



交通银行网上支付平台 商户端开发编程说明

API.Com 用户

(版本号 v1.0)

交通银行

目录

1. 开发包说明.....	4
2. 商户证书下载与更新.....	4
2-1. 银行申请.....	4
2-2. 商户证书下载.....	4
2-3. 商户证书更新.....	9
3. 证书与开发包安装.....	10
3-1. 安装前检查.....	10
3-2. 开发包安装.....	10
3-3. 配置参数文件.....	12
3-4. 证书相关放置与安装.....	12
4. 商户 B2C 订单支付编程	14
4-1. 支付交易流程.....	14
4-1-1. 商户订单接收处理.....	15
4-1-2. 网上支付首页面处理.....	15
4-1-3. 支付验证处理.....	16
4-1-4. 通知商户支付结果.....	16
4-2. 提交订单.....	17
4-2-1. 订单接口：.....	17
4-2-2. 签名方式：.....	20
4-2-3. 程序示例：.....	20
4-3. 获取支付结果.....	23
4-3-1. 获取支付结果接口：.....	23
4-3-2. 验签方式：.....	24
4-3-3. 程序示例：.....	24
5. 商户联机交易 API 编程	27
5-1. API 组件调用接口	27
5-2. 交易接口返回 XML 包.....	31
5-2-1. 结算账户余额查询.....	31
5-2-2. 批量订单查询.....	32
5-2-3. 当日订单查询.....	32
5-2-4. 历史订单查询.....	33
5-2-5. 退款明细查询.....	34
5-2-6. 退款录入.....	34
5-2-7. 对账单下载.....	35
5-2-8. 持卡人身份验证（VIP 商户使用）	35
5-3. API 联机交易签名与验签	36
5-4. 程序示例.....	37
5-4-1. 结算帐户余额查询.....	37
5-4-2. 批量订单查询.....	38
5-4-3. 当日订单查询.....	41
5-4-4. 历史订单查询.....	44
5-4-5. 退款明细查询.....	48

5-4-6.	退款录入.....	50
5-4-7.	对帐单下载.....	51
5-4-8.	持卡人身份验证.....	52

1. 开发包说明

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

- cert 提供商户端测试环境的商户测试证书、银行端测试环境测试根证书及银行端生产环境根证书；
- demo 存放交易演示 Asp 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。
- setup 存放 API 的安装文件。
- lib 提供商户编程 API 所需全部 DLL 文件；
 - B2CClientCOMCtrl.dll 联机交易开发编程 DLL 库；
 - SIGN.dll 签名与验签 API 库
 - 其余为 SIGN.dll 依赖库以及 WINDOWS 依赖库

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，主要用于在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”的权限，然后该指定用户登录企业网银，通过使用密码信封下载商户证书。

2-2. 商户证书下载

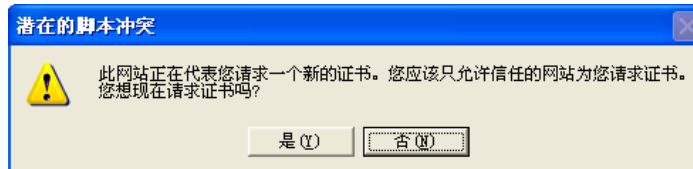
商户在配置和调试 API 前，应先下载商户在我行申请的商户证书，并导出到应用服务器上。

◇ 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，在菜单中选择“**商户配置**”->“**商户证书下载**”，进入证书下载页面：



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



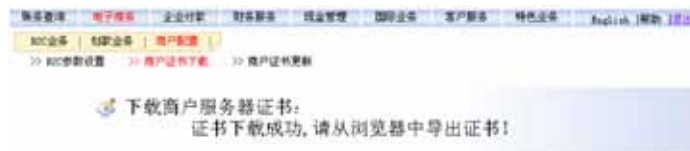
选择安全级别为“高”，点击“下一步”按钮



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 B2CMerchant.xml 中（配置方式见后）。



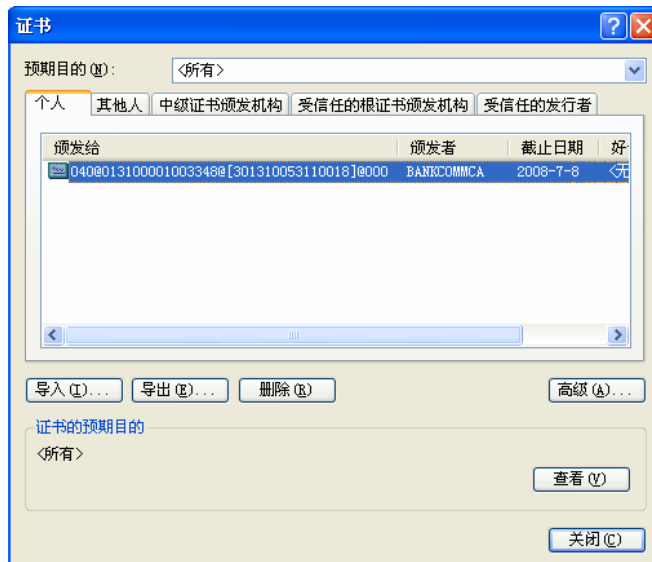
点击“确定”，完成证书下载



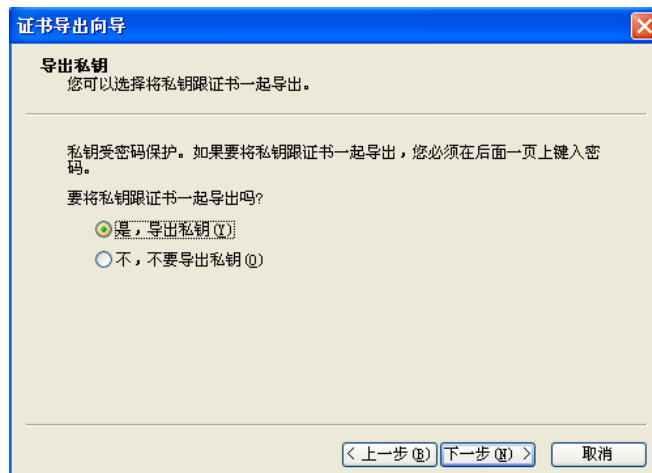
◇ 导出

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。

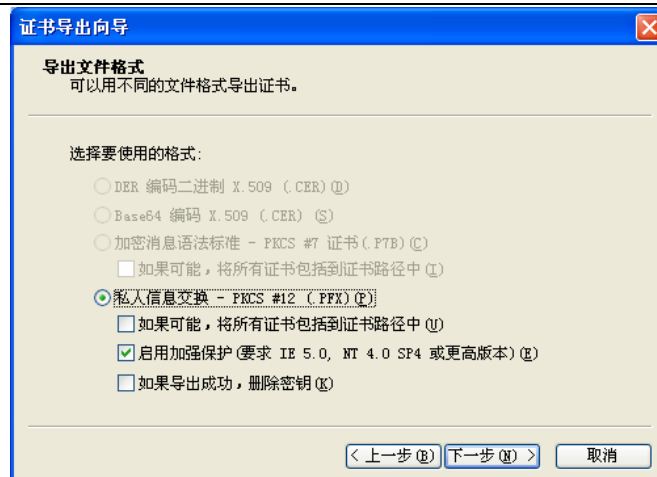
在 IE 浏览器中选择“工具”->“Internet 选项”->“内容”->“证书”->“个人”，
后便可看到商户证书：



