果，方法：

int getNamedValueNum() ;//获得响应结果中的数据项个数

NamedValue getNamedValue(int i) ; //根据索引得到相应数据项

String getValueByName(String name); //根据名称得到相应数据项的值

#### 3. NamedValue 类

一项响应数据，成员变量：

name ：数据名称

value : 值

## 5.2 交易接口返回 XML 包

我行的支付系统对商户通过 API 发起的交易请求给予的回应都以 XML 报文方式返回。  
在 API 提供的交易接口中，已对交易响应的 XML 报文进行了解析。

### 5.2.1 结算账户余额查询

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
    <opResult>
      <merchantID>商户号</ merchantID >
      <settAccount>结算账号</settAccount>
      <accountName>账户名称</accountName>
      <currType>币种</currType>
      <currBalance>当前余额</currBalance>
      <validBalance>可用余额</validBalance>
    </opResult>
  </opRep>
  <signData>签名数据</signData>
</BOCOMB2C>
```

### 5.2.2 批量订单查询

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
    <opResult>
      <merchantID>商户号</ merchantID >
    </opResult>
    <opResultSet>
      <opResult>
        <order>订单号</order>
        <orderDate>订单日期</orderDate>
        <orderTime>订单时间</orderTime>
      </opResult>
    </opResultSet>
  </opRep>
</BOCOMB2C>
```

```

        <curType>币种 CNY (人民币)</curType>
        <amount>金额</amount>
        <tranDate>支付日期</tranDate>
        <tranTime>支付时间</tranTime>
        <tranState>支付交易状态 1[成功]</tranState>
        <orderState>订单状态[0 未支付、1 已支付、2 已撤销、3 已部分退货、4
退货处理中、5 已全额退货]</orderState>
        <fee>手续费</fee>
        <bankSerialNo>银行流水号</bankSerialNo>
        <bankBatNo>银行批次号</bankBatNo>
        <cardType>交易卡类型[0:借记卡 1:准贷记卡 2:贷记卡]</cardType>
        <merchantBatNo>商户批次号</merchantBatNo>
        <merchantComment>商户备注</merchantComment>
        <bankComment>银行备注</bankComment>
    </opResult>
</opResultSet>
</opRep>
<signData>签名数据</signData>
</BOCOMB2C>

```

### 5.2.3 当批订单查询

```

<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
    <opRep>
        <retCode>返回码</retCode>
        <errMsg>错误描述</errMsg>
        <opResult>
            <merchantID>商户号</ merchantID >
            <totalNumber>总笔数</totalNumber>
            <beginIndex>本次查询返回记录起始号</beginIndex>
        </opResult>
    </opResultSet>
    <opResult>
        <order>订单号</order>
        <orderDate>订单日期</orderDate>
        <orderTime>订单时间</orderTime>
        <curType>币种 CNY (人民币)</curType>
        <amount>金额</amount>
        <tranDate>支付日期</tranDate>
        <tranTime>支付时间</tranTime>
        <tranState>支付交易状态 1[成功]</tranState>
        <orderState>订单状态[0 未支付、1 已支付、2 已撤销、3 已部分退货、4

```

```

退货处理中、5 已全额退货]</orderState>
    <fee>手续费</fee>
    <bankSerialNo>银行流水号</bankSerialNo>
    <bankBatNo>银行批次号</bankBatNo>
    <cardType>交易卡类型[0:借记卡 1:准贷记卡 2:贷记卡]</cardType>
    <merchantBatNo>商户批次号</merchantBatNo>
    <merchantComment>商户备注</merchantComment>
    <bankComment>银行备注</bankComment>
  </opResult>
</opResultSet>
</opRep>
<signData>签名数据</signData>
</BOCOMB2C>

```

## 5.2.4 历史订单查询查询

```

<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
    <opResult>
      <merchantID>商户号</ merchantID >
      <totalNumber>总笔数</totalNumber>
      <beginIndex>本次查询返回记录起始号</beginIndex>
    </opResult>
    <opResultSet>
      <opResult>
        <order>订单号</order>
        <orderDate>订单日期</orderDate>
        <orderTime>订单时间</orderTime>
        <curType>币种 CNY ( 人民币 ) </curType>
        <amount>金额</amount>
        <tranDate>支付日期</tranDate>
        <tranTime>支付时间</tranTime>
        <tranState>支付交易状态 1[成功]</tranState>
        <orderState>订单状态[0 未支付、1 已支付、2 已撤销、3 已部分退货、4
退货处理中、5 已全额退货]</orderState>
        <fee>手续费</fee>
        <bankSerialNo>银行流水号</bankSerialNo>
        <bankBatNo>银行批次号</bankBatNo>
        <cardType>交易卡类型[0:借记卡 1:准贷记卡 2:贷记卡]</cardType>
        <merchantBatNo>商户批次号</merchantBatNo>

