



中国农业银行

AGRICULTURAL BANK OF CHINA

农行网上支付平台

B2B 直接支付商户接口编程指南

Java Edition

V1.3

修订历史纪录

日期	版本	说明	作者
2004/10/25	V1.0	初稿	网上支付平台开发小组
2008/01/12	V1.1	为了避免同 B2C 系统的冲突，修改 B2B 类的命名空间	网上支付平台开发小组
2008/04/16	V1.2	添加“批量查询交易结果”的接口	网上支付平台开发小组
2008/12/25	V1.3	添加“打印电子交易凭证”的接口	网上支付平台开发小组

目录

1. 简介	4
1.1 目的	4
1.2 术语、定义和缩略语	4
1.3 适用读者	5
2. 概览	6
2.1 总体架构图	6
2.2 功能描述	6
2.3 接口开发软件包说明	7
3. 安装步骤	9
3.1 安装前检查	9
3.2 接口开发软件包安装	9
3.3 接口开发软件包配置	9
3.4 配置测试应用	9
4. 配置文件说明	10
5. 直接支付商户交易说明	11
5.1 业务流程简介	11
5.2 交易流程	13
5.2.1 直接支付	13
5.2.2 交易查询	15
5.2.3 下载交易记录	16
5.3 交易使用时机	17
5.4 组成交易信息对象	18
5.5 直接支付请求	19
5.6 交易结果接收	20
5.7 查询交易结果	21
5.8 下载交易记录	23
5.9 批量查询交易结果	23
5.10 打印电子交易凭证	24
附录一、程序范例	26
A、直接支付请求范例	26
B、查询交易结果请求范例	28
C、查询下载交易记录请求范例	29
D、接收交易结果通知请求范例	30
E、批量查询交易记录请求范例	31
F、打印电子交易凭证请求范例	32
附录二、响应码一览表	37
附录三、TRUSTPAY CLIENT API	40
COM.HITRUST.B2B.TRUSTPAY.CLIENT.TrxRESPONSE	40
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B.TrnxINFO	41
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B.TrnxITEMS	42
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B.TrnxITEM	43
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B.TrnxREMARKS	44
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B.TrnxREMARK	45

COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B. FUNDTRANSFERREQUEST	46
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B. TRNXRESULT.....	48
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B. QUERYTRNXREQUEST.....	50
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B. DOWNLOADTRNXREQUEST.....	52
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B. QUERYTRNXBATCHREQUEST	52
COM.HITRUST.B2B.TRUSTPAY.CLIENT.B2B. PRINTTRNXVOUCHERREQUEST	53

1. 简介

1.1 目的

本文档是 **B2B** 商户软件包的使用、编程指南，用于指导负责商户交易平台运行、维护的技术人员通过中国农业银行网上支付平台提供的商户端开发软件包来进行 **B2B** 交易系统的配置、交易发送、交易结果的接收以及其它所需的功能。

本商户软件包提供了对**直接支付商户**交易功能的支持。

1.2 术语、定义和缩略语

■ SSL:

SSL (Secure Socket Layer 即安全套接层) 协议。是 Netscape Communication 公司推出在网络传输层之上提供的一种基于 RSA 和保密密钥的用于浏览器和 Web 服务器之间的安全连接技术。

■ B2B 交易网站:

B2B 的网站目前主要可以分为两大类:

- 核心企业针对自己的上下游厂商开通的 B2B 网站，主要面向自己的分销商、供应商以及内部的各个部门，为它们提高便捷有效的网上交易服务。这类网站的目的是为核心企业的生产和经营服务。
- 另外一类主要是一种中间媒介式的网站，它的核心思想是“撮合”，为网上所有希望进行在线交易的公司和企业提供商贸信息、提供网上洽谈、签订合同、在线支付等各种服务。这类网站不像第一类网站，它们的出发点主要不是为了经营自己的产品。中国农副产品交易市场网站就是这类网站。

■ 直接支付商户

直接支付商户本身作为卖方企业。

商户将付款方的企业客户号、工商登记号和收款方的账户以及转账的金额直接发送的农行网上支付平台，付款方通过 **B2B** 交易管理系统经过复核操作后，直接将资金划到收款方账户中。

■ 交易类型

为了能够支持 **B2B** 直接支付商户所需要的交易功能，商户编程软件包提供了以下功能接口供商户调用。

- 直接支付

商户网上交易平台提交资金直接划拨的请求到支付网关。

支付网关验证信息的有效性后，记录下该请求。

- 查询交易记录

商户平台提交查询交易执行结果的请求到支付网关，支付网关回应交易的最新处理状态，商户根据交易的处理状态进行后续的业务处理。

- 下载交易记录

商户平台提交下载交易记录的请求到支付网关，支付网关根据商户平台指定的日期回应当天所有交易状态为最终状态的交易记录明细，商户根据交易的处理状态进行后续的业务处理。

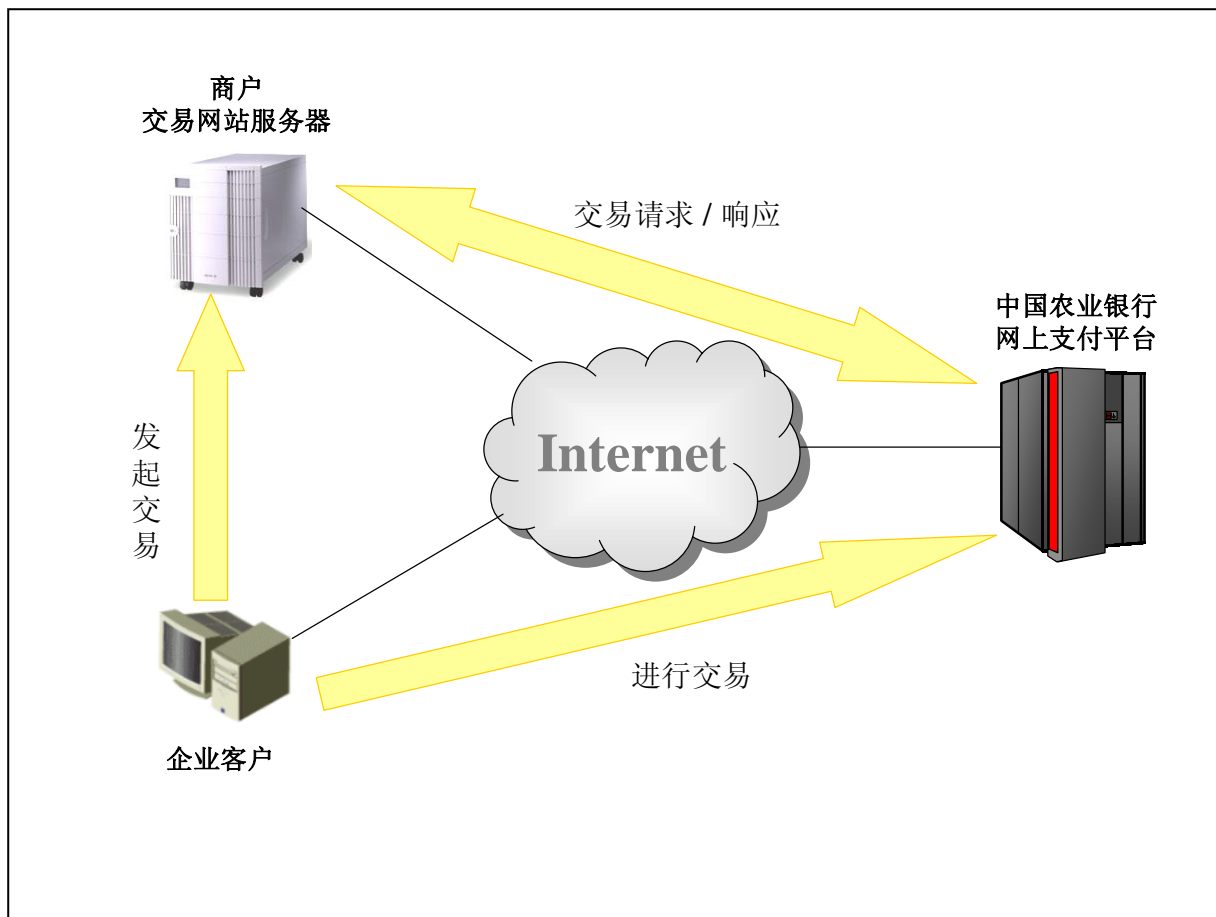
1.3 适用读者

本手册读者对象是商户交易平台的运行、维护、开发的技术人员。我们需要读者具备以下技能，以便通过商户软件包进行商户交易系统和银行支付网关的整合。

- 熟悉通用的 Web 开发技术
- 了解 SSL 和数字证书知识
- 熟悉自身电子商务运营环境和规则
- 熟悉 B2B 交易的网上交易流程

2. 概览

2.1 总体架构图



架构图说明:

- 企业客户在商户交易网站进行交易。
- 商户交易网站将交易请求发送到银行网站支付平台。
- 企业客户登陆网上支付平台进行交易处理。
- 银行网上支付平台将交易处理结果通知商户交易平台。

2.2 功能描述

本软件包为商户端提供了进行交易处理所需的功能，为了方便商户的开发，缩短上线时间，软件包中的商户进行交易所需的功能进行了封装，商户只需进行简单的编程即可实现所需的功能。

接口采用电子证书的方式来保证商户与网上支付平台间的身份验证、中间信息传递的完整性，以便进行电子商务安全当中非常重要的交易身份辨识、不可抵赖、防止篡改等功能。

因为本软件包提供了对直接支付商户的支持，商户在开发时，需要根据自己在农行申请开通



的商户类型进行相应功能的调用。如果错误调用不属于自己业务范围的功能，农行支付平台会拒绝此交易。

在后面的开发指导中，我们会分别对直接支付商户所需的功能进行描述。

■ 直接支付商户的功能接口：

● 直接支付

商户网上交易平台提交资金直接划拨的请求到支付网关。

支付网关验证信息的有效性后，记录下该请求。

● 查询交易记录

商户平台提交查询交易执行结果的请求到支付网关，支付网关回应交易的最新处理状态，商户根据交易的处理状态进行后续的业务处理。

● 下载交易记录

商户平台提交下载交易记录的请求到支付网关，支付网关根据商户平台指定的日期回应当天所有交易状态为最终状态的交易记录明细，商户根据交易的处理状态进行后续的业务处理。

2.3 接口开发软件包说明

银行提供的接口开发软件包 TrustPayClient-B2B-Java-Vx.x.x.zip (x.x.x 为接口开发软件包的版本号) 包含下列文档。

文件名称	说明
/docs/B2B 直接支付商户接口-编程指南-Java-Edition-Vx.x.x.pdf	本文件
/docs/农行网上支付平台-商户使用手册.pdf	商户使用手册
/docs/农行网上支付平台-USB Key 安装及使用手册.pdf	USB Key 安装及使用手册
/lib/TrustPayClient-B2B-Vx.x.jar	农行网上支付平台商户端接口 Java 软件包。
/lib/TrustMerchantB2B.properties	接口配置文件
/demo/Merchant.html	接口范例首页
/demo/Pay01.html	商户（直接支付）范例首页
/demo/MerchantFundTransfer.html	直接支付请求交易范例页面
/demo/MerchantFundTransfer.jsp	直接支付请求交易范例程序
/demo/MerchantQueryTrnx.html	查询交易结果范例页面
/demo/MerchantQueryTrnx.jsp	查询交易结果范例程序
/demo/MerchantDownloadTrnx.html	下载交易记录范例页面
/demo/MerchantDownloadTrnx.jsp	下载交易记录范例程序



/demo/MerchantQueryTrnxBatch.html	批量查询交易结果范例页面
/demo/MerchantQueryTrnxBatch.jsp	批量查询交易结果范例程序
/demo/TrnxResult.jsp	商户交易结果接收程序
/cert/abc.truststore	农行根证书
/cert/trustpay.cer	网上支付平台证书
/ABCIcon/*.jpg	用于商户在自行开发的页面上，如果有指向农行的图片链接，请使用这些图片做为农行标识

3. 安装步骤

3.1 安装前检查

- 1、本接口软件包采用 JDK 1.3 标准。
- 2、请确定服务器已经安装了下列软件：
 - ◆ Sun JSSE V1.0 或更高的版本

3.2 接口开发软件包安装

- 1、将银行提供的接口开发软件包 TrustPayClient-B2B-Java-Vx.x.x.zip (x.x.x 为接口开发软件包的版本号) 解压缩到商户自定的安装目录中。
- 2、请参考《农行网上支付平台 - 商户使用手册 V1.0》登录网上支付平台下载商户交易证书，并将商户交易证书保存到服务器的硬盘或签名服务器中。
- 3、将 TrustPayClient-B2B-Vx.x.jar 加入应用服务器的 CLASSPATH 中。

3.3 接口开发软件包配置

- 1、开启接口配置文件 TrustMerchant.properties，依照银行提供的信息设定相对应的参数。并将 TrustMerchant.properties 所在的目录加入应用服务器的 CLASSPATH 中。详细配置文件的说明请参考下一章的说明。

3.4 配置测试应用

- 1、配置 WebApp，指向接口软件包安装目录demo。
- 2、开启浏览器进入 <http://your.server.name/your.virtual.directory/Merchant.html>，确定接口软件包是否已正确安装及配置。