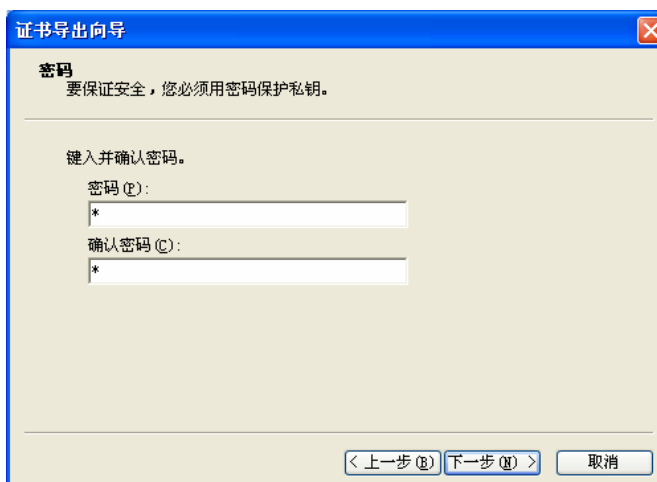
选择商户证书，点击“导出”按钮，可将该证书导出



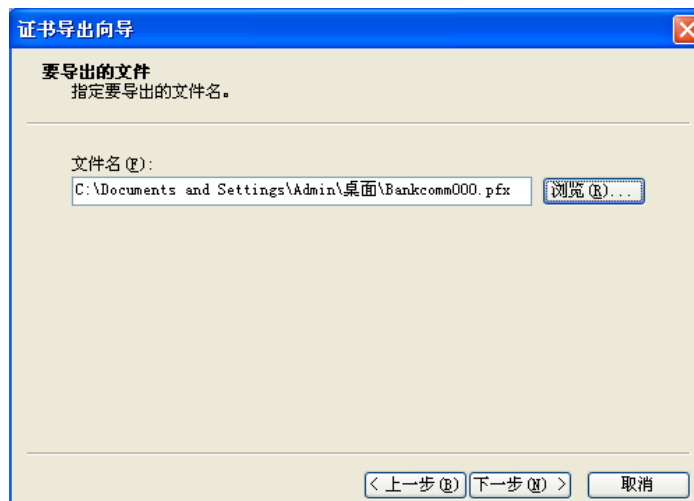
注意导出时应选择“是，导出私钥 (Y)”



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, 完成证书导出。在商户证书导出时, 可重新

再设置一次商户证书密码。



2-3. 商户证书更新

商户证书更新主要是为因故遗失、损坏证书的商户提供证书自行更新服务。商户下载新的证书，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：



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

3. 证书与开发包安装

3-1. 安装前检查

首先,请查看待安装环境的[系统根目录](#)。查看方式:在 Windows 的命令行下输入'set'并回车,查看环境变量为:'SystemRoot' 的值,通常为:'C:\WINDOWS' 或 'C:\WINNT'。

3-2. 开发包安装

安装方式 1: 将开发包中 Lib 目录下包含 B2CClientCOMCtrl.dll 的所有 DLL 文件复制到[系统根目录](#)\system32 目录下,并在命令行中执行 "regsvr32 B2CClientCOMCtrl.dll" 注册该 DLL。

安装方式 2: 运行开发包中 setup 目录下的 setup.exe 进行安装。安装文件将会复制所有文件并注册 DLL。



3-3. 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 的所有接口将返回错误。

将开发包中的 bocomm 目录复制到 C:\ 根目录下，进入 bocomm\ini 目录下，编辑 B2CMerchant.xml 文件。文件内容示例如下：

```
<?xml version="1.0" encoding = "GB2312"?>
<BOCOMB2C>
  <ApiURL> https://61.172.242.229/corporbank/NsTrans </ApiURL> <!--通讯方式 ( http /
https ) -->
  <OrderURL>https://61.172.242.228/netpay/MerPayB2C</OrderURL><!-- 订单支付请求
URL-->
  <!-- 商户可修改参数 --->
  <EnableLog>false</EnableLog> <!-- 日志开关( true:表示写日志,false: 表示不写日志 )-->
  <LogPath>C:\bocomm\log</LogPath> <!-- 日志文件存放目录-->
  <SettlementFilePath>C:\bocomm\settlement</SettlementFilePath> <!-- 下载结算明细文件
存放目录-->
  <MerchantCertFile>C:\bocomm\cert\301310070116554.PFX</MerchantCertFile><!-- .Com
环境必须:商户证书文件 ( 绝对路径 ) -->
  <MerchantCertPassword>11111111</MerchantCertPassword> <!-- .COM 环境必须:商户
证书私钥加密密码-->
  <RootCertFile>C:\bocomm\cert\BOCTestCA.cer</RootCertFile> <!--.COM 环境必须:根证
书文件-->
</BOCOMB2C>
```

删除的内容: ConnectMeth

o

删除的内容: d

删除的内容: ConnectMetho

d

带格式的: 字体颜色: 黑色

3-4. 证书相关放置与安装

✧ 放置商户证书

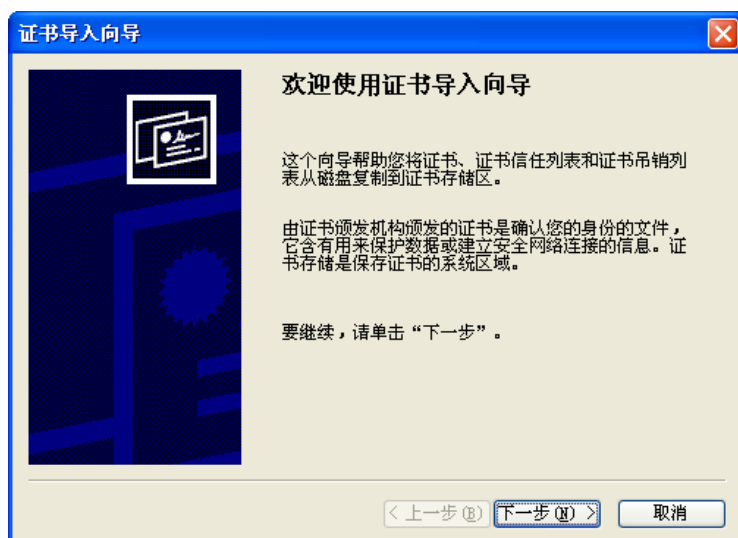
将交通银行提供的根证书（cer 格式文件）和从网银上下载的商户证书（pfx 格式文件）移动到 C:\bocomm\cert 目录下。