```

```
<merchantComment>商户备注</merchantComment>
<bankComment>银行备注</bankComment>
</opResult>
</opResultSet>
</opRep>
<signData>签名数据</signData>
</BOCOMB2C>
```

### 5.2.5 退款明细查询

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
    <opResult>
      <merchantID>商户号</merchantID>
      <totalNumber>总笔数</totalNumber>
      <startIndex>本次查询返回记录起始号</startIndex>
    </opResult>
    <opResultSet>
      <opResult>
        <order>订单号</order>
        <orderDate>订单日期</orderDate>
        <curType>币种 CNY</curType>
        <amount>退款金额</amount>
        <refundType>退款类型 1 支付撤销、2 退款</refundType>
        <date>退款日期</date>
        <time>退款时间</time>
        <state>退款交易状态 1 - 成功 ; 2 - 失败</state>
        <reject>退款拒绝原因</reject>
        <fee>退款手续费</fee>
        <account>退款账号</account>
        <merchantComment>商户备注</merchantComment>
        <bankComment>银行备注</bankComment>
      </opResult>
    </opResultSet>
  </opRep>
  <signData>签名数据</signData>
</BOCOMB2C>
```

## 5.2.6 退款录入

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
    <opResult>
      <merchantID>商户号</merchantID>
      <serial>退款流水号</serial>
      <account>退款账号</account>
    </opResult>
  </opRep>
  <signData>签名数据</signData>
</BOCOMB2C>
```

## 5.2.7 对帐单下载

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
    <opResult>
      <merchantID>商户号</merchantID>
      <totalSum>支付结算总金额</totalSum>
      <totalNumber>支付结算总笔数</totalNumber>
      <totalFee>手续费</totalFee>
      <settlementFile>对帐文件内容</settlementFile>
    </opResult>
  </opRep>
  <signData>签名数据</signData>
</BOCOMB2C>
```

## 5.2.8 持卡人身份验证（VIP 商户使用）

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <opRep>
    <retCode>返回码</retCode>
    <errMsg>错误描述</errMsg>
```

```

        <opResult>
        </opResult>
    </opRep>
    <signData>签名数据</signData>
</BOCOMB2C>

```

## 5.2.9 当批订单完整查询

```

<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
    <opRep>
        <retCode>返回码</retCode>
        <errMsg>错误描述</errMsg>
        <opResult>
            <merchantID>商户号</ merchantID >
        </opResult>
        <opResultSet>
            <opResult>
                <order>订单号</order>
                <orderDate>订单日期</orderDate>
                <orderTime>订单时间</orderTime>
                <curType>币种 CNY ( 人民币 ) </curType>
                <amount>金额</amount>
                <tranDate>支付日期</tranDate>
                <tranTime>支付时间</tranTime>
                <tranState>支付交易状态 1[成功]</tranState>
                <orderState>订单状态[0 未支付、1 已支付、2 已撤销、3 已部分退货、4
退货处理中、5 已全额退货]</orderState>
                <fee>手续费</fee>
                <bankSerialNo>银行流水号</bankSerialNo>
                <bankBatNo>银行批次号</bankBatNo>
                <cardType>交易卡类型[0:借记卡 1：准贷记卡 2:贷记卡]</cardType>
                <merchantBatNo>商户批次号</merchantBatNo>
                <merchantComment>商户备注</merchantComment>
                <bankComment>银行备注</bankComment>
            </opResult>
        </opResultSet>
    </opRep>
    <signData>签名数据</signData>
</BOCOMB2C>

```

## 5.3 API 联机交易签名与验签

API 交易中发送请求数据的签名与接收银行响应数据的验签，都由 API 内部封装并处理，不需要商户系统做其他处理。

## 5.4 程序示例

详细编程规范请参照 demo 下的相应 JSP 页面

### 5.4.1 结算账号余额查询

```
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<html>
<head>
<title>结算帐户查询测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String code,err,msg;
    BOCOMB2CClient client = new BOCOMB2CClient();
    int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");//该代码只需调用一次
    if (ret != 0){ //初始化失败
        out.print("初始化失败,错误信息："+client.getLastErr());
    }
    else{
        BOCOMB2COPReply rep = client.queryAccountBalance();//结算帐户查询
        if (rep == null){
            err = client.getLastErr();
            out.print("交易错误信息："+err + "<br>");
        }
        else{
            code = rep.getRetCode();//得到交易返回码
            msg = rep.getErrorMessage();
            out.print("交易返回码："+code + "<br>");
            out.print("交易错误信息："+msg + "<br>");
            if("0".equals(code)){//表示交易成功
                out.print("<br>-----<br>");
                OpResult opr = rep.getOpResult();
            }
        }
    }
}%>
```