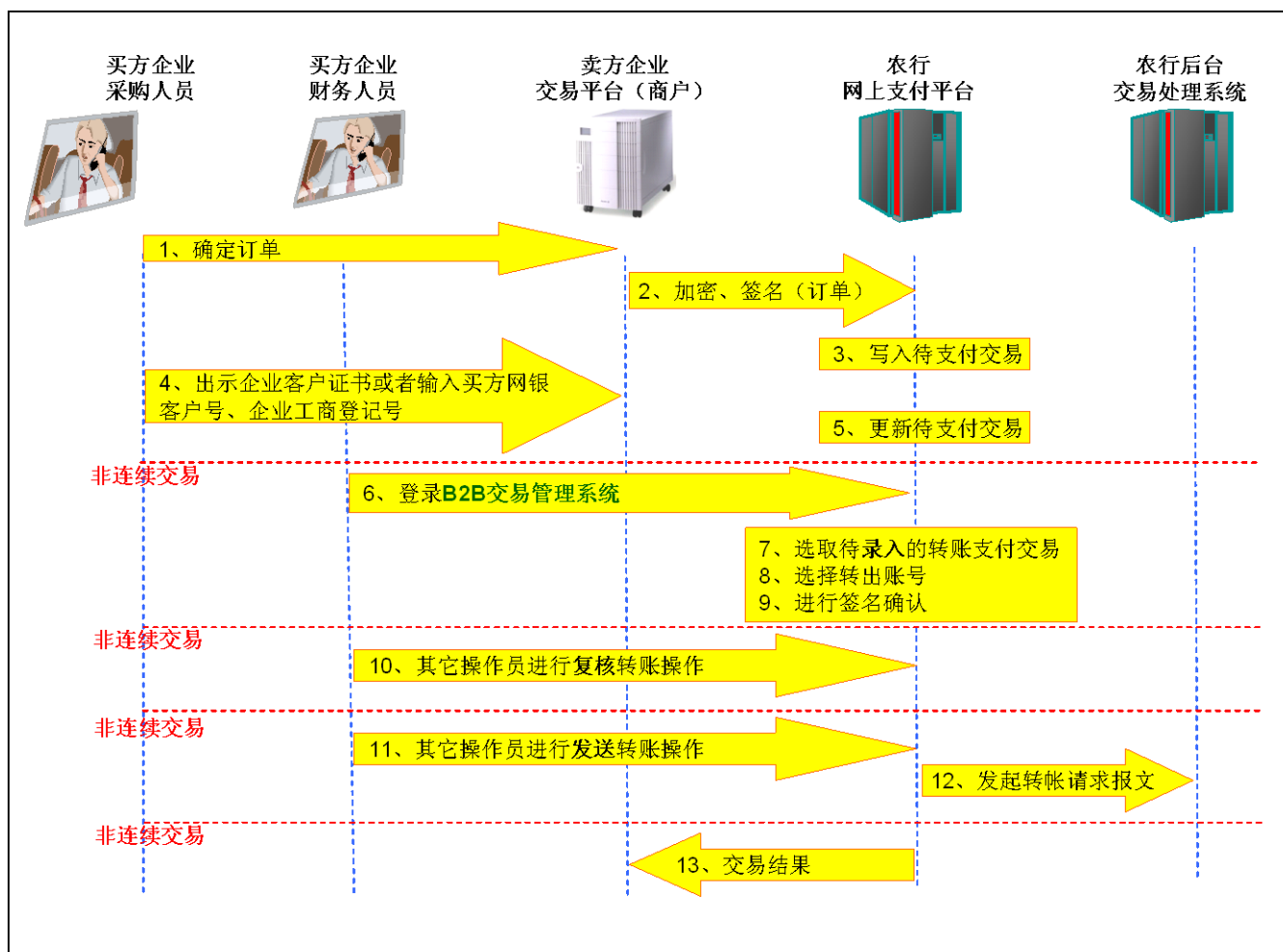
4. 配置文件说明

配置段	参数名称	数值类型	说明
网上支付平台 系统配置段 <i>*请依照银行的 指示设定</i>	TrustPayConnectMethod	字符串	网上支付平台通讯方式 http: 使用 HTTP 通讯方式 https: 使用 HTTPS 通讯方式
	TrustPayServerName	字符串	网上支付平台服务器名 可以使用服务器的域名或服务器的 IP 地址
	TrustPayServerPort	数字	网上支付平台交易端口
	TrustPayNewLine	数字	网上支付平台接口特性 1 或 2
	TrustPayTrxURL	字符串	网上支付平台交易网址
	TrustPayCertFile	字符串	网上支付平台证书
	TrustStoreFile	字符串	农行根证书文件
	TrustStorePassword	字符串	农行根证书文件密码
商户资料段	MerchantID	字符串	商户编号
商户系统配置段	LogPath	字符串	日志文件存放绝对目录。 <i>*请勿输入文件名</i>
	MerchantKeyStoreType	数字	证书储存媒体 0: File 1: 硬件签名服务器
	MerchantCertFile	字符串	商户证书储存目录档名 当 KeyStoreType=0 时，必须设定。 必须为 PKCS#12 的文件格式。
	MerchantCertPassword	字符串	商户私钥加密密码 当 KeyStoreType=0 时，必须设定。
	SignServerIP	字符串	签名服务器 IP 地址 当 KeyStoreType=1 时，必须设定。
	SignServerPort	数字	签名服务器端口 当 KeyStoreType=1 时，必须设定。
	SignServerPassword	字符串	签名服务器密码 当 KeyStoreType=1 时，必须设定。

5. 直接支付商户交易说明

本节对直接支付商户交易处理的流程进行说明。商户在交易的过程中会需要使用到各个不同的类来完成所需要的交易。本章的说明着重在流程的说明，类的详细说明请参考《附录三、TrustPay Client API》。

5.1 业务流程简介



业务流程描述

- 1) 买方企业采购人员登录卖方企业交易平台进行商品选购，并确认订单。
- 2) 商户交易平台将订单信息签名以后，发送到网上支付平台。采购完成。
- 3) 如果交易资料正确，支付网关为买方企业记录下该笔待支付的订单，形成一笔待录入的转账记录。

- 4) 买方企业采购人员出示买方企业客户证书或者输入『买方网银客户号』『买方企业工商登记号』（防止输入错误）。
- 5) 支付网关根据买方企业客户证书或者『买方网银客户号』『买方企业工商登记号』判断该企业是否存在。如果交易资料正确，支付网关此笔转账记录。
- 6) 买方企业财务人员登录网上支付平台 B2B 交易管理系统，选择待录入的直接支付的转账记录，并执行交易录入操作。
- 7) 支付网关显示订单的详细信息，该信息能够使买方财务人员确信该订单是本企业的订单。
- 8) 买方企业财务人员选择支付资金转出的账户。
- 9) 买方企业财务人员签名确定录入此直接支付交易。
- 10) 录入此支付交易后，其它具有复核权限的人员对此转账交易进行复核操作。
- 11) 该交易复核通过后，具有此账户发送权限的企业操作员将此交易发送到银行后台进行实际的转账支付操作，并接收交易结果。
- 12) 支付平台将在交易到达最终状态时主动发起通知，通知商户支付结果接收页面。商户交易平台也可以使用软件包中提供的查询交易随时查询交易的处理结果。并根据处理结果进行后续的业务处理。

交易处理结果的取得方式

- 直接支付交易是非实时交易，企业操作员在对交易处理完成之后（转账成功、驳回该笔转账请求），支付平台会通过服务器直接把交易结果通知给商户所提供的交易结果接收 URL。
- 商户也可以通过人工或者通过商户软件包提供的交易查询功能查询特定交易的交易处理结果。

5.2 交易流程

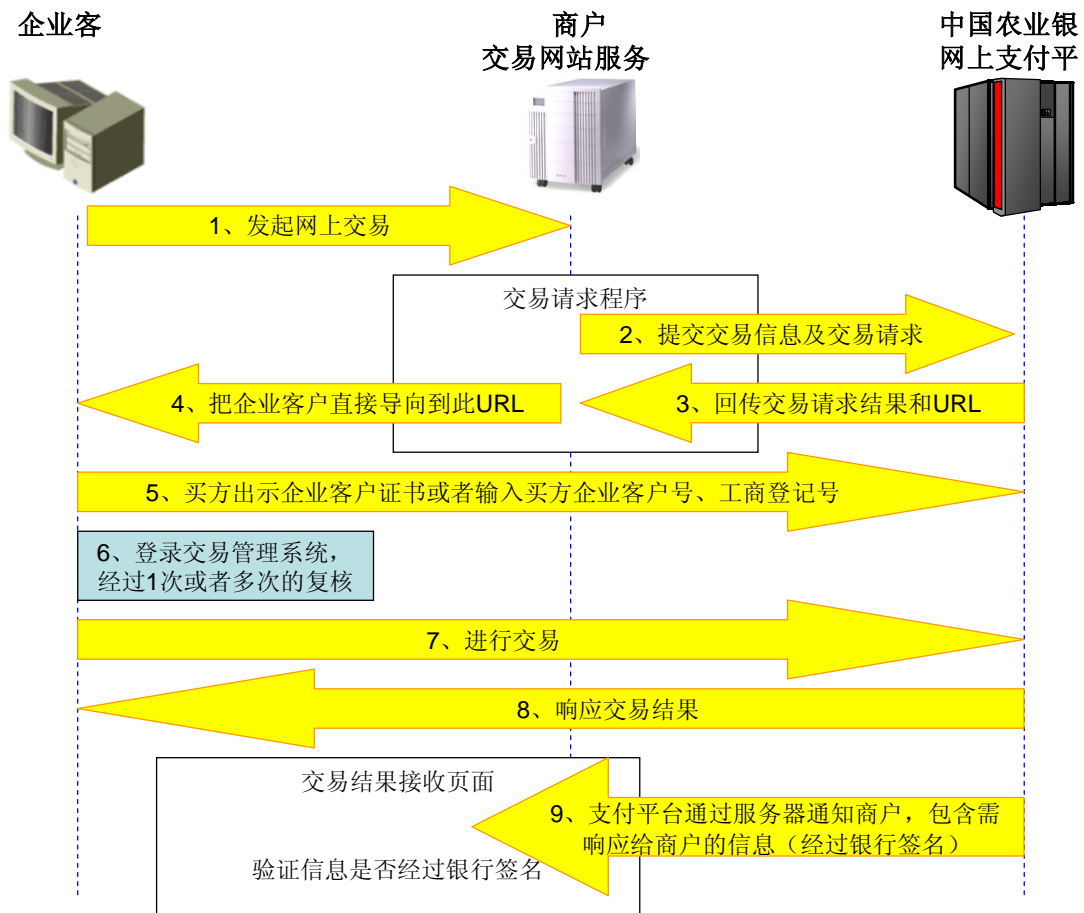
本节将按照交易类型的不同，说明直接支付商户交易平台如何与网上支付平台通信，来完成交易的过程。