✧ 安装银行根证书

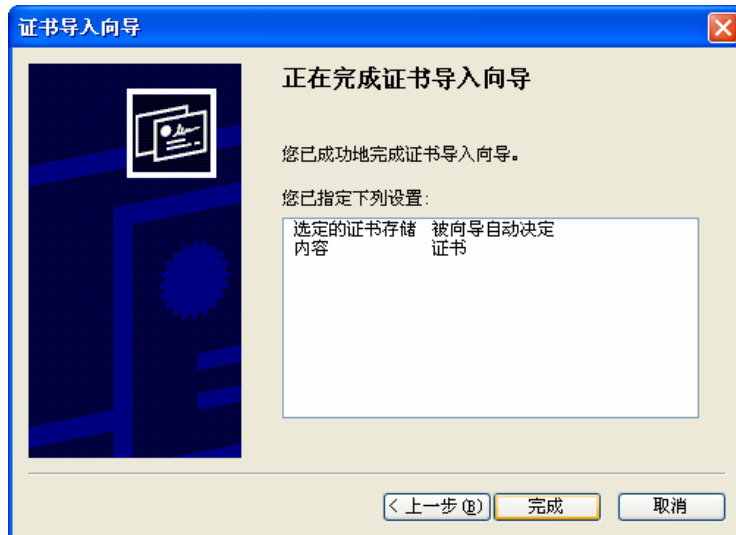
鼠标双击银行根证书文件，如下：



点击“安装证书(I)...”按钮；



继续多次点击“下一步”按钮，得到如下界面：

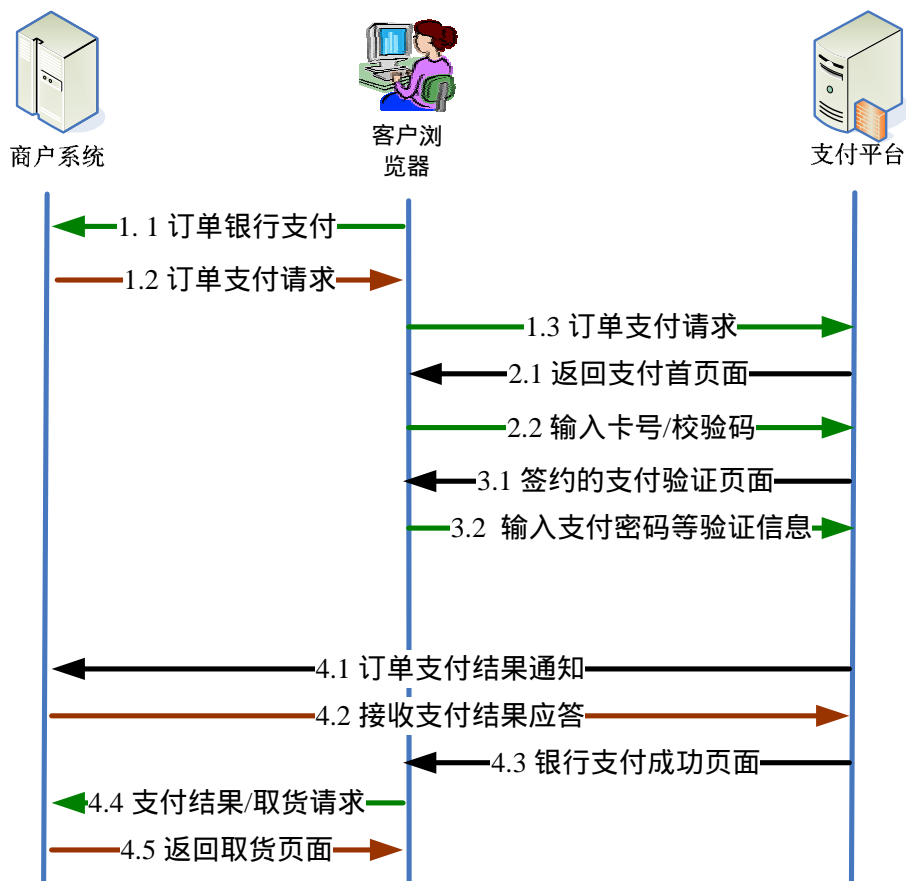


点击“完成”按钮，完成根证书的安装。

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。因不同的版本的额度及校验方式均有所不同，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	必填	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 3 个月以上的唯一性
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.	代理商家名称	N	proxyMerName	30	二级商户编号/或证件号码
18.	代理商家证件类型	N	proxyMerType	4	代理商家证件类型
19.	代理商家证件号码	N	proxyMerCredentials	20	代理商家证件号码
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 detech 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：

```
<form name="form1" method="post" action="https://xxx/netpay/MerPayB2C">
```

```
<!-- 消息版本号 固定填写 1.0.0.0-->
```

```
<input type="hidden" name="interfaceVersion" value="1.0.0.0">
```

```
<!-- 商户代码 -->
```

```
<input type="hidden" name="merID" maxlength='15' value="">
```

```
<!-- 订单号 -->
```

```
<input type="hidden" name="orderid" maxlength='20' value="">
```

```
<!-- 商户订单日期 yyyymmdd -->
```

```
<input type="hidden" name="orderDate" value="">
```

```
<!-- 商户订单时间 hhmmss -->
```

```
<input type="hidden" name="orderTime" value="">

<!-- 交易类别 固定填写 0:B2C -->

<input type="hidden" name="tranType" value="0" >

<!-- 订单金额 -->

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

<!-- 交易币种 固定填写 CNY:人民币-->

<input type="hidden" name="curType" value="CNY">

<!-- 订单内容 -->

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

<!-- 商家备注 -->

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

<!-- 物流配送标志 不做物流配送则填写 0-->

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

<!-- 通知方式 0 不通知 1 通知 2 抓取页面 -->

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

<!-- 主动通知 URL -->

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

<!-- 取货 URL -->

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

<!-- 自动跳转时间（秒） -->

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

<!-- 商户批次号商家对账使用 -->

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

<!-- 代理商家名称 -->

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

<!-- 代理商家类型 -->

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

<!-- 代理商家证件号码 -->

<input type="hidden" name="proxyMerCredentials" value="">
```

```
<!-- 渠道编号 0:html 渠道 -->

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

<!-- 商家签名 -->

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

</form>
```

4-2-2. 签名方式：

将 1 至 20 号域以 | 分隔 后，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

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：

```
<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" onload = "javascript: form1.submit()">
    <%
      Dim interfaceVersion
      Dim merID
      Dim orderid
      Dim orderDate
      Dim orderTime
      Dim tranType
      Dim amount
      Dim curType
      Dim orderContent
      Dim orderMono
      Dim phdFlag
      Dim notifyType
```

```
Dim merURL
Dim goodsURL
Dim jumpSeconds
Dim payBatchNo
Dim proxyMerName
Dim proxyMerType
Dim proxyMerCredentials
Dim netType
Dim sourceMsg
interfaceVersion = Request("interfaceVersion")

orderid = Request("orderid")
orderDate = Request("orderDate")
orderTime = Request("orderTime")
tranType = Request("tranType")
amount = Request("amount")
curType = Request("curType")
orderContent = Request("orderContent")
orderMono = Request("orderMono")
phdFlag = Request("phdFlag")
notifyType = Request("notifyType")
merURL = Request("merURL")
goodsURL = Request("goodsURL")
jumpSeconds = Request("jumpSeconds")
payBatchNo = Request("payBatchNo")
proxyMerName = Request("proxyMerName")
proxyMerType = Request("proxyMerType")
proxyMerCredentials = Request("proxyMerCredentials")
netType = Request("netType")
Dim ret
Dim signData
Set B2CAPIATL = Server.CreateObject("B2CCClientCOMCtrl.B2CCClientCOM")
ret = B2CAPIATL.Initialize("C:\bocomm\ini\B2CMerchant.xml") // 初始化
merID = B2CAPIATL.GetMerchantID()

sourceMsg = interfaceVersion & "|" & merID & "|" & orderid & "|" & orderDate & "|" &
orderTime & "|" & tranType & "|" & amount & "|" & curType & "|" & orderContent & "|" &
orderMono & "|" & phdFlag & "|" & notifyType & "|" & merURL & "|" & goodsURL & "|" &
jumpSeconds & "|" & payBatchNo & "|" & proxyMerName & "|" & proxyMerType & "|" &
proxyMerCredentials & "|" & netType
signData = B2CAPIATL.Sign_detachsign(sourceMsg)
%>

<form name = "form1" method = "post" action =
"<% Response.write(B2CAPIATL.GetOrderReqURL())%>">
```

```
<input type = "hidden" name = "interfaceVersion" value =
"<%Response.write(interfaceVersion)%>">
    <input type = "hidden" name = "merID" value = "<%Response.write(merID)%>">
    <input type = "hidden" name = "orderid" value =
"<%Response.write(orderid)%>">
    <input type = "hidden" name = "orderDate" value =
"<%Response.write(orderDate)%>">
    <input type = "hidden" name = "orderTime" value =
"<%Response.write(orderTime)%>">
    <input type = "hidden" name = "tranType" value =
"<%Response.write(tranType)%>">
    <input type = "hidden" name = "amount" value =
"<%Response.write(amount)%>">
    <input type = "hidden" name = "curType" value =
"<%Response.write(curType)%>">
    <input type = "hidden" name = "orderContent" value =
"<%Response.write(orderContent)%>">
    <input type = "hidden" name = "orderMono" value =
"<%Response.write(orderMono)%>">
    <input type = "hidden" name = "phdFlag" value =
"<%Response.write(phdFlag)%>">
    <input type = "hidden" name = "notifyType" value =
"<%Response.write(notifyType)%>">
    <input type = "hidden" name = "merURL" value =
"<%Response.write(merURL)%>">
    <input type = "hidden" name = "goodsURL" value =
"<%Response.write(goodsURL)%>">
    <input type = "hidden" name = "jumpSeconds" value =
"<%Response.write(jumpSeconds)%>">
    <input type = "hidden" name = "payBatchNo" value =
"<%Response.write(payBatchNo)%>">
    <input type = "hidden" name = "proxyMerName" value =
"<%Response.write(proxyMerName)%>">
    <input type = "hidden" name = "proxyMerType" value =
"<%Response.write(proxyMerType)%>">
    <input type = "hidden" name = "proxyMerCredentials" value =
"<%Response.write(proxyMerCredentials)%>">
    <input type = "hidden" name = "netType" value =
"<%Response.write(netType)%>">
    <input type = "hidden" name = "merSignMsg" value =
"<%Response.write(signData)%>">
</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	
4.	交易币种	3	TRANCURRTYPE	CNY(156)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。

程序示例：