名称

余额

```
String accountNo = opr.getValueByName("settAccount"); //得到账号
String accountName = opr.getValueByName("accountName"); // 得到 账 号

String currType = opr.getValueByName("currType"); //得到币种
String currBalance = opr.getValueByName("currBalance"); //得到当前余额
String validBalance = opr.getValueByName("validBalance"); // 得到 可 用

out.print("账号:"+accountNo);
out.print("<br>");
out.print("账号名称:"+accountName);
out.print("<br>");
out.print("币种:"+currType);
out.print("<br>");
out.print("当前余额:"+currBalance);
out.print("<br>");
out.print("可用余额:"+validBalance);
out.print("<br>");
out.print("<p></p>");
    }
}
}

%>

<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>
```

## 5.4.2 批量订单查询

```
<% @ page import="com.bocom.netpay.b2cAPI.*"%>
<% @ page language="java" contentType="text/html; charset=GBK"%>
<html>
<head>
<title>批量订单查询测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String orders = request.getParameter("orders");
    String detail = request.getParameter("detail");
```

```

int showdetail = 0;
String code,err,msg;
BOCOMB2CCClient client = new BOCOMB2CCClient();
int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");//该代码只需调用一次
if (ret != 0){ //初始化失败
    out.print("初始化失败,错误信息 : "+client.getLastErr());
}else {
    try{
        showdetail = Integer.parseInt(detail);
    }
    catch(Exception xcp){ showdetail = 0;}
    BOCOMB2COPReply rep = client.queryOrder(orders); //批量订单查询
    if (rep == null) {
        err = client.getLastErr();
        out.print("交易错误信息 : " + err + "<br>");
    } else {
        code = rep.getRetCode(); //得到交易返回码
        err = rep.getLastErr();
        msg = rep.getErrorMessage();
        out.print("交易返回码 : " + code + "<br>");
        out.print("交易错误信息 : " + msg + "<br>");
        if("0".equals(code)){//表示交易成功
            String num;
            int index;
            OpResultSet oprSet = rep.getOpResultSet();
            int iNum = oprSet.getOpresultNum();
            out.print("总交易记录数 : ");
            out.print(iNum);
            out.print("<br>-----<br>");
            for (index = 0; index <= iNum - 1; index++) {
                String order =
oprSet.getResultValueByName(index,"order"); //订单号
                String orderDate =
oprSet.getResultValueByName(index,
"orderDate"); //订单日期
                String orderTime =
oprSet.getResultValueByName(index,
"orderTime"); //订单时间
                String curType =
oprSet.getResultValueByName(index,"curType"); //币种
                String amount =
oprSet.getResultValueByName(index,"amount"); //金额
                String tranDate =
oprSet.getResultValueByName(index,"tranDate"); //交易日期
                String tranTime =
oprSet.getResultValueByName(index,"tranTime"); //交易时间
            }
        }
    }
}

```

```

        String tranState      =
oprSet.getResultValueByName(index,"tranState");      //支付交易状态
        String orderState    =      oprSet.getResultValueByName(index,
"orderState");    //订单状态
        String fee            =      oprSet.getResultValueByName(index,
"fee");          //手续费
        String bankSerialNo  =      oprSet.getResultValueByName(index,
"bankSerialNo"); //银行流水号
        String bankBatNo     =      oprSet.getResultValueByName(index,
"bankBatNo");    //银行批次号
        String cardType      =      oprSet.getResultValueByName(index,
"cardType");     //交易卡类型 0:借记卡 1：准贷记卡 2:贷记卡
        String merchantBatNo =      oprSet.getResultValueByName(index,
"merchantBatNo"); //商户批次号
        String merchantComment = oprSet.getResultValueByName(index,
"merchantComment");//商户备注
        String bankComment   =      oprSet.getResultValueByName(index,
"bankComment");  //银行备注

        out.print("订单号：" + order);
        out.print("<br>");
        out.print("订单日期：" + orderDate);
        out.print("<br>");
        out.print("订单时间：" + orderTime);
        out.print("<br>");
        out.print("币种：" + curType);
        out.print("<br>");
        out.print("金额：" + amount);
        out.print("<br>");
        out.print("交易日期：" + tranDate);
        out.print("<br>");
        out.print("交易时间：" + tranTime);
        out.print("<br>");
        out.print("支付交易状态[1:成功]：" + tranState);
        out.print("<br>");
        out.print("定单状态[0 所有 1 已支付 2 已撤销 3 部分退货 4 退货
处理中 5 全部退货]：" + orderState);
        out.print("<br>");
        out.print("手续费：" + fee);
        out.print("<br>");
        out.print("银行流水号：" + bankSerialNo);
        out.print("<br>");
        out.print("银行批次号：" + bankBatNo);
        out.print("<br>");
        out.print("交易卡类型[0:借记卡 1：准贷记卡 2:贷记卡]：" +