5.2.1 直接支付

商户网上交易平台提交资金直接划拨的请求到支付网关，支付网关验证信息的有效性后，记录下该请求，然后等待买方企业客户去复核。当复核通过后，买方企业客户进行实际交易，并将交易结果自动提交给商户平台。

具体步骤如下：

1. 买方企业客户登录商户网站，确认订单
2. 商户提交直接支付请求给网上支付平台
3. 网上支付平台返回给商户一个 URL，商户将这个 URL 页面直接导向给买方企业客户
4. 买方企业客户进入到这个页面，填写买方企业客户代码或者工商登记号，提交给网上支付平台
5. 买方企业客户登录到 B2B 交易管理系统，进行录入、复核、发送交易
6. 交易完成后，商户交易网站可以通过接收网上支付平台发送的支付结果通知或者通过网上支付平台提供的查询交易结果去查询交易是否成功



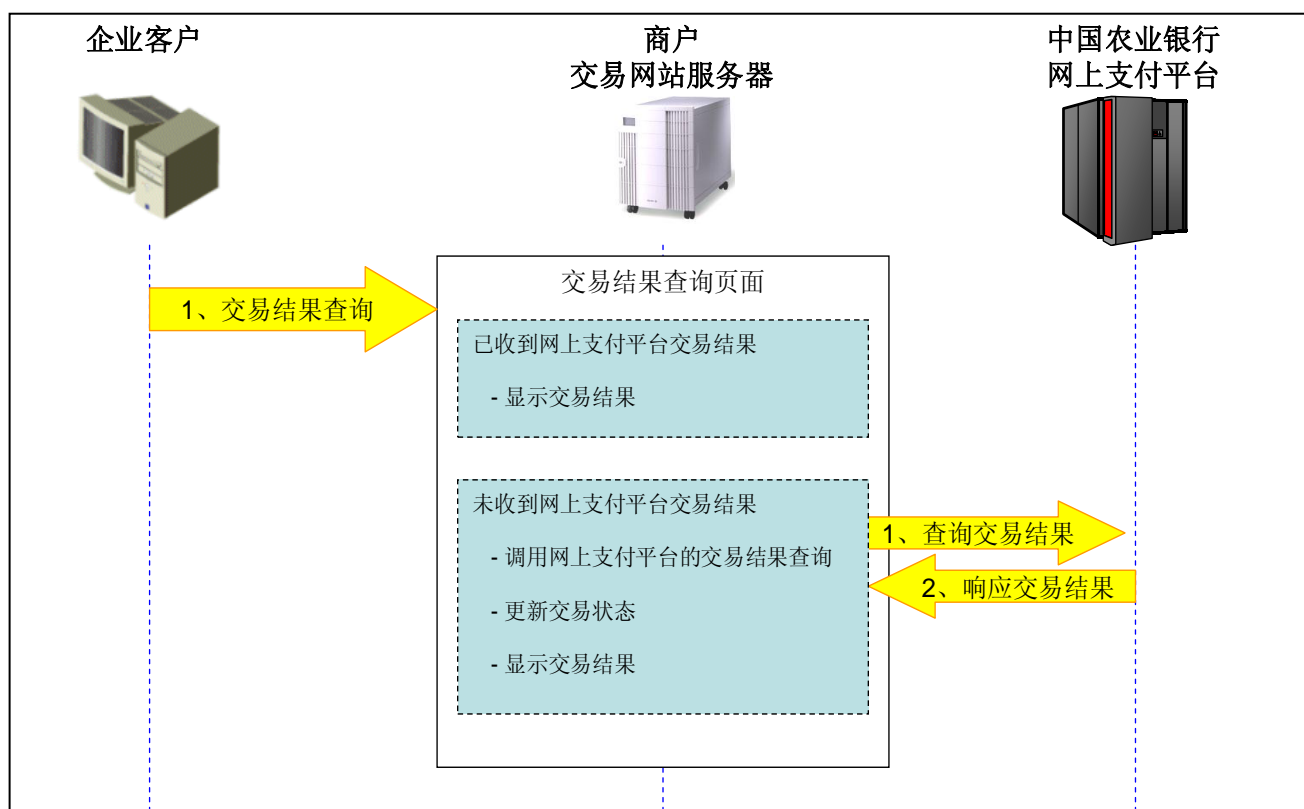
5.2.2 交易查询

网上支付平台为了防止网络异常中断所造成的交易结果丢失，确保交易结果正确送达商户网站，建议商户实现下列网上支付平台所提供的机制。

◆ 查询交易结果

针对未收到银行交易结果回复的交易记录，或银行响应交易状态未明的交易记录，商户可以在任何时刻主动发起交易记录查询请求（详细交易说明请参考 5.7 查询交易结果），查询交易的状态。

例如：在商户网站提供买方企业操作员交易结果查询的功能，如该订单未收到网上支付平台交易结果，则调用网上支付平台的查询交易结果取得交易结果，然后以取得的交易结果更新商户网站的交易状态。



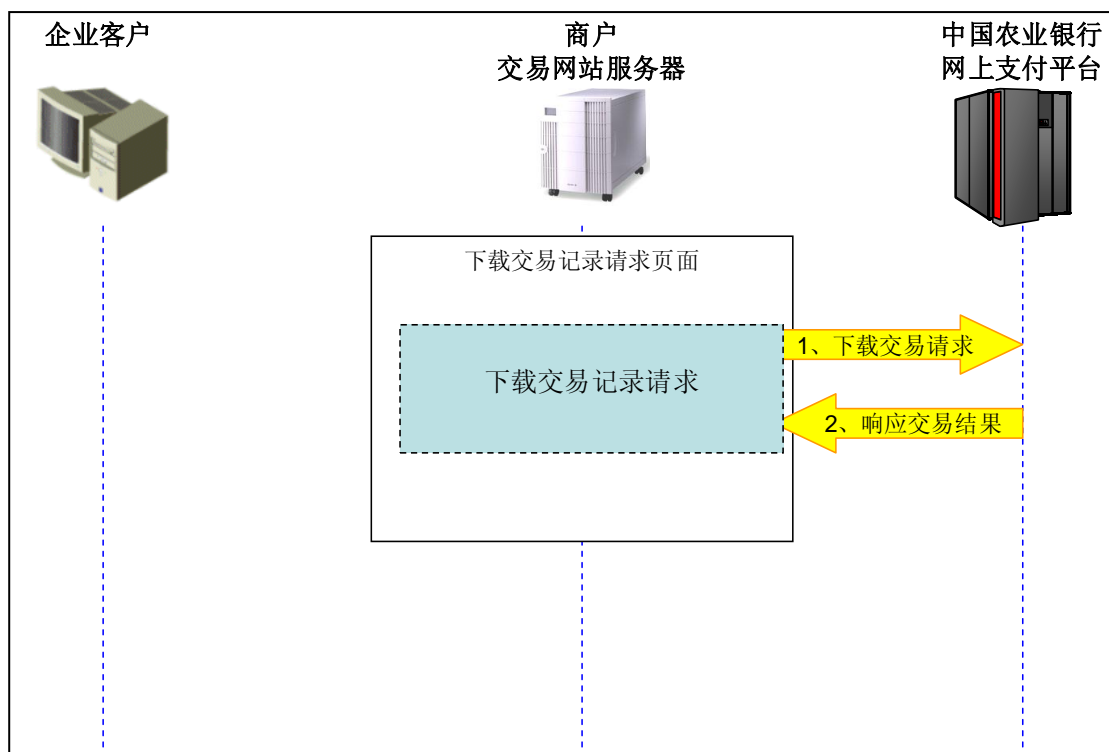
5.2.3 下载交易记录

网上支付平台为了使商户更方便的查询、统计交易记录，提供给商户下载交易记录功能接口，建议商户实现下列网上支付平台所提供的机制。

◆ 下载交易记录

使用下载交易记录功能，商户可以下载指定日期里的当天所有交易状态为最终状态的交易记录。

交易最终状态指的是此笔交易不能再进行后续处理时的状态，包括交易成功状态（2）、交易失败状态（3）、交易作废状态（5）。



5.3 交易使用时机

网上支付平台只提供对下列交易类型的支持，具体的业务逻辑，如何组织这些交易流程由商户根据实际情况自己定制：

■ 直接支付

适用于直接支付的商户交易网站，商户在交易中充当卖方角色，当商户准备把资金从付款方账户直接划拨到收款方账户时使用。

■ 交易结果接收

企业客户在网上支付平台上进行直接转账的录入、复核、发送操作，交易成功或者失败后，支付平台会通过服务器直接把交易结果通知给商户所提供的交易结果接收 URL。商户必须开发此接收程序，否则无法收到交易结果的通知。