```
<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>
<body bgcolor = "#FFFFFF" text = "#000000">
    <%
        Response.Write("商户结果页面")
        Response.Write("<br>")
        Response.Write("-----")
        Response.Write("<br>")
        Dim result
        Dim arr
        Dim Length
        Dim result1
        Dim lastIndex
        Set APICtrl = Server.CreateObject("B2CCClientCOMCtrl.B2CCClientCOM")
        ret = APICtrl.Initialize("C:\\bocomm\\ini\\B2CMerchant.xml") //初始化
```



```
//以上代码执行一次
If ret <> 0 Then
    //初始化失败
    Response.Write("初始化失败,错误代码：" & ret)
    Err = APICtrl.GetLastError()
    Response.Write(Err)
End If
Length = 0
result = Request("notifyMsg")

If Not IsNull(result) And Not IsEmpty(result) Then
    lastIndex = InstrRev(result, "|")
    signMsg = Mid(result, lastIndex + 1)
    srcMsg = Mid(result, 1, lastIndex)

End If
Dim ret
Dim msg
ret = APICtrl.Sign_detachverify(srcMsg, signMsg)
If (ret <> 0) Then
    Response.Write("验签失败" + ret)
    msg = APICtrl.GetErrorMessage()
    Dim Err
    Err = APICtrl.GetLastError()
    Response.Write("交易错误信息" + msg)
    Response.Write(Err)
Else
    arr = Split(srcMsg, "|")
    %>
    <table width="75%" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td width="14%">商户客户号</td>
            <td width="86%"><% Response.write(arr(0)) %></td>
        </tr>
        <tr>
            <td width="14%">订单编号</td>
            <td width="86%"><% Response.write(arr(1)) %></td>
        </tr>
        <tr>
            <td width="14%">交易金额</td>
            <td width="86%"><% Response.write(arr(2)) %></td>
        </tr>
        <tr>
            <td width="14%">交易币种</td>
```

```

        <td width="86%"><%Response.write(arr(3))%></td>
    </tr>
    <tr>
        <td width="14%">平台批次号</td>
        <td width="86%"><%Response.write(arr(4))%></td>
    </tr>
    <tr>
        <td width="14%">商户批次号</td>
        <td width="86%"><%Response.write(arr(5))%></td>
    </tr>
    <tr>
        <td width="14%">交易日期</td>
        <td width="86%"><%Response.write(arr(6))%></td>
    </tr>
    <tr>
        <td width="14%">交易时间</td>
        <td width="86%"><%Response.write(arr(7))%></td>
    </tr>
    <tr>
        <td width="14%">交易流水号</td>
        <td width="86%"><%Response.write(arr(8))%></td>
    </tr>
    <tr>
        <td width="14%">交易结果</td>
        <td width="86%"><%Response.write(arr(9))%></td>
    </tr>
    <tr>
        <td width="14%">手续费总额</td>
        <td width="86%"><%Response.write(arr(10))%></td>
    </tr>
    <tr>
        <td width="14%">银行卡类型</td>
        <td width="86%"><%Response.write(arr(11))%></td>
    </tr>
    <tr>
        <td width="14%">银行备注</td>
        <td width="86%"><%Response.write(arr(12))%></td>
    </tr>
    <tr>
        <td width="14%">错误信息描述</td>
        <td width="86%"><%Response.write(arr(13))%></td>
    </tr>
</table>
<%End if%>

```

```
<p>
    &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 联机接口实现实时、自动的订单查询和对账等业务。

商户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 组件调用接口

商户还可以通过银行提供的 COM 组件接口,调用 API 联机交易。COM 组件名称: B2CClientCOMCtrl.B2CClientCOM,提供的 COM 组件接口如下:

1 初始化

int Initialize(BSTR filename)

从商户指定配置文件中获得初始化参数;返回 0:成功,其他:出错

2 得到订单支付请求 URL

BSTR GetOrderReqURL()

得到交通银行 B2C 订单支付接收 URL

3 得到 COM 控件错误信息

BSTR GetLastErr()

得到 COM 处理错误后的信息

4 得到银行交易处理返回码

BSTR GetRetCode()

若返回是"0"：正确；其他：有错误

5 得到银行交易处理返回错误信息

BSTR GetErrorMessage()

在交易处理返回码不为"0"时，得到错误信息

6 获得交易结果数据值

BSTR GetResultValueByName(BSTR name)

根据数据项名称，从交易返回结果中获得相应的数据值

7 获得交易结果中的记录个数

int GetResultSetSize()

当交易为订单查询并交易成功返回，获得本次返回的记录个数

8 获得交易结果集中的数据值

BSTR GetResultSetValueByName(int index, BSTR name)

根据指定的结果集中的记录索引和数据名称，得到相应的数据值

9 结算帐户余额查询接口

int QueryAccountBalance() // 可为 NULL

商户结算帐户余额查询，也可以查询参数指定的账号，若帐户错误，则查询支付结算帐户信息。返回 0 为交易成功，其他为失败

10 批量订单查询接口

int QueryOrders(BSTR orders)

//要查询的订单笔数，最大 20 笔 //是否查询详细信息，否/

//多笔订单号，之间用 '|' 分隔；如 " 10001|10002|10003|10004 "

返回 0 为交易成功，其他为失败；[返回数据参看 5-2 节](#)；

11 当日订单查询接口

查询当日多笔订单的支付结果，每次最多返回 50 条记录

```
int QueryCurOrder(
    BSTR begTime,           //开始时间[HHmmss]
    BSTR endTime,          //结束时间[HHmmss]
    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，可为空,默认填 0
    BSTR begOrder,          //起始订单号，可为空
    BSTR endOrder,          //截至订单号，可为空
    int sortField,          //排序字段 1，订单号；2，金额；3，时间
    int sortOrder)          // 1：升序；2，降序

返回 0 为交易成功，其他为失败；返回数据参看 5-2 节；
```

12 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 50 条记录

```
int QueryHistoryOrder(
    BSTR begDate,           //开始日期[yyyyMMDD]
    BSTR endDate,          //结束日期[yyyyMMDD]
    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，可为空,默认填 0
    BSTR begOrder,          //起始订单号，可为空
    BSTR endOrder,          //截至订单号，可为空
    int sortField,          //排序字段 1，订单号；2，金额；3，时间
    int sortOrder)          // 1：升序；2，降序

返回 0 为交易成功，其他为失败；返回数据参看 5-2 节；
```

13 退款明细查询接口

商户查询退款信息，每次最多返回 50 条记录

```
int QueryRefund(
    BSTR begDate,           //(录入)开始日期[yyyyMMDD]
    BSTR 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
    BSTR order,             //订单号，可为空
    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
```

)
 返回 0 为交易成功，其他为失败；[返回数据参看 5-2 节](#)；

14 退款录入接口

商户录入退款信息

```
int Refund(
    BSTR operator    //商户操作员号
    BSTR order       //订单号
    BSTR date        //订单日期[yyyyMMDD]
    BSTR amount      //退款金额，可为空，若部分退款需填
    BSTR comment     //退款备注
)
```

返回 0 为交易成功，其他为失败；[返回数据参看 5-2 节](#)；

15 对帐单下载接口

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

```
int DownLoadSettlement(
    BSTR date)    //指定日期[yyyyMMDD]
```

返回 0 为交易成功，其他为失败；[返回数据参看 5-2 节](#)；

16 持卡人身份验证接口

商户系统调用该交易接口，输入银行卡号、客户名称、证件类型（非必输）、证件号（非必输），请求验证持卡人的身份

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

返回 0 为交易成功，其他为失败；[返回数据参看 5-2 节](#)；

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

17 当日订单完整查询接口

查询当日多笔订单的支付结果，一次返回全部记录

```
int QueryCurOrderFull(
    BSTR begTime,          //开始时间[HHmmss]
    BSTR endTime,          //结束时间[HHmmss]
    int flag                //订单状态 0 所有 1 已支付 2 已撤销 3 已退款 ..
    BSTR begOrder,         //起始订单号，可为空
    BSTR endOrder,         //截至订单号，可为空
    int sortField,         //排序字段 1，订单号；2，金额；3，时间
    int sortOrder)        // 1：升序；2，降序
    返回 0 为交易成功，其他为失败；返回数据参看 5-2 节；
```