```

```
cardType);

        out.print("<br>");
        out.print("商户批次号：" + merchantBatNo);
        out.print("<br>");
        out.print("商户备注：" + merchantComment);
        out.print("<br>");
        out.print("银行备注：" + bankComment);
        out.print("<p></p>");
    }
}
}
}

%>
<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>
```

### 5.4.3 当批订单查询

```
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<html>
<head>
<title>当批订单查询测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String begTime;    //开始时间[yyyyMMddHHmmss]
    String endTime;    //结束时间[yyyyMMddHHmmss]
    int flag = 0;       //订单状态 0 所有 1 已支付 2 已撤销 3 部分退货 4 退货处理中
5 全部退货
    int begIndex = 0;   //起始索引号，可为 NULL,默认填 0
    String begOrder;    //起始订单号，可为 NULL
    String endOrder;    //截至订单号，可为 NULL
    int sortField = 1;  //排序字段 1，订单号；2，金额；3，时间
    int sortOrder = 1;  //1：降序；2：升序
    String code;
    String err ;
    String msg ;
```

```
begTime = request.getParameter("begTime");
endTime = request.getParameter("endTime");
try{
    flag = Integer.parseInt(request.getParameter("flag"));
}
catch(Exception xcp){flag = 0;}
try{
    begIndex = Integer.parseInt(request.getParameter("begIndex"));
}
catch(Exception xcp){begIndex = 0;}
begOrder = request.getParameter("begOrder");
endOrder = request.getParameter("endOrder");
try{
    sortField = Integer.parseInt(request.getParameter("sortField"));
}
catch(Exception xcp){sortField = 1;}
try{
    sortOrder = Integer.parseInt(request.getParameter("sortOrder"));
}
catch(Exception xcp){sortOrder = 1;}

BOCOMB2CClient client = new BOCOMB2CClient();
int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");//该代码只需调用一次
if (ret != 0){ //初始化失败
    out.print("初始化失败,错误信息 : "+client.getLastErr());
}
else{
    BOCOMB2COPReply                                rep                                =
client.queryCurOrder(begTime,endTime,flag,begIndex,begOrder,endOrder,sortField,sortOrder);
    if (rep == null){
        err = client.getLastErr();
        out.print("交易错误信息 : "+err + "<br>");
    }else{
        code = rep.getRetCode() ;//得到交易返回码
        msg = rep.getErrorMessage();
        out.print("交易返回码 : "+code + "<br>");
        out.print("交易错误信息 : "+msg + "<br>");
        if("0".equals(code)){//表示交易成功
            String num ;
            int iNum = 0;
            int index =0;
            OpResult opr = rep.getOpResult();
            String total = opr.getValueByName("totalNumber");//得到返回记录个数
            OpResultSet oprSet = rep.getOpResultSet();
```

```

        iNum = oprSet.getOprresultNum();
        out.print("总交易记录数：");
        out.print(total);
        out.print("<br>本次返回记录数：");
        out.print(iNum);
        out.print("<br>-----<br>");
        for (index =0;index<=iNum-1;index++){
            String order          =
oprSet.getResultValueByName(index,"order");          //订单号
            String orderDate      =          oprSet.getResultValueByName(index,
"orderDate");    //订单日期
            String orderTime      =          oprSet.getResultValueByName(index,
"orderTime");    //订单时间
            String curType        =
oprSet.getResultValueByName(index,"curType");          //币种
            String amount         =
oprSet.getResultValueByName(index,"amount");          //金额
            String tranDate       =
oprSet.getResultValueByName(index,"tranDate");          //交易日期
            String tranTime       =
oprSet.getResultValueByName(index,"tranTime");          //交易时间
            String tranState      =
oprSet.getResultValueByName(index,"tranState");          //支付交易状态
            String orderState     =          oprSet.getResultValueByName(index,
"orderState");    //订单状态
            String fee            =          oprSet.getResultValueByName(index,
"fee");            //手续费
            String bankSerialNo   =          oprSet.getResultValueByName(index,
"bankSerialNo");  //银行流水号
            String bankBatNo      =          oprSet.getResultValueByName(index,
"bankBatNo");     //银行批次号
            String cardType       =          oprSet.getResultValueByName(index,
"cardType");      //交易卡类型 0:借记卡 1：准贷记卡 2:贷记卡
            String merchantBatNo  =          oprSet.getResultValueByName(index,
"merchantBatNo"); //商户批次号
            String merchantComment =          oprSet.getResultValueByName(index,
"merchantComment");//商户备注
            String bankComment    =          oprSet.getResultValueByName(index,
"bankComment");  //银行备注
            out.print("订单号：" + order);
            out.print("<br>");
            out.print("订单日期：" + orderDate);
            out.print("<br>");
            out.print("订单时间：" + orderTime);