■ 查询交易结果

针对未收到银行交易结果回复的交易记录，或银行响应交易状态未明的交易记录，商户可以发起交易查询请求，查询交易的结果。

■ 下载交易记录

希望统计或者查询一天内的所有交易记录状态为最终状态（成功、失败、作废）的交易记录信息。

5.4 组成交易信息对象

商户需要把具体的交易信息根据不同的交易类型组成不同的交易请求发送到网上支付平台，其中是 `TrnxInfo` 是一个都要用到的交易信息对象，所以集中描述如何组成这个对象：

- 1、生成交易信息 `TrnxInfo` 对象 `com.hitrust.b2b.trustpay.client.b2b.TrnxInfo`（必填信息）
- 2、设定 `TrnxInfo` 对象的属性
 - ◆ `TrnxOpr` 交易人员（可选信息）（提供买方操作员在商户的交易网站上输入姓名或员工代码，便于买方在后续交易中知道此交易是由谁发起的）
- 3、生成交易备注信息对象 `com.hitrust.b2b.trustpay.client.b2b.TrnxInfo.TrnxRemarks`，并把 `TrnxRemarks` 对象加入到 `TrnxInfo` 对象中（可选信息，如果不需要显示备注信息，只需要创建一个空的 `TrnxRemarks` 对象）
- 4、生成交易明细对象 `com.hitrust.b2b.trustpay.client.b2b.TrnxInfo.TrnxItems`，并把 `TrnxItems` 对象加入到 `TrnxInfo` 对象中（可选信息，如果不需要显示交易明细信息，只需要创建一个空的 `TrnxItems` 对象）



5.5 直接支付请求

商户交易网站做直接支付请求，需要把资金从付款方账户直接划转到收款方账户，具体步骤为：

商户首先提交直接支付请求给网上支付平台，网上支付平台需要进行复核确认后继续进行交易，交易完成后，商户交易网站可以通过接收网上支付平台发送的支付结果通知或者通过网上支付平台提供的查询交易结果去查询交易是否成功。

1、生成交易信息对象 `com.hitrust.b2b.trustpay.client.b2b.TrnxInfo`（参见 5.4 组成交易信息对象 `TrnxInfo`）

2、生成直接支付请求对象 `com.hitrust.b2b.trustpay.client.b2b.FundTrnasferRequest`

3、设定直接支付请求对象的属性

◆ <code>MerchantTrnxNo</code>	商户交易编号（必要信息）
◆ <code>TrnxDate</code>	交易日期（必要信息）
◆ <code>TrnxTime</code>	交易时间（必要信息）
◆ <code>TrnxAmount</code>	交易金额（必要信息）
◆ <code>AccountDBNo</code>	收款方账号（必要信息）
◆ <code>AccountDBName</code>	收款方账户名
◆ <code>AccountDBBank</code>	收款方账户开户行联行号
◆ <code>TrnxInfo</code>	交易信息对象（必要信息）
◆ <code>ResultNotifyURL</code>	交易结果接收页面网址（必要信息）
◆ <code>MerchantRemarks</code>	商户备注信息

4、使用直接支付请求对象的 `postRequest()` 方法传送直接支付请求并取得交易结果对象。

5、使用交易结果对象的 `isSuccess()` 方法辨别直接支付请求是否成功。

6、若请求成功，等待网上支付平台复核此笔交易，然后进行后续操作。

7、若请求失败，可以使用交易结果对象的 `getReturnCode ()` 及 `getErrorMessage()` 方法取得交易失败代码和原因。

交易结果对象的 `ReturnCode` 所回传的响应码请参考《附录二、响应码一览表》的说明。



5.6 交易结果接收

买方企业操作员在网上支付平台上进行在线交易的操作，交易成功后，网上支付平台会将交易的结果通过服务器通知到商户交易平台指定的交易结果接收的 URL。

注意：

- a) 在开发此交易结果接收页面的时候，为了是网上支付平台服务器确定已经成功的把交易结果消息通知给商户，就要求商户在此页面的代码里加上如下一段（注意 URL 地址要用 `<URL></URL>`括起来）：

`<URL>此交易结果接收页面的 URL 地址</URL>`

- b) 此页面的 URL 应该是在公网能访问的网址，而且端口号必须为 80 或者 443（http 默认端口为 80；https 默认端口为 443）

商户取得银行通知信息的步骤说明如下：

- 1、取得网上支付平台 post 的 MSG 参数
- 2、利用此参数生成交易结果对象 `com.hitrust.b2b.trustpay.client.b2b.TrnxResult`
- 3、使用交易结果对象的 `isSuccess()`方法辨别网上支付平台发送的响应信息是否正确（信息格式、签名是否正确），转账交易状态请确定响应信息中的 `TrnxStatus` 值。
- 4、若交易成功，则商户可以使用交易结果对象的 `getValue()`方法取得下列回传值来进行后续的作业：

◆ MerchantID	商户代码
◆ CorporationCustomerNo	买方企业客户代码
◆ MerchantTrnxNo	商户交易编号
◆ TrnxSN	交易流水号
◆ TrnxType	交易类型
◆ TrnxAMT	交易金额
◆ FreezeNo	冻结序号
◆ OrginalFreezeNo	原冻结序号
◆ AccountNo	付款方账号
◆ AccountName	付款方名称
◆ AccountBank	付款方账户行联行号
◆ AccountDBNo	收款方账号
◆ AccountDBName	收款方名称

- ◆ AccountDBBank 收款方账户行联行号
- ◆ TrnxTime 交易时间
- ◆ TrnxStatus 转账交易状态（2 为转账成功）

5、若交易失败，则商户可以使用交易结果对象的 **getValue()** 方法取得下列回传值来进行后续的作业：

- ◆ MerchantID 商户代码
- ◆ CorporationCustomerNo 买方企业客户代码
- ◆ MerchantTrnxNo 商户交易编号
- ◆ TrnxType 交易类型
- ◆ TrnxAMT 交易金额
- ◆ TrnxTime 交易时间
- ◆ TrnxStatus 交易状态

5.7 查询交易结果

商户可以发起交易查询请求，查询交易的结果。商户发起查询交易的步骤说明如下：

1、生成交易查询请求对象 **com.hitrust.b2b.trustpay.client.b2b. QueryTrnxRequest**

2、设定交易查询请求对象的属性

- ◆ MerchantTrnxNo 商户交易编号号（必要信息）
- ◆ MerchantRemarks 商户备注信息

3、调用交易查询请求对象的 **postRequest()** 方法传送交易查询请求并取得交易结果对象

4、使用交易结果对象的 **isSuccess()** 方法辨别查询交易是否成功执行。

5、若查询交易执行成功，则可以通过交易响应对象(**TrxResponse**)的 **getValue()** 方法取得如下信息。

- ◆ MerchantID 商户代码
- ◆ CorporationCustomerNo 买方企业客户代码
- ◆ MerchantTrnxNo 商户交易编号
- ◆ TrnxSN 交易流水号
- ◆ TrnxType 交易类型
- ◆ TrnxAMT 交易金额
- ◆ AccountNo 付款方账号
- ◆ AccountName 付款方名称
- ◆ AccountBank 付款方账户行联行号
- ◆ AccountDBNo 收款方账号



- ◆ AccountDBName 收款方名称
- ◆ AccountDBBank 收款方账户行联行号
- ◆ TrnxTime 交易时间
- ◆ TrnxStatus 交易状态（2 为成功）