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>
        <curType>币种</curType>
        <amount>金额</amount>
        <tranDate>支付日期</tranDate>
        <tranTime>支付时间</tranTime>
        <tranState>支付交易状态</tranState> <!-- 1 成功、2 失败、3 未决-->
        <orderState>订单状态</orderState> <!-- 1 已支付 2 已撤销 3 已退款-->
      </opResult>
    </opResultSet>
  </opRep>
  <signData>签名数据</signData>
</BOCOMB2C>
```

5-2-3. 当日订单查询

```
<?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>
      </opResult>
    </opResultSet>
  </opRep>
  <signData>签名数据</signData>
</BOCOMB2C>
```



```
<orderDate>订单日期</orderDate>
<curType>币种</curType>
<amount>金额</amount>
<tranDate>支付日期</tranDate>
<tranTime>支付时间</tranTime>
<tranState>支付交易状态</tranState><!-- 1 成功、2 失败、3 未决-->
<orderState>订单状态</orderState> <!-- 1 已支付 2 已撤销 3 已退
款-->
<fee>手续费</fee>
<cardNo>支付银行卡号</cardNo><!--VIP 服务,可能为空-->
<userName>用户姓名</userName><!--VIP 服务,可能为空-->
<ID>身份证号</ID><!--VIP 服务,可能为空-->
</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>
        <curType>币种</curType>
        <amount>金额</amount>
        <tranDate>支付日期</tranDate>
        <tranTime>支付时间</tranTime>
        <tranState>支付交易状态</tranState><!-- 1 成功、2 失败、3 未决-->
        <orderState>订单状态</orderState> <!-- 1 已支付 2 已撤销 3 已退
款-->
        <fee>手续费</fee>
        <cardNo>支付银行卡号</cardNo><!--VIP 服务,可能为空-->
```

```
<userName>用户姓名</userName><!--VIP 服务,可能为空-->
<ID>身份证号</ID><!--VIP 服务,可能为空-->
</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</curType>
        <amount>退款金额</amount>
        <refundType>退款类型</refundType> <!-- 1 撤销、2 退货-->
        <date>退款日期</date>
        <time>退款时间</time>
        <state>退款交易状态</state> <!-- 1 成功、2 失败、3 未决-->
        <reject>退款拒绝原因</reject>
        <fee>退款手续费</fee>
        <account>退款账号</account>
      </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>
    </opResult>
  </opRep>
```

```
<signData>签名数据</signData>
</BOCOMB2C>
```

5-2-9. 当日订单完整查询

```
<?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>
        <curType>币种</curType>
        <amount>金额</amount>
        <tranDate>支付日期</tranDate>
        <tranTime>支付时间</tranTime>
        <tranState>支付交易状态</tranState><!-- 1 成功、2 失败、3 未决-->
        <orderState>订单状态</orderState> <!-- 1 已支付 2 已撤销 3 已退款-->
        <fee>手续费</fee>
        <cardNo>支付银行卡号</cardNo><!--VIP 服务,可能为空-->
        <userName>用户姓名</userName><!--VIP 服务,可能为空-->
        <ID>身份证号</ID><!--VIP 服务,可能为空-->
      </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

5-4-1. 结算帐户余额查询

```
<%  
Set APICtrl = Server.CreateObject("B2CClientCOMCtrl.B2CClientCOM")  
ret = APICtrl.Initialize("C:\bocomm\ini\B2CMerchant.xml") //初始化  
    //以上代码执行一次  
If ret <> 0 Then  
    //初始化失败  
    Response.Write("初始化失败,错误信息:")  
    Err = APICtrl.GetLastError()  
    Response.Write(Err)  
Else  
    ret = APICtrl.QueryAccountBalance() //结算帐户查询  
    If ret <> 0 Then  
        Dim Err  
        Err = APICtrl.GetLastError()  
        Response.Write("交易错误信息：")  
        Response.Write(Err)  
    Else  
        code = APICtrl.GetRetCode() //得到交易返回码  
        msg = APICtrl.GetErrorMessage()  
        If msg = "" Then  
            msg = "null"  
        End If  
        Response.Write("交易返回码：" + code + "<br>")  
        Response.Write("交易错误信息：" + msg + "<br>")  
        If code = "0" Then  
            Response.Write("<br>-----<br>")  
  
            accountNo = APICtrl.GetResultValueByName("settAccount")  
                //得到账号  
            accountName = APICtrl.GetResultValueByName("accountName")  
                //得到账号名称  
            currType = APICtrl.GetResultValueByName("currType")  
                //得到币种  
            currBalance = APICtrl.GetResultValueByName("currBalance")  
                //得到当前余额  
            validBalance = APICtrl.GetResultValueByName("validBalance")  
                //得到可用余额
```

```
Response.Write("账号:" + accountNo)
Response.Write("<br>")
Response.Write("账号名称:" + accountName)
Response.Write("<br>")
Response.Write("币种:" + currType)
Response.Write("<br>")
Response.Write("当前余额:" + currBalance)
Response.Write("<br>")
Response.Write("可用余额:" + validBalance)
Response.Write("<br>")
Response.Write("<p></p>")
End If
End If
End If
%>
```

5-4-2. 批量订单查询

```
<%
orders = Request("orders")
detail = Request("detail")
ordernum = Request("ordernum")
Set APICtrl = Server.CreateObject("B2CClientCOMCtrl.B2CClientCOM")
ret = APICtrl.Initialize("C:\\bocomm\\ini\\B2CMerchant.xml") //初始化
//以上代码执行一次
If ret <> 0 Then
    //初始化失败
    Err = APICtrl.GetLastError()
    Response.Write("初始化失败,错误信息：")
    Response.Write(Err)

Else
    ret = APICtrl.QueryOrders(orders) //批量订单查询
    If ret <> 0 Then
        Dim Err
        Err = APICtrl.GetLastError()
        Response.Write("交易错误信息：")
        Response.Write(Err)
    Else
        Dim code
        Dim msg
        code = APICtrl.GetRetCode() //得到交易返回码
```

```
msg = APICtrl.GetErrorMessage()
If msg = "" Then
    msg = "null"
End If

Response.Write("交易返回码：" + code + "<br>")
Response.Write("交易错误信息：" + msg + "<br>")
If code = "0" Then
    Dim num
    Dim index
    num = APICtrl.GetResultSetSize() //得到返回记录个数
    Response.Write("总交易记录数：")
    Response.Write(num)
    Response.Write("<br>-----<br>")
    For index = 0 To num - 1
        Order = APICtrl.GetResultSetValueByName (index, "order")
            //订单号
        orderDate= APICtrl.GetResultSetValueByName (index,"orderDate")
            //订单日期
        orderTime=APICtrl.GetResultSetValueByName(index,"orderTime")
            //定单时间
        curType= APICtrl.GetResultSetValueByName (index,"curType")
            //币种
        amount = APICtrl.GetResultSetValueByName (index,"amount")
            //金额
        tranDate = APICtrl.GetResultSetValueByName(index,"tranDate")
            //交易日期
        tranTime = APICtrl.GetResultSetValueByName(index,"tranTime")
            //交易时间
        tranState = APICtrl.GetResultSetValueByName(index,"tranState")
            //支付交易状态
        orderState= APICtrl.GetResultSetValueByName(index,"orderState")
            //订单状态
        fee = APICtrl.GetResultSetValueByName(index, "fee")
            //手续费
        bankSerialNo= APICtrl.GetResultSetValueByName(index,
"bankSerialNo")            //银行流水号
        bankBatNo=APICtrl.GetResultSetValueByName(index, "bankBatNo")
            //银行批次号
        cardType = APICtrl.GetResultSetValueByName(index,"cardType")
            //交易卡类型 0 借记卡 1：准贷记卡 2 贷记卡
        merchantBatNo=APICtrl.GetResultSetValueByName(index,
"merchantBatNo")            //商户批次号
        merchantComment = APICtrl.GetResultSetValueByName(index,
```

```

"merchantComment")      //商户备注
                        bankComment = APICtrl.GetResultSetValueByName(index,
"bankComment")          //银行备注

                        Response.Write("订单号：" + order)
                        Response.Write("<br>")
                        Response.Write("订单日期：" + orderDate)
                        Response.Write("<br>")
                        Response.Write("订单时间：" + orderTime)
                        Response.Write("<br>")
                        Response.Write("币种：" + curType)
                        Response.Write("<br>")
                        Response.Write("金额：" + amount)
                        Response.Write("<br>")
                        Response.Write("交易日期：" + tranDate)
                        Response.Write("<br>")
                        Response.Write("交易时间：" + tranTime)
                        Response.Write("<br>")
                        Response.Write("支付交易状态：" + tranState)
                        Response.Write("<br>")
                        Response.Write("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)
                        Response.Write("<br>")
                        Response.Write("手续费：" + fee)
                        Response.Write("<br>")
                        Response.Write("银行流水号：" + bankSerialNo)
                        Response.Write("<br>")
                        Response.Write("银行批次号：" + bankBatNo)
                        Response.Write("<br>")
                        Response.Write("交易卡类型[0:借记卡 1:准贷记卡 2:贷记卡]："
" + cardType)

                        Response.Write("<br>")
                        Response.Write("商户批次号：" + merchantBatNo)
                        Response.Write("<br>")
                        Response.Write("商户备注：" + merchantComment)
                        Response.Write("<br>")
                        Response.Write("银行备注：" + bankComment)
                        Response.Write("<p></p>")
                        Response.Write("<p></p>")

                        Next
                    End If
                End If
            End If
        %>