```

```

        out.print("<br>");
        out.print("币种：" + curType);
        out.print("<br>");
        out.print("金额：" + amount);
        out.print("<br>");
        out.print("交易日期：" + tranDate);
        out.print("<br>");
        out.print("交易时间：" + tranTime);
        out.print("<br>");
        out.print("支付交易状态[1:成功]：" + tranState);
        out.print("<br>");
        out.print("定单状态[0 所有 1 已支付 2 已撤销 3 部分退货 4 退货
处理中 5 全部退货]：" + orderState);
        out.print("<br>");
        out.print("手续费：" + fee);
        out.print("<br>");
        out.print("银行流水号：" + bankSerialNo);
        out.print("<br>");
        out.print("银行批次号：" + bankBatNo);
        out.print("<br>");
        out.print("交易卡类型[0:借记卡 1：准贷记卡 2:贷记卡]：" +
cardType);

        out.print("<br>");
        out.print("商户批次号：" + merchantBatNo);
        out.print("<br>");
        out.print("商户备注：" + merchantComment);
        out.print("<br>");
        out.print("银行备注：" + bankComment);
        out.print("<p></p>");
    }
}
}
}
%>
<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>

```

#### 5.4.4 历史订单查询查询

```

<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>

```

```
<html>
<head>
<title>历史订单查询测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String begDate;    //开始时间[yyyyMMddHHmmss]
    String endDate;    //结束时间[yyyyMMddHHmmss]
    int flag;          //订单状态 0 所有 1 已支付 2 已撤销 3 部分退货 4 退货处理中
5 全部退货
    int begIndex;      //起始索引号, 可为 NULL,默认填 0
    String begOrder;   //起始订单号, 可为 NULL
    String endOrder;   //截至订单号, 可为 NULL
    int sortField;     //排序字段 1, 订单号; 2, 金额; 3, 时间
    int sortOrder;     //1: 降序; 2, 升序
    String code;
    String err;
    String msg;
    begDate = request.getParameter("begDate");
    endDate = request.getParameter("endDate");
    try{
        flag = Integer.parseInt(request.getParameter("flag"));
    }
    catch(Exception xcp){flag = 0;}
    try{
        begIndex = Integer.parseInt(request.getParameter("begIndex"));
    }
    catch(Exception xcp){begIndex = 0;}
    begOrder = request.getParameter("begOrder");
    endOrder = request.getParameter("endOrder");
    try{
        sortField = Integer.parseInt(request.getParameter("sortField"));
    }
    catch(Exception xcp){sortField = 1;}
    try{
        sortOrder = Integer.parseInt(request.getParameter("sortOrder"));
    }
    catch(Exception xcp){sortOrder = 1;}

    BOCOMB2CClient client = new BOCOMB2CClient();
    int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");//该代码只需调用一次
    if (ret != 0){ //初始化失败
```



```

        out.print("初始化失败,错误信息 : "+client.getLastErr());
    }
    else{
        BOCOMB2COPReply                                rep                                =
client.queryHistoryOrder(begDate,endDate,flag,begIndex,begOrder,endOrder,sortField,sortOrder)
;

        if (rep == null){
            err = client.getLastErr();
            out.print("交易错误信息 : "+err + "<br>");
        }else{
            code = rep.getRetCode() ;//得到交易返回码
            err = rep.getLastErr();
            msg = rep.getErrorMessage();
            out.print("交易返回码 : "+code + "<br>");
            out.print("交易错误信息 : "+msg + "<br>");
            if("0".equals(code)){//表示交易成功
                String num ;
                int iNum = 0;
                int index =0;
                OpResult opr = rep.getOpResult();
                String total = opr.getValueByName("totalNumber"); //得到返回记录个数
                OpResultSet oprSet = rep.getOpResultSet();
                iNum = oprSet.getOpresultNum();
                out.print("总交易记录数 : ");
                out.print(total);
                out.print("<br>本次返回记录数 : ");
                out.print(iNum);
                out.print("<br>-----<br>");
                for (index =0;index<=iNum-1;index++){
                    String order                                =
oprSet.getResultValueByName(index,"order");                                //订单号
                    String orderDate                                =                                oprSet.getResultValueByName(index,
"orderDate");                                //订单日期
                    String orderTime                                =                                oprSet.getResultValueByName(index,
"orderTime");                                //订单时间
                    String curType                                =                                =
oprSet.getResultValueByName(index,"curType");                                //币种
                    String amount                                =                                =
oprSet.getResultValueByName(index,"amount");                                //金额
                    String tranDate                                =                                =
oprSet.getResultValueByName(index,"tranDate");                                //交易日期
                    String tranTime                                =                                =
oprSet.getResultValueByName(index,"tranTime");                                //交易时间
                    String tranState                                =