注意：以下是对于交易类型的返回值以及中文说明：

- FUND_TRANSFER 直接支付
- TRNXTYPE_DOWNLOAD_TRNX 下载交易记录
- QUERY_TRNX 查询交易结果

6、若查询交易执行失败，可以使用交易结果对象的 `getReturnCode ()` 及 `getErrorMessage()` 方法取得交易失败代码和原因。交易结果对象的 `ReturnCode` 属性所回传的响应码请参考《附录二、响应码一览表》的说明。



5.8 下载交易记录

商户可以发起下载交易记录请求，查询指定日期里的具有交易最终状态的所有交易记录。商户发起查询交易的步骤说明如下：

- 1、生成下载交易记录请求对象 `com.hitrust.b2b.trustpay.client.b2b.DownloadTrnxRequest`
- 2、设定下载交易记录请求对象的属性
 - ◆ `MerchantTrnxDate` 商户所要查询的交易记录的日期（必要信息）
 - ◆ `MerchantRemarks` 商户备注信息
- 3、调用下载交易记录请求对象的 `postRequest()` 方法传送下载交易记录请求并取得交易记录列表
- 4、使用交易结果对象的 `isSuccess()` 方法辨别查询交易是否成功执行
- 5、若下载交易记录请求执行成功，则可以取得交易明细，每笔交易明细为字符串类型，使用逗号分隔不同的字段。字段信息依次如下：
 - ◆ 交易类型 目前只提供直接支付（FUND_TRANSFER）一种交易类型
 - ◆ 商户交易编号
 - ◆ 交易金额
 - ◆ 交易时间
 - ◆ 交易状态 成功（2）、失败（3）、作废（5）
- 6、若查询交易执行失败，可以使用交易结果对象的 `getReturnCode()` 及 `getErrorMessage()` 方法取得交易失败代码和原因。交易结果对象的 `getReturnCode()` 所回传的响应码请参考《附录二、响应码一览表》的说明。

5.9 批量查询交易结果

商户可以发起交易批量查询请求，批量查询交易的结果。商户发起批量查询交易的步骤说明如下：

- 1、生成交易查询请求对象 `com.hitrust.b2b.trustpay.client.b2b.QueryTrnxBatchRequest`
- 2、设定交易查询请求对象的属性
 - ◆ `TrnxStatus` 交易状态
 - ◆ `StartAccDate` 起始查询日期
 - ◆ `EndAccDate` 终止查询日期
- 3、调用交易查询请求对象的 `postRequest()` 方法传送交易查询请求并取得交易结果对象

- 4、使用交易结果对象的 `isSuccess()` 方法辨别查询交易是否成功执行。
- 5、若查询交易执行成功，则可以通过取出交易响应对象(`TrxResponse`)的内容,再调用 `XMLDocument` 的 `getDocuments(String aTag)` 和 `getValueNotNull(String aTag)` 方法分段取得下列信息。

◆ <code>MerchantTrnxNo</code>	订单号
◆ <code>TrnxTime</code>	交易时间
◆ <code>PayAccountName</code>	付款方账户名
◆ <code>PayAccount</code>	付款方帐号
◆ <code>TrnxAmount</code>	交易金额
◆ <code>TrnxStatus</code>	交易状态(0000 为成功)

- 6、若查询交易执行失败，可以使用交易结果对象的 `getReturnCode ()` 及 `getErrorMessage()` 方法取得交易失败代码和原因。交易结果对象的 `ReturnCode` 属性所回传的响应码请参考《附录二、响应码一览表》的说明。

5.10 打印电子交易凭证

商户可以发起打印电子交易凭证请求，打印电子交易凭证。商户发起打印电子交易凭证的步骤说明如下：

- 1、生成交易查询请求对象 `com.hitrust.b2b.trustpay.client.b2b. PrintTrnxVoucherRequest`
- 2、设定交易查询请求对象的属性

◆ <code>MerchantTrnxNo</code>	订单号
-------------------------------	-----
- 3、调用交易查询请求对象的 `postRequest()` 方法传送交易查询请求并取得交易结果对象
- 4、使用交易结果对象的 `isSuccess()` 方法辨别查询交易是否成功执行。
- 5、若查询交易执行成功，则可以通过取出交易响应对象(`TrxResponse`)的内容,再调用 `XMLDocument` 的 `getDocuments(String aTag)` 和 `getValueNotNull(String aTag)` 方法分段取得下列信息。

◆ <code>VoucherNo</code>	电子交易凭证号码
◆ <code>AccountName</code>	付款人户名
◆ <code>AccountNo</code>	付款人账号
◆ <code>AccountBank</code>	付款人开户银行
◆ <code>AccountDBName</code>	收款人户名
◆ <code>AccountDBNo</code>	收款人账号
◆ <code>AccountDBBank</code>	收款人开户银行



◆ TrnxAMT	交易金额
◆ TrnxSN	交易流水号
◆ MerchantTrnxNo	订单号
◆ TrnxDate	交易日期
◆ TrnxTime	时间戳
◆ PrtTime	打印日期

6、若查询交易执行失败，可以使用交易结果对象的 `getReturnCode ()` 及 `getErrorMessage()` 方法取得交易失败代码和原因。交易结果对象的 `ReturnCode` 属性所回传的响应码请参考《附录二、响应码一览表》的说明。



附录一、程序范例

A、直接支付请求范例

```
<%@ page contentType="text/html; charset=gb2312" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.b2b.*" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.*" %>
<% response.setHeader("Cache-Control", "no-cache"); %>
<HTML>
<HEAD><TITLE>农行网上支付平台-商户接口范例-直接支付请求</TITLE></HEAD>
<BODY BGCOLOR='#FFFFFF' TEXT='#000000' LINK='#0000FF' VLINK='#0000FF' ALINK='#FF0000'>
<CENTER>直接支付请求<br>
<%
//1、取得支付请求所需要的信息
String tMerchantTrnxNo      = request.getParameter("MerchantTrnxNo"      );
String tTrnxAmountStr       = request.getParameter("TrnxAmount"         );
String tTrnxDate            = request.getParameter("TrnxDate"           );
String tTrnxTime            = request.getParameter("TrnxTime"           );
String tAccountDBNo         = request.getParameter("AccountDBNo"        );
String tAccountDBName       = request.getParameter("AccountDBName"      );
String tAccountDBBank       = request.getParameter("AccountDBBank"      );
String tMerchantRemarks    = request.getParameter("MerchantRemarks"   );
String tResultNotifyURL     = request.getParameter("ResultNotifyURL"    );
double  tTrnxAmount         = Double.parseDouble(tTrnxAmountStr);

//2、生成 TrnxInfo 对象
TrnxItems tTrnxItems = new TrnxItems();
tTrnxItems.addTrnxItem(new TrnxItem("0001", "显示器", 1000.00f, 2));
tTrnxItems.addTrnxItem(new TrnxItem("0002", "硬盘", 600.00f, 5));
tTrnxItems.addTrnxItem(new TrnxItem("IP000001", "中国移动 IP 卡", 100.00f, 1));

TrnxRemarks tTrnxRemarks = new TrnxRemarks();
tTrnxRemarks.addTrnxRemark(new TrnxRemark("合同号", "555000000"));
tTrnxRemarks.addTrnxRemark(new TrnxRemark("采购时间", "2003/11/12 14:23:34"));
tTrnxRemarks.addTrnxRemark(new TrnxRemark("交易类型", "买入"));
tTrnxRemarks.addTrnxRemark(new TrnxRemark("其它说明", "能够使买方确信该交易是自己交易的信息"));

TrnxInfo tTrnxInfo = new TrnxInfo();
tTrnxInfo.setTrnxOpr("TrnxOperator0001");
tTrnxInfo.setTrnxRemarks(tTrnxRemarks);
tTrnxInfo.setTrnxItems(tTrnxItems);

//3、生成直接支付请求对象
FundTransferRequest tFundTransferRequest = new FundTransferRequest();
tFundTransferRequest.setTrnxInfo(tTrnxInfo); //设定交易细项 (必
要信息)
tFundTransferRequest.setMerchantTrnxNo(tMerchantTrnxNo); //设定商户交易编号 (必
要信息)
tFundTransferRequest.setTrnxAmount(tTrnxAmount); //设定交易金额 (必
要信息)
tFundTransferRequest.setTrnxDate(tTrnxDate); //设定交易日期 (必
要信息)
tFundTransferRequest.setTrnxTime(tTrnxTime); //设定交易时间 (必
要信息)
tFundTransferRequest.setAccountDBNo(tAccountDBNo); //设定收款方账号 (必
要信息)
tFundTransferRequest.setAccountDBName(tAccountDBName); //设定收款方账户名 (必
要信息)
tFundTransferRequest.setAccountDBBank(tAccountDBBank); //设定收款方账户开户行联行号
(必要信息)
tFundTransferRequest.setResultNotifyURL(tResultNotifyURL); //设定交易结果回传网址
(必要信息)
tFundTransferRequest.setMerchantRemarks(tMerchantRemarks); //设定商户备注信息

//4、传送直接支付请求并取得支付网址
TrxResponse tTrxResponse = tFundTransferRequest.postRequest();
if (tTrxResponse.isSuccess()) {
//5、直接支付请求提交成功
```



```
out.println("TrnxType      = [" + tTrxResponse.getValue("TrnxType"  ) + "]<br>");
out.println("TrnxAMT      = [" + tTrxResponse.getValue("TrnxAMT"  ) + "]<br>");
out.println("MerchantID    = [" + tTrxResponse.getValue("MerchantID" ) + "]<br>");
out.println("MerchantTrnxNo = [" + tTrxResponse.getValue("MerchantTrnxNo") + "]<br>");
}
else {
    //6、直接支付请求提交失败，商户自定后续动作
    out.println("ReturnCode  = [" + tTrxResponse.getReturnCode  () + "]<br>");
    out.println("ErrorMessage = [" + tTrxResponse.getErrorMessage() + "]<br>");
}
%>
<a href='Merchant.html'>回商户首页</a></CENTER>
</BODY></HTML>
```