```


5-4-3. 当日订单查询

```
<%
Dim begTime      '开始时间[yyyyMMddHHmmss]
Dim endTime      '结束时间[yyyyMMddHHmmss]
Dim flag         '订单状态 0 所有 1 已支付 2 已撤销
Dim begIndex     '起始索引号, 可为 NULL, 默认填 0
Dim begOrder     '起始订单号, 可为 NULL
Dim endOrder     '截至订单号, 可为 NULL
Dim sortField    '排序字段 1, 订单号; 2, 金额; 3, 定单时间
Dim sortOrder    '1: 降序 2: 升序;

begTime      = Request("begTime")
endTime      = Request("endTime")
flag         = Request("flag")
begIndex     = Request("begIndex")
begOrder     = Request("begOrder")
endOrder     = Request("endOrder")
sortField    = Request("sortField")
sortOrder    = Request("sortOrder")

If Not IsNumeric(flag) Then
    flag = 0
End If
If Not IsNumeric(begIndex) Then
    begIndex = 0
End If
If Not IsNumeric(sortField) Then
    sortField = 1
End If
If Not IsNumeric(sortOrder) Then
    sortOrder = 1
End If
If sortField > 3 Then
    sortField = 1
End If
If sortOrder > 2 Then
    sortOrder = 1
End If

Set APICtrl = Server.CreateObject("B2CCClientCOMCtrl.B2CCClientCOM")
ret = APICtrl.Initialize("C:\\bocomm\\ini\\B2CMerchant.xml") //初始化
//以上代码执行一次
```

```
If ret <> 0 Then
    Response.Write("初始化失败,错误信息:")
    Err = APICtrl.GetLastError()
    Response.Write(Err)
Else
    ret = APICtrl.QueryCurOrder(begTime, endTime, Int(flag), Int(begIndex),
begOrder, endOrder, Int(sortField), Int(sortOrder))
    If ret<>0 Then
        Dim Err
        Err = APICtrl.GetLastError()
        Response.Write("交易错误信息：")
        Response.Write(Err)
    Else
        Dim code
        Dim msg
        code = APICtrl.GetRetCode() //得到交易返回码
        msg = APICtrl.GetErrorMessage()
        If msg = "" Then
            msg = "null"
        End If
        Response.Write("交易返回码：" + code + "<br>")
        Response.Write("交易错误信息：" + msg + "<br>")
        If code = "0" Then
            Dim num
            Dim index
            total = APICtrl.GetResultValueByName("totalNumber") //得到返回记录
个数
            num = APICtrl.GetResultSetSize() //得到返回记录个数
            Response.Write("总交易记录数：")
            Response.Write(total)
            Response.Write("<br>本次返回记录数：")
            Response.Write(num)
            Response.Write("<br>-----<br>")
            For index = 0 To num -1
                order = APICtrl.GetResultSetValueByName(index, "order")
                //订单号
                orderDate = APICtrl.GetResultSetValueByName(index,"orderDate")
                //订单日期
                orderTime=APICtrl.GetResultSetValueByName(index, orderTime")
                //订单时间
                curType = APICtrl.GetResultSetValueByName(index,"curType")
                //币种
                amount = APICtrl.GetResultSetValueByName(index,"amount")
                //金额
            Next
        End If
    End If
End If
```

```
tranDate = APICtrl.GetResultSetValueByName(index, "tranDate")
    //交易日期
tranTime = APICtrl.GetResultSetValueByName(index, "tranTime")
    //交易时间
tranState = APICtrl.GetResultSetValueByName(index, "tranState")
    //支付交易状态
orderState= APICtrl.GetResultSetValueByName(index, "orderState")
    //订单状态
fee = APICtrl.GetResultSetValueByName(index, "fee")
    //手续费
bankSerialNo=APICtrl.GetResultSetValueByName(index,
"bankSerialNo")    //银行流水号
bankBatNo=APICtrl.GetResultSetValueByName(index,
"bankBatNo")    //银行批次号
cardType =APICtrl.GetResultSetValueByName(index, "cardType")
    //交易卡类型 0 借记卡 1：准贷记卡 2 贷记卡
merchantBatNo=APICtrl.GetResultSetValueByName(index,
"merchantBatNo")    //商户批次号
merchantComment = APICtrl.GetResultSetValueByName(index,
"merchantComment")    //商户备注
bankComment=APICtrl.GetResultSetValueByName(index,
"bankComment")    //银行备注
Response.Write("订单号：" + order)
Response.Write("<br>")
Response.Write("订单日期：" + orderDate)
Response.Write("<br>")
Response.Write("订单时间：" + orderTime)
Response.Write("<br>")
Response.Write("币种：" + curType)
Response.Write("<br>")
Response.Write("金额：" + amount)
Response.Write("<br>")
Response.Write("交易日期：" + tranDate)
Response.Write("<br>")
Response.Write("交易时间：" + tranTime)
Response.Write("<br>")
Response.Write("支付交易状态[1:成功]：" + tranState)
Response.Write("<br>")
Response.Write("订单状态[0 所有 1 已支付 2 已撤销 ]：" +
orderState)

Response.Write("<br>")
Response.Write("手续费：" + fee)
Response.Write("<br>")
Response.Write("银行流水号：" + bankSerialNo)
```

```

Response.Write("<br>")
Response.Write("银行批次号：" + bankBatNo)
Response.Write("<br>")
Response.Write("交易卡类型[0:借记卡 1:准贷记卡 2:贷记卡]：" + cardType)

Response.Write("<br>")
Response.Write("商户批次号：" + merchantBatNo)
Response.Write("<br>")
Response.Write("商户备注：" + merchantComment)
Response.Write("<br>")
Response.Write("银行备注：" + bankComment)
Response.Write("<p></p>")

Next
End If
End If
End If
%>

```

5-4-4. 历史订单查询

```

<%
Dim begDate '开始日期[yyyyMMddHHmmss]
Dim endDate '结束日期[yyyyMMddHHmmss]
Dim 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
Dim begIndex '起始索引号，可为 NULL,默认填 0
Dim begOrder '起始订单号，可为 NULL
Dim endOrder '截至订单号，可为 NULL
Dim sortField '排序字段 1，订单号；2，金额；3，定单时间
Dim sortOrder '1：desc 降序；2：asc 升序

begDate = Request("begDate")
endDate = Request("endDate")
flag = Request("flag")
begIndex = Request("begIndex")
begOrder = Request("begOrder")
endOrder = Request("endOrder")
sortField = Request("sortField")
sortOrder = Request("sortOrder")

If Not IsNumeric(flag) Then
flag = 0