```

```

oprSet.getResultValueByName(index,"tranState");           //支付交易状态
        String orderState = oprSet.getResultValueByName(index,
"orderState");           //订单状态
        String fee = oprSet.getResultValueByName(index,
"fee");           //手续费
        String bankSerialNo = oprSet.getResultValueByName(index,
"bankSerialNo"); //银行流水号
        String bankBatNo = oprSet.getResultValueByName(index,
"bankBatNo"); //银行批次号
        String cardType = oprSet.getResultValueByName(index,
"cardType"); //交易卡类型 0:借记卡 1：准贷记卡 2:贷记卡
        String merchantBatNo = oprSet.getResultValueByName(index,
"merchantBatNo"); //商户批次号
        String merchantComment = oprSet.getResultValueByName(index,
"merchantComment");//商户备注
        String bankComment = oprSet.getResultValueByName(index,
"bankComment"); //银行备注

        out.print("订单号：" + order);
        out.print("<br>");
        out.print("订单日期：" + orderDate);
        out.print("<br>");
        out.print("订单时间：" + orderTime);
        out.print("<br>");
        out.print("币种：" + curType);
        out.print("<br>");
        out.print("金额：" + amount);
        out.print("<br>");
        out.print("交易日期：" + tranDate);
        out.print("<br>");
        out.print("交易时间：" + tranTime);
        out.print("<br>");
        out.print("支付交易状态[1:成功]：" + tranState);
        out.print("<br>");
        out.print("定单状态[0 所有 1 已支付 2 已撤销 3 部分退货 4 退货
处理中 5 全部退货]：" + orderState);
        out.print("<br>");
        out.print("手续费：" + fee);
        out.print("<br>");
        out.print("银行流水号：" + bankSerialNo);
        out.print("<br>");
        out.print("银行批次号：" + bankBatNo);
        out.print("<br>");
        out.print("交易卡类型[0:借记卡 1：准贷记卡 2:贷记卡]：" +
cardType);
  
```

```
        out.print("<br>");
        out.print("商户批次号：" + merchantBatNo);
        out.print("<br>");
        out.print("商户备注：" + merchantComment);
        out.print("<br>");
        out.print("银行备注：" + bankComment);
        out.print("<p></p>");
    }
}
}
}
%>
<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>
```

### 5.4.5 退款明细查询

```
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<html>
<head>
<title>退款明细查询测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String begDate = request.getParameter("begDate");
    String endDate = request.getParameter("endDate");
    int refundtype = 99;
    try{
        refundtype = Integer.parseInt(request.getParameter("refundtype"));
    }
    catch(Exception xcp){refundtype = 0;}
    String order = request.getParameter("order");
    int flag = 0;
    try{
        flag = Integer.parseInt(request.getParameter("flag"));
    }
    catch(Exception xcp){flag = 0;}
    int begIndex = 0;
```

```

try{
    begIndex = Integer.parseInt(request.getParameter("begIndex"));
}
catch(Exception xcp){begIndex = 0;}
String code ;
String err ;
String msg ;
BOCOMB2CCClient client = new BOCOMB2CCClient();
int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");
if (ret != 0){ //初始化失败
    out.print("初始化失败,错误信息 : "+client.getLastErr());
}
else {
    BOCOMB2COPReply rep = client.queryRefund( begDate, endDate,
refundtype,order,flag,begIndex) ;//退款明细查询
    if (rep == null){
        err = client.getLastErr();
        out.print("交易错误信息 : "+err + "<br>");
    }
    else{
        code = rep.getRetCode(); //得到交易返回码
        msg = rep.getErrorMessage(); //得到交易返回信息
        out.print("交易返回码 : "+code + "<br>");
        out.print("交易错误信息 : "+msg + "<br>");
        if("0".equals(code)){//表示交易成功
            int num ;
            String index ;
            OpResult opr = rep.getOpResult();
            String total = opr.getValueByName("totalNumber"); //得到返回记录个数
            OpResultSet oprSet = rep.getOpResultSet();
            int iNum = oprSet.getOpresultNum();
            int iIndex = 0;
            int iTotal = Integer.parseInt(total);
            out.print("总交易记录数 : ");
            out.print(total);
            out.print("<br>本次返回记录数 : ");
            out.print(iNum);
            out.print("<br>-----<br>");
            for (iIndex=0;iIndex<=iNum-1;iIndex++) {
                String order1 = oprSet.getResultValueByName
(iIndex,"order"); //订单号
                String orderDate = oprSet.getResultValueByName
(iIndex,"orderDate"); //订单日期
                String curType = oprSet.getResultValueByName