B、查询交易结果请求范例

```
<%@ page contentType="text/html; charset=gb2312" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.b2b.*" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.*" %>
<% response.setHeader("Cache-Control", "no-cache"); %>
<HTML>
<HEAD><TITLE>农行网上支付平台-商户接口范例-交易查询请求</TITLE></HEAD>
<BODY BGCOLOR='#FFFFFF' TEXT='#000000' LINK='#0000FF' VLINK='#0000FF' ALINK='#FF0000'>
<CENTER>交易查询请求<br>
<%
//1、取得交易查询请求所需要的信息
String tMerchantTrnxNo      = request.getParameter("MerchantTrnxNo"      );
String tMerchantRemarks    = request.getParameter("MerchantRemarks"    );

//2、生成交易查询请求对象
QueryTrnxRequest tQueryTrnxRequest = new QueryTrnxRequest();
tQueryTrnxRequest.setMerchantTrnxNo(tMerchantTrnxNo);           //设定商户交易编号      (必要信
息)
tQueryTrnxRequest.setMerchantRemarks(tMerchantRemarks);      //设定商户备注信息

//3、传送交易查询请求并取得交易网址
TrxResponse tTrxResponse = tQueryTrnxRequest.postRequest();
if (tTrxResponse.isSuccess()) {
//4、交易查询请求提交成功
    out.println("TrnxType      = [" + tTrxResponse.getValue("TrnxType"      ) + "]<br>");
    out.println("TrnxAMT      = [" + tTrxResponse.getValue("TrnxAMT"      ) + "]<br>");
    out.println("MerchantID    = [" + tTrxResponse.getValue("MerchantID"    ) + "]<br>");
    out.println("MerchantTrnxNo = [" + tTrxResponse.getValue("MerchantTrnxNo") + "]<br>");
    out.println("ReturnCode    = [" + tTrxResponse.getValue("ReturnCode")    + "]<br>");
}
else {
//5、交易查询请求提交失败，商户自定后续动作
    out.println("ReturnCode    = [" + tTrxResponse.getReturnCode () + "]<br>");
    out.println("ErrorMessage = [" + tTrxResponse.getErrorMessage() + "]<br>");
}
%>
<a href='Merchant.html'>回商户首页</a></CENTER>
</BODY></HTML>
```



C、查询下载交易记录请求范例

```
<%@ page contentType="text/html; charset=gb2312" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.b2b.*" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.*" %>
<%@ page import = "java.util.ArrayList" %>
<% response.setHeader("Cache-Control", "no-cache"); %>
<HTML>
<HEAD><TITLE>农行网上支付平台-商户接口范例-下载交易记录请求</TITLE></HEAD>
<BODY BGCOLOR='#FFFFFF' TEXT='#000000' LINK='#0000FF' VLINK='#0000FF' ALINK='#FF0000'>
<CENTER>下载交易记录请求<br>
<%
//1、取得下载交易记录请求所需要的信息
String tMerchantTrnxDate = request.getParameter("MerchantTrnxDate");
String tMerchantRemarks = request.getParameter("MerchantRemarks" );

//2、生成下载交易记录请求对象
DownloadTrnxRequest tDownloadTrnxRequest = new DownloadTrnxRequest();
tDownloadTrnxRequest.setMerchantTrnxDate(tMerchantTrnxDate);           //设定商户交易编号      (必
要信息)
tDownloadTrnxRequest.setMerchantRemarks(tMerchantRemarks);         //设定商户备注信息

//3、传送下载交易记录请求并取得支付网址
TrxResponse tTrxResponse = tDownloadTrnxRequest.postRequest();
if (tTrxResponse.isSuccess()) {
//4、下载交易记录请求提交成功
XMLDocument tDetailRecords = new XMLDocument(tTrxResponse.getValue("TrnxDetail"));
ArrayList tRecords = tDetailRecords.getDocuments("TrnxRecord");
String[] iRecord = new String[tRecords.size()];
if(tRecords.size() > 0) {
    for(int i = 0; i < tRecords.size(); i++) {
        iRecord[i] = ((XMLDocument)tRecords.get(i)).toString();
        out.println("Record-" + i + " = [" + iRecord[i] + "]<br>");
    }
}
else {
    out.println("<br>指定的日期里没有交易记录<br><br>");
}
}
else {
//5、下载交易记录请求提交失败，商户自定后续动作
    out.println("ReturnCode = [" + tTrxResponse.getReturnCode () + "]<br>");
    out.println("ErrorMessage = [" + tTrxResponse.getErrorMessage() + "]<br>");
}
%>
<a href='Merchant.html'>回商户首页</a></CENTER>
</BODY></HTML>
```