```

```
End If
If Not IsNumeric(begIndex) Then
    begIndex = 0
End If
If Not IsNumeric(sortField) Then
    sortField = 1
End If
If Not IsNumeric(sortOrder) Then
    sortOrder = 1
End If
If sortField>3 Then
    sortField = 1
End If
If sortOrder>2 Then
    sortOrder = 1
End If

Set APICtrl = Server.CreateObject("B2CClientCOMCtrl.B2CClientCOM")
ret = APICtrl.Initialize("C:\\bocomm\\ini\\B2CMerchant.xml") //初始化
//以上代码执行一次
If ret<>0 Then
    //初始化失败
    Err = APICtrl.GetLastError()
    Response.Write("初始化失败,错误信息：")
    Response.Write(Err)
Else
    ret = APICtrl.QueryHistoryOrder(begDate, endDate, Int(flag), Int(begIndex),
begOrder, endDate, Int(sortField), Int(sortOrder))
    If ret<> 0 Then
        Dim Err
        Err = APICtrl.GetLastError()
        Response.Write("交易错误信息:")
        Response.Write(Err)
    Else
        Dim code
        Dim msg
        code = APICtrl.GetRetCode() //得到交易返回码
        msg = APICtrl.GetErrorMessage()
        If msg = "" Then
            msg = "null"
        End If

        Response.Write("交易返回码：" + code + "<br>")
        Response.Write("交易错误信息：" + msg + "<br>")
    End If
End If
```

个数

```
If code = "0" Then
    Dim num
    Dim index
    total = APICtrl.GetResultValueByName("totalNumber") //得到返回记录

    num = APICtrl.GetResultSetSize() //得到返回记录个数
    Response.Write("总交易记录数：")
    Response.Write(total)
    Response.Write("<br>本次返回记录数：")
    Response.Write(num)
    Response.Write("<br>-----<br>")
    For index = 0 To num - 1
        order = APICtrl.GetResultSetValueByName(index, "order")
            //订单号
        orderDate = APICtrl.GetResultSetValueByName(index, "orderDate")
            //订单日期
        orderTime = APICtrl.GetResultSetValueByName(index, "orderTime")
            //订单时间
        curType = APICtrl.GetResultSetValueByName(index, "curType")
            //币种
        amount = APICtrl.GetResultSetValueByName(index, "amount")
            //金额
        tranDate = APICtrl.GetResultSetValueByName(index, "tranDate")
            //交易日期
        tranTime = APICtrl.GetResultSetValueByName(index, "tranTime")
            //交易时间
        tranState = APICtrl.GetResultSetValueByName(index, "tranState")
            //支付交易状态
        orderState = APICtrl.GetResultSetValueByName(index, "orderState")
            //订单状态
        fee = APICtrl.GetResultSetValueByName(index, "fee")
            //手续费
        bankSerialNo = APICtrl.GetResultSetValueByName(index,
            "bankSerialNo") //银行流水号
        bankBatNo = APICtrl.GetResultSetValueByName(index,
            "bankBatNo") //银行批次号
        cardType = APICtrl.GetResultSetValueByName(index, "cardType")
            //交易卡类型 0 借记卡 1：准贷记卡 2 贷记卡
        merchantBatNo = APICtrl.GetResultSetValueByName(index,
            "merchantBatNo") //商户批次号
        merchantComment = APICtrl.GetResultSetValueByName(index,
            "merchantComment") //商户备注
        bankComment = APICtrl.GetResultSetValueByName(index,
            "bankComment") //银行备注
```

```
Response.Write("订单号：" + order)
Response.Write("<br>")
Response.Write("订单日期：" + orderDate)
Response.Write("<br>")
Response.Write("订单时间：" + orderTime)
Response.Write("<br>")
Response.Write("币种：" + curType)
Response.Write("<br>")
Response.Write("金额：" + amount)
Response.Write("<br>")
Response.Write("交易日期：" + tranDate)
Response.Write("<br>")
Response.Write("交易时间：" + tranTime)
Response.Write("<br>")
Response.Write("支付交易状态[1:成功]：" + tranState)
Response.Write("<br>")
Response.Write("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)
Response.Write("<br>")
Response.Write("手续费：" + fee)
Response.Write("<br>")
Response.Write("银行流水号：" + bankSerialNo)
Response.Write("<br>")
Response.Write("银行批次号：" + bankBatNo)
Response.Write("<br>")
Response.Write("交易卡类型[0:借记卡 1：准贷记卡 2:贷记卡]："
+ cardType)

Response.Write("<br>")
Response.Write("商户批次号：" + merchantBatNo)
Response.Write("<br>")
Response.Write("商户备注：" + merchantComment)
Response.Write("<br>")
Response.Write("银行备注：" + bankComment)
Response.Write("<p></p>")
Next
End If
End If
End If
%>
```

5-4-5. 退款明细查询

```
<%  
  
    begDate = Request("begDate")  
    endDate = Request("endDate")  
    refundtype = Request("refundtype")  
    order = Request("order")  
    flag = Request("flag")  
    begIndex = Request("begIndex")  
  
    Set APICtrl = Server.CreateObject("B2CClientCOMCtrl.B2CClientCOM")  
    ret = APICtrl.Initialize("C:\bocomm\ini\B2CMerchant.xml") //初始化  
    //以上代码执行一次  
    If ret <> 0 Then  
        //初始化失败  
        Response.Write("初始化失败,错误信息 : ")  
        Err = APICtrl.GetLastError()  
        Response.Write(Err)  
    Else  
        If Not IsNumeric(refundtype) Then  
            refundtype = 0  
        End If  
        If Not IsNumeric(flag) Then  
            flag = 0  
        End If  
        If Not IsNumeric(begIndex) Then  
            begIndex = 0  
        End If  
  
        ret = APICtrl.QueryRefund( begDate, endDate, Int(refundtype), order, Int(flag),  
Int(begIndex)) //退款明细查询  
        If ret <> 0 Then  
            Dim Err  
            Err = APICtrl.GetLastError()  
            Response.Write("交易错误信息 : ")  
            Response.Write(Err)  
        Else  
            Dim code  
            Dim msg  
            code = APICtrl.GetRetCode() //得到交易返回码  
            msg = APICtrl.GetErrorMessage()  
            If msg = "" Then  
                msg = "null"  
            End If  
        End If  
    End If  
End If
```


个数

```
End If

Response.Write("交易返回码：" + code + "<br>")
Response.Write("交易错误信息：" + msg + "<br>")
If code = "0" Then
    Dim num
    Dim index
    total = APICtrl.GetResultValueByName("totalNumber") //得到返回记录

    num = APICtrl.GetResultSetSize() //得到返回记录个数
    Response.Write("总交易记录数：")
    Response.Write(total)
    Response.Write("<br>本次返回记录数：")
    Response.Write(num)
    Response.Write("<br>-----<br>")
    For index = 0 To num - 1
        order = APICtrl.GetResultSetValueByName (index,"order")
            //订单号
        orderDate =APICtrl.GetResultSetValueByName (index,"orderDate")
            //订单日期
        curType = APICtrl.GetResultSetValueByName (index, "curType")
            //币种
        amount = APICtrl.GetResultSetValueByName (index, "amount")
            //金额
        refundType=APICtrl.GetResultSetValueByName(index,"refundType")
            //退款类型
        state = APICtrl.GetResultSetValueByName (index, "tranState")
            //退款交易状态
        fee = APICtrl.GetResultSetValueByName (index, "fee")
            //手续费
        merchantComment = APICtrl.GetResultSetValueByName(index,
            "merchantComment") //商户备注
        bankComment=APICtrl.GetResultSetValueByName(index,
            "bankComment") //银行备注
        Response.Write("订单号：" + order)
        Response.Write("<br>")
        Response.Write("订单日期：" + orderDate)
        Response.Write("<br>")
        Response.Write("币种：" + curType)
        Response.Write("<br>")
        Response.Write("退款金额：" + amount)
        Response.Write("<br>")
        Response.Write("退款类型：" + refundType)
        Response.Write("<br>")
    End For
End If
```

```
Response.Write("退款交易状态:" + state)
Response.Write("<br>")
Response.Write("手续费:" + fee)
Response.Write("<br>")
Response.Write("商户备注：" + merchantComment)
Response.Write("<br>")
Response.Write("银行备注：" + bankComment)
Response.Write("<p></p>")
Response.Write("<p></p>")
Next
End If
End If
End If
%>
```