```

第 53 页/共 60 页

## 5.4.6 退款录入

```
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<html>
<head>
<title>退款录入测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
String operator = request.getParameter("operator");
String order = request.getParameter("order");
String orderdate = request.getParameter("date");
String amount = request.getParameter("amount");
String comment = request.getParameter("comment");
String code ;
String err ;
String msg ;
BOCOMB2CCClient client = new BOCOMB2CCClient();
int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");
if (ret != 0){ //初始化失败
    out.print("初始化失败,错误信息 : "+client.getLastErr());
}else {
    BOCOMB2COPReply rep = client.Refund( operator,order,orderdate,amount,comment); //退
款录入
    if (rep == null){
        err = client.getLastErr();
        out.print("交易错误信息 : "+err + "<br>");
    }else{
        code = rep.getRetCode(); //得到交易返回码
        err = rep.getLastErr();
        msg = rep.getErrorMessage();
        out.print("交易返回码 : "+code + "<br>");
        out.print("交易错误信息 : "+msg + "<br>");
        if("0".equals(code)){//表示交易成功
            out.print("<br>-----<br>");
            OpResult opr = rep.getOpResult();
            String serial = opr.getValueByName ("serial"); //退款流水号
            String account = opr.getValueByName ("account"); //退款账号

            out.print("退款流水号"+serial);
        }
    }
}
```

```
        out.print("<br>");
        out.print("退款账号"+account);
        out.print("<br>");
        out.print("<p></p>");
    }
}
}
%>
<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>
```

### 5.4.7 对帐单下载

```
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<html>
<head>
<title>对帐单下载测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String settleDate = request.getParameter("settleDate");
    String code,err,msg;
    BOCOMB2CCClient client = new BOCOMB2CCClient();
    int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");
    if (ret != 0){    //初始化失败
        out.print("初始化失败,错误信息："+client.getLastErr());
    }
    else{
        BOCOMB2COPReply rep = client.downloadSettlement(settleDate) ;//对帐单下载
        if (rep == null){
            err = client.getLastErr();
            out.print("交易错误信息："+err + "<br>");
        }else{
            code = rep.getRetCode(); //得到交易返回码
            err = rep.getLastErr();
            msg = rep.getErrorMessgae();
            out.print("交易返回码："+code + "<br>");
            out.print("交易错误信息："+msg + "<br>");
        }
    }
%>
```

```
        if("0".equals(code)){//表示交易成功
            out.print("<br>-----<br>");
            OpResult opr = rep.getOpResult();
            String totalSum = opr.getValueByName("totalSum");//总金额
            String totalNumber = opr.getValueByName("totalNumber");//总笔数
            String totalFee = opr.getValueByName("totalFee");//总手续费

            out.print("总金额"+totalSum);
            out.print("<br>");
            out.print("总笔数"+totalNumber);
            out.print("<br>");
            out.print("总手续费"+totalFee);
            out.print("<br>");
            out.print("<p></p>");
        }
    }
}

%>
<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>
```

### 5.4.8 持卡人身份验证

```
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<html>
<head>
<title>持卡人身份验证(VIP 商户</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String card = request.getParameter("card");
    String custName = request.getParameter("custName");
    String certType = request.getParameter("certType");
    String certNo = request.getParameter("certNo");
    String code;
    String err;
    String msg;
    BOCOMB2CClient client = new BOCOMB2CClient();
```



```
int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");
if (ret != 0){ //初始化失败
    out.print("初始化失败,错误信息："+client.getLastErr());
}
else{
    BOCOMB2COPReply rep = client.verifyCustID( card, custName,certType,certNo) ;
    if (rep == null){
        err = client.getLastErr();
        out.print("交易错误信息："+err + "<br>");
    }else{
        code = rep.getRetCode(); //得到交易返回码
        err = rep.getLastErr();
        msg = rep.getErrorMessage();
        out.print("交易返回码："+code + "<br>");
        out.print("交易错误信息："+msg + "<br>");
    }
}
}
%>
<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>
```