5-4-6. 退款录入

```
<%
operator = Request("operator")
order      = Request("order")
refunddate = Request("date")
amount     = Request("amount")
comment    = Request("comment")
Set APICtrl = Server.CreateObject("B2CClientCOMCtrl.B2CClientCOM")
ret = APICtrl.Initialize("C:\bocomm\ini\B2CMerchant.xml") //初始化
//以上代码执行一次
If ret <> 0 Then
    //初始化失败
    Response.Write("初始化失败,错误信息:")
    Err = APICtrl.GetLastError()
    Response.Write(Err)
Else
    ret = APICtrl.Refund( operator, order, refunddate, amount, comment) //退款录入
    If ret <> 0 Then
        Dim Err
        Err = APICtrl.GetLastError()
        Response.Write("交易错误信息:")
        Response.Write(Err)
    Else
        code = APICtrl.GetRetCode() //得到交易返回码
        msg = APICtrl.GetErrorMessage()
        If msg = "" Then
```

```
msg = "null"
End If

Response.Write("交易返回码：" + code + "<br>")
Response.Write("交易错误信息：" + msg + "<br>")
If code = "0" Then
    Response.Write("<br>-----<br>")
    serial = APICtrl.GetResultValueByName("serial") // 退款流水号
    account = APICtrl.GetResultValueByName("account") // 退款账号

    Response.Write("退款流水号" + serial)
    Response.Write("<br>")
    Response.Write("退款账号" + account)
    Response.Write("<br>")
    Response.Write("<p></p>")
End If
End If
End If
%>
```

5-4-7. 对帐单下载

```
<%
settleDate = Request("settleDate")
Set APICtrl = Server.CreateObject("B2CCClientCOMCtrl.B2CCClientCOM")
ret = APICtrl.Initialize("C:\\bocomm\\ini\\B2CMerchant.xml") // 初始化
// 以上代码执行一次
If ret <> 0 Then
    Response.Write("初始化失败, 错误信息:")
    Err = APICtrl.GetLastError()
    Response.Write(Err)
Else
    ret = APICtrl.DownloadSettlement(settleDate) // 对帐单下载
    If ret <> 0 Then
        Dim Err
        Err = APICtrl.GetLastError()
        Response.Write("交易错误信息:")
        Response.Write(Err)
    Else
        code = APICtrl.GetRetCode() // 得到交易返回码
        msg = APICtrl.GetErrorMessage()
```

```
        If msg = "" Then
            msg = "null"
        End If
        Response.Write("交易返回码：" + code + "<br>")
        Response.Write("交易错误信息：" + msg + "<br>")
        If code = "0" Then
            Response.Write("<br>-----<br>")
            totalSum = APICtrl.GetResultValueByName("totalSum")    //总金额
            totalNumber = APICtrl.GetResultValueByName("totalNumber") //总笔数
            totalFee = APICtrl.GetResultValueByName("totalFee")    //总手续费

            Response.Write("总金额：" + totalSum)
            Response.Write("<br>")
            Response.Write("总笔数：" + totalNumber)
            Response.Write("<br>")
            Response.Write("总手续费：" + totalFee)
            Response.Write("<br>")
            Response.Write("<p></p>")
        End If
    End If
End If
%>
```

5-4-8. 持卡人身份验证

```
<%
    card        = Request("card")
    custName     = Request("custName")
    certType     = Request("certType")
    certNo       = Request("certNo")

    Set APICtrl = Server.CreateObject("B2CClientCOMCtrl.B2CClientCOM")
    ret = APICtrl.Initialize("C:\\bocomm\\ini\\B2CMerchant.xml") //初始化
    //以上代码执行一次
    If ret <> 0 Then
        //初始化失败
        Response.Write("初始化失败,错误信息：")
        Err = APICtrl.GetLastError()
        Response.Write(Err)
    Else
        ret = APICtrl.VerifyCustID( card, custName, certType, certNo) //退款录入
    End If
%>
```

```
If ret<>0 Then
    Dim Err
    Err = APICtrl.GetLastError()
    Response.Write("交易错误信息：")
    Response.Write(Err)
Else
    Dim code
    Dim msg
    code = APICtrl.GetRetCode() //得到交易返回码
    msg = APICtrl.GetErrorMessage()
    If msg = "" Then
        msg = "null"
    End If

    Response.Write("交易返回码：" + code + "<br>")
    Response.Write("交易错误信息：" + msg + "<br>")
End If
End If
%>
```

5-4-9. 当日订单完整查询

```
<%
    Dim begTime      '开始时间[yyyyMMddHHmmss]
    Dim endTime      '结束时间[yyyyMMddHHmmss]
    Dim flag          '订单状态 0 所有 1 已支付 2 已撤销
    Dim begOrder      '起始订单号，可为 NULL
    Dim endOrder      '截至订单号，可为 NULL
    Dim sortField     '排序字段 1，订单号；2，金额；3，定单时间
    Dim sortOrder     '1：降序 2：升序；

    begTime      = Request("begTime")
    endTime      = Request("endTime")
    flag          = Request("flag")
    begOrder      = Request("begOrder")
    endOrder      = Request("endOrder")
    sortField     = Request("sortField")
    sortOrder     = Request("sortOrder")

    If flag = Null Then
        flag = 0
    End If
```

```
If Not IsNumeric(flag) Then
    flag = 0
End If
If Not IsNumeric(sortField) Then
    sortField = 1
End If
If Not IsNumeric(sortOrder) Then
    sortOrder = 1
End If
If sortField>3 Then
    sortField = 1
End If
If sortOrder>2 Then
    sortOrder = 1
End If

Set APICtrl = Server.CreateObject("B2CCClientCOMCtrl.B2CCClientCOM")
ret = APICtrl.Initialize("C:\\bocomm\\ini\\B2CMerchant.xml") //初始化
//以上代码执行一次
If ret <> 0 Then
    Response.Write("初始化失败,错误信息:")
    Err = APICtrl.GetLastError()
    Response.Write(Err)
Else
    ret = APICtrl.QueryCurOrderFull(begTime, endTime, Int(flag), begOrder,
endOrder, Int(sortField), Int(sortOrder))
    If ret<>0 Then
        Dim Err
        Err = APICtrl.GetLastError()
        Response.Write("交易错误信息：")
        Response.Write(Err)
    Else
        Dim code
        Dim msg
        code = APICtrl.GetRetCode() //得到交易返回码
        msg = APICtrl.GetErrorMessage()
        If msg = "" Then
            msg = "null"
        End If
        num = APICtrl.GetResultSetSize() //得到返回记录个数
        Response.Write("交易返回码：" + code + "<br>")
        Response.Write("交易错误信息：" + msg + "<br>")

        If code = "0" Then
```

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

Response.Write("订单号：" + order)
Response.Write("<br>")
Response.Write("订单日期：" + orderDate)
Response.Write("<br>")
Response.Write("订单时间：" + orderTime)
Response.Write("<br>")
Response.Write("币种：" + curType)
```

```
Response.Write("<br>")
Response.Write("金额：" + amount)
Response.Write("<br>")
Response.Write("交易日期：" + tranDate)
Response.Write("<br>")
Response.Write("交易时间：" + tranTime)
Response.Write("<br>")
Response.Write("支付交易状态[1:成功]：" + tranState)
Response.Write("<br>")
Response.Write("定单状态[0 所有 1 已支付 2 已撤销]：" +
orderState)

Response.Write("<br>")
Response.Write("手续费：" + fee)
Response.Write("<br>")
Response.Write("银行流水号：" + bankSerialNo)
Response.Write("<br>")
Response.Write("银行批次号：" + bankBatNo)
Response.Write("<br>")
Response.Write("交易卡类型[0:借记卡 1：准贷记卡 2:贷记卡]：" + cardType)

Response.Write("<br>")
Response.Write("商户批次号：" + merchantBatNo)
Response.Write("<br>")
Response.Write("商户备注：" + merchantComment)
Response.Write("<br>")
Response.Write("银行备注：" + bankComment)
Response.Write("<p></p>")
Next
End If
End If
End If
%>
```