## 5.4.9 当日订单完整查询

```
<% @ page import="com.bocom.netpay.b2cAPI.*" %>
<% @ page language="java" contentType="text/html; charset=GBK" %>
<html>
<head>
<title>当批订单完整查询测试</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>

<body bgcolor="#FFFFFF" text="#000000">
<%
    String begTime; //开始时间[yyyyMMddHHmmss]
    String endTime; //结束时间[yyyyMMddHHmmss]
    int flag = 0; //订单状态 0 所有 1 已支付 2 已撤销 3 部分退货 4 退货处理中
5 全部退货
    String begOrder; //起始订单号，可为 NULL
    String endOrder; //截至订单号，可为 NULL
    int sortField = 1; //排序字段 1，订单号；2，金额；3，时间
    int sortOrder = 1; // 1：降序；2，升序
```

```
String code;
String err ;
String msg ;
begTime      = request.getParameter("begTime");
endTime      = request.getParameter("endTime");
try{
    flag      = Integer.parseInt(request.getParameter("flag"));
}
catch(Exception xcp){flag = 0;}
begOrder     = request.getParameter("begOrder");
endOrder     = request.getParameter("endOrder");
try{
    sortField  = Integer.parseInt(request.getParameter("sortField"));
}
catch(Exception xcp){sortField = 1;}
try{
    sortOrder  = Integer.parseInt(request.getParameter("sortOrder"));
}
catch(Exception xcp){sortOrder = 1;}

BOCOMB2CCClient client = new BOCOMB2CCClient();
int ret = client.initialize("C:\\bocommjava\\ini\\B2CMerchant.xml");//该代码只需调用一次
if (ret != 0){ //初始化失败
    out.print("初始化失败,错误信息 : "+client.getLastErr());
}
else{
    BOCOMB2COPReply      rep
client.queryCurOrderFull(begTime,endTime,flag,begOrder,endOrder,sortField,sortOrder);
    if (rep == null){
        err = client.getLastErr();
        out.print("交易错误信息 : "+err + "<br>");
    }else{
        code = rep.getRetCode() ;//得到交易返回码
        err = rep.getLastErr();
        msg = rep.getErrorMessage();
        out.print("交易返回码 : "+code + "<br>");
        out.print("交易错误信息 : "+msg + "<br>");
        if("0".equals(code)){//表示交易成功
            String num ;
            int iNum = 0;
            int index =0;
            OpResult opr = rep.getOpResult();
            OpResultSet oprSet = rep.getOpResultSet();
            iNum = oprSet.getOpresultNum();
        }
    }
}
```

```

        out.print("<br>本次返回记录数：");
        out.print(iNum);
        out.print("<br>-----<br>");
        for (index =0;index<=iNum-1;index++){
            String order                =
oprSet.getResultValueByName(index,"order");           //订单号
            String orderDate            =           oprSet.getResultValueByName(index,
"orderDate");    //订单日期
            String orderTime            =           oprSet.getResultValueByName(index,
"orderTime");    //订单时间
            String curType              =
oprSet.getResultValueByName(index,"curType");          //币种
            String amount               =
oprSet.getResultValueByName(index,"amount");           //金额
            String tranDate             =
oprSet.getResultValueByName(index,"tranDate");         //交易日期
            String tranTime             =
oprSet.getResultValueByName(index,"tranTime");         //交易时间
            String tranState            =
oprSet.getResultValueByName(index,"tranState");        //支付交易状态
            String orderState           =           oprSet.getResultValueByName(index,
"orderState");    //订单状态
            String fee                  =           oprSet.getResultValueByName(index,
"fee");           //手续费
            String bankSerialNo        =           oprSet.getResultValueByName(index,
"bankSerialNo"); //银行流水号
            String bankBatNo           =           oprSet.getResultValueByName(index,
"bankBatNo");    //银行批次号
            String cardType             =           oprSet.getResultValueByName(index,
"cardType");     //交易卡类型 0:借记卡 1：准贷记卡 2:贷记卡
            String merchantBatNo       =           oprSet.getResultValueByName(index,
"merchantBatNo"); //商户批次号
            String merchantComment      =           oprSet.getResultValueByName(index,
"merchantComment");//商户备注
            String bankComment         =           oprSet.getResultValueByName(index,
"bankComment"); //银行备注
            out.print("订单号：" + order);
            out.print("<br>");
            out.print("订单日期：" + orderDate);
            out.print("<br>");
            out.print("订单时间：" + orderTime);
            out.print("<br>");
            out.print("币种：" + curType);
            out.print("<br>");

```

```

        out.print("金额：" + amount);
        out.print("<br>");
        out.print("交易日期：" + tranDate);
        out.print("<br>");
        out.print("交易时间：" + tranTime);
        out.print("<br>");
        out.print("支付交易状态[1:成功]：" + tranState);
        out.print("<br>");
        out.print("定单状态[0 所有 1 已支付 2 已撤销 3 部分退货 4 退货
处理中 5 全部退货]：" + orderState);
        out.print("<br>");
        out.print("手续费：" + fee);
        out.print("<br>");
        out.print("银行流水号：" + bankSerialNo);
        out.print("<br>");
        out.print("银行批次号：" + bankBatNo);
        out.print("<br>");
        out.print("交易卡类型[0:借记卡 1：准贷记卡 2:贷记卡]：" +
cardType);

        out.print("<br>");
        out.print("商户批次号：" + merchantBatNo);
        out.print("<br>");
        out.print("商户备注：" + merchantComment);
        out.print("<br>");
        out.print("银行备注：" + bankComment);
        out.print("<p></p>");
    }
}
}
}
%>
<p><a href="Index.htm">返回首页</a></p>
<p>&nbsp;</p>
</body>
</html>

```