D、接收交易结果通知请求范例

```
<%@ page contentType="text/html; charset=gb2312" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.b2b.*" %>
<% response.setHeader("Cache-Control", "no-cache"); %>
<HTML>
<HEAD><TITLE>农行网上交易平台-商户接口范例-交易结果接收</TITLE></HEAD>
<BODY BGCOLOR='#FFFFFF' TEXT='#000000' LINK='#0000FF' VLINK='#0000FF' ALINK='#FF0000'>
<CENTER>交易结果<br>
<URL>http://mall.hitrust.com.cn/TrnxResult.jsp</URL>
<%
//1、取得 MSG 参数，并利用此参数值生成交易结果对象
TrnxResult tResult = new TrnxResult(request.getParameter("MSG"));

//2、判断交易结果状态，进行后续操作
if (tResult.isSuccess()) {
//3、交易成功
    out.println("商户代码"                = [" + tResult.getValue("MerchantID"                ) +
"]<br>");
    out.println("企业客户代码"            = [" + tResult.getValue("CorporationCustomerNo" ) +
"]<br>");
    out.println("商户交易编号"            = [" + tResult.getValue("MerchantTrnxNo"          ) +
"]<br>");
    out.println("<font color=red>交易流水号" = [" +
tResult.getValue("TrnxSN"                ) + "]</font><br>");
    out.println("交易类型"                = [" + tResult.getValue("TrnxType"                ) +
"]<br>");
    out.println("交易金额"                = [" + tResult.getValue("TrnxAMT"                ) +
"]<br>");
    out.println("冻结序号"                = [" + tResult.getValue("FreezeNo"                ) +
"]<br>");
    out.println("原冻结序号"              = [" + tResult.getValue("OrginalFreezeNo"          ) +
"]<br>");
    out.println("付款方帐号"              = [" + tResult.getValue("AccountNo"              ) +
"]<br>");
    out.println("付款方帐户名"            = [" + tResult.getValue("AccountName"            ) +
"]<br>");
    out.println("付款方开户行联行号"      = [" + tResult.getValue("AccountBank"            ) +
"]<br>");
    out.println("收款方帐号"              = [" + tResult.getValue("AccountDBNo"            ) +
"]<br>");
    out.println("收款方帐户名"            = [" + tResult.getValue("AccountDBName"          ) +
"]<br>");
    out.println("收款方开户行联行号"      = [" + tResult.getValue("AccountDBBank"          ) +
"]<br>");
    out.println("交易时间"                = [" + tResult.getValue("TrnxTime"                ) +
"]<br>");
    out.println("交易状态"                = [" + tResult.getValue("TrnxStatus"            ) +
"]<br>");
    /*
    out.println("MerchantID"              = [" + tResult.getValue("MerchantID"              ) +
"]<br>");
    out.println("CorporationCustomerNo" = [" + tResult.getValue("CorporationCustomerNo" ) +
"]<br>");
    out.println("MerchantTrnxNo"         = [" + tResult.getValue("MerchantTrnxNo"          ) +
"]<br>");
    out.println("TrnxSN"                 = [" + tResult.getValue("TrnxSN"                 ) +
"]<br>");
    out.println("TrnxType"               = [" + tResult.getValue("TrnxType"               ) +
"]<br>");
    out.println("TrnxAMT"                 = [" + tResult.getValue("TrnxAMT"                 ) +
"]<br>");
    out.println("FreezeNo"                = [" + tResult.getValue("FreezeNo"                ) +
"]<br>");
    out.println("OrginalFreezeNo"         = [" + tResult.getValue("OrginalFreezeNo"          ) +
"]<br>");
    out.println("AccountNo"              = [" + tResult.getValue("AccountNo"              ) +
"]<br>");
    out.println("AccountName"            = [" + tResult.getValue("AccountName"            ) +
"]<br>");
    out.println("AccountBank"            = [" + tResult.getValue("AccountBank"            ) +
"]<br>");
*/
}
```



```
        out.println("AccountDBNo"           = [" + tResult.getValue("AccountDBNo"           ) +  
"]<br>");  
        out.println("AccountDBName"         = [" + tResult.getValue("AccountDBName"         ) +  
"]<br>");  
        out.println("AccountDBBank"         = [" + tResult.getValue("AccountDBBank"         ) +  
"]<br>");  
        out.println("TrnxTime"              = [" + tResult.getValue("TrnxTime"              ) +  
"]<br>");  
        out.println("TrnxStatus"            = [" + tResult.getValue("TrnxStatus"            ) +  
"]<br>");  
        /*  
        }  
        else {  
            //4、交易失败  
            out.println("商户代码"           = [" + tResult.getValue("MerchantID"           ) +  
"]<br>");  
            out.println("企业客户代码"       = [" + tResult.getValue("CorporationCustomerNo" ) +  
"]<br>");  
            out.println("商户交易编号"       = [" + tResult.getValue("MerchantTrnxNo"       ) +  
"]<br>");  
            out.println("交易类型"           = [" + tResult.getValue("TrnxType"           ) + "]<br>");  
            out.println("交易金额"           = [" + tResult.getValue("TrnxAMT"           ) +  
"]<br>");  
            out.println("交易时间"           = [" + tResult.getValue("TrnxTime"           ) +  
"]<br>");  
            out.println("交易状态"           = [" + tResult.getValue("TrnxStatus"         ) +  
"]<br>");  
            /*  
            out.println("MerchantID"          = [" + tResult.getValue("MerchantID"          ) +  
"]<br>");  
            out.println("CorporationCustomerNo = [" + tResult.getValue("CorporationCustomerNo" ) +  
"]<br>");  
            out.println("MerchantTrnxNo"      = [" + tResult.getValue("MerchantTrnxNo"      ) +  
"]<br>");  
            out.println("TrnxType"            = [" + tResult.getValue("TrnxType"            ) +  
"]<br>");  
            out.println("TrnxAMT"             = [" + tResult.getValue("TrnxAMT"             ) +  
"]<br>");  
            out.println("TrnxTime"            = [" + tResult.getValue("TrnxTime"            ) +  
"]<br>");  
            out.println("TrnxStatus"          = [" + tResult.getValue("TrnxStatus"          ) +  
"]<br>");  
            /*  
            //out.println("ReturnCode" = [" + tResult.getReturnCode () + "]<br>");  
            //out.println("ErrorMessage = [" + tResult.getErrorMessage() + "]<br>");  
        }  
        %>  
        <a href='Merchant.html'>回商户首页</a></CENTER>  
    </BODY></HTML>
```

E、批量查询交易记录请求范例

```
<%@ page contentType="text/html; charset=gb2312" %>  
<%@ page import = "com.hitrust.b2b.trustpay.client.b2b.*" %>  
<%@ page import = "java.util.ArrayList" %>  
<%@ page import = "com.hitrust.b2b.trustpay.client.*" %>  
<%@ page import = "java.util.Vector" %>  
<%  
response.setHeader("Cache-Control", "no-cache");  
%>  
<HTML>  
<HEAD><TITLE>农行网上支付平台-商户接口范例-交易记录批量查询请求</TITLE></HEAD>  
<BODY BGCOLOR='#FFFFFF' TEXT='#000000' LINK='#0000FF' VLINK='#0000FF' ALINK='#FF0000'>  
<CENTER>交易记录批量查询请求<br>  
<%  
//1、取得交易批量查询请求所需要的信息  
  
String tTrnxStatus = request.getParameter("TrnxStatus");  
String tStartAccDate = request.getParameter("StDate");  
String tEndAccDate = request.getParameter("EndDate");
```



```

//2、生成交易批量查询请求对象
QueryTrnxBatchRequest tQueryTrnxBatchRequest = new QueryTrnxBatchRequest();

tQueryTrnxBatchRequest.setTrnxStatus(tTrnxStatus);           //交易状态
tQueryTrnxBatchRequest.setStartAccDate(tStartAccDate);       //起始查询日期 YYYYMMDD （必要信
息）
tQueryTrnxBatchRequest.setEndAccDate(tEndAccDate);           //终止查询日期 YYYYMMDD （必要信息）

//3、传送交易批量查询请求并取得支付网址
TrxResponse tTrxResponse = tQueryTrnxBatchRequest.postRequest();
if (tTrxResponse.isSuccess()) {
//4、交易查询请求提交成功
XMLDocument tXMLDocument = tTrxResponse.getResponseMessage();
ArrayList tResults = tXMLDocument.getDocuments("Row");
if (tResults.size()==0)
out.print("没有交易记录<br>");
else{
    out.print("订单号      |      交易时间      |      付款方账户名      |      付款方帐号      |      交易金额
|      交易状态<br>");
    for(int i=0;i<tResults.size();i++){
        String tRows = tResults.get(i).toString();
        XMLDocument tChild = new XMLDocument(tRows);
        out.println( tChild.getValueNotNull("MerchantTrnxNo" )      + "|");
        out.println( tChild.getValueNotNull("TrnxTime" )      + "|");
        out.println( tChild.getValueNotNull("PayAccountName" )      + "|");
        out.println( tChild.getValueNotNull("PayAccount") + "|");
        out.println( tChild.getValueNotNull("TrnxAmount")      + "|");
        out.println( tChild.getValueNotNull("TrnxStatus")      + "<br>");
    }
}
}
else {
//5、交易批量查询请求提交失败，商户自定后续动作
out.println("ReturnCode = [" + tTrxResponse.getReturnCode () + "]"<br>");
out.println("ErrorMessage = [" + tTrxResponse.getErrorMessage () + "]"<br>");
}
}%>
<a href='Merchant.html'>回商户首页</a></CENTER>
</BODY></HTML>
  
```

F、打印电子交易凭证请求范例

```

<%@ page contentType="text/html; charset=gb2312" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.b2b.*" %>
<%@ page import = "com.hitrust.b2b.trustpay.client.*" %>
<% response.setHeader("Cache-Control", "no-cache"); %>
<HTML>
<HEAD><TITLE>农行网上支付平台-商户接口范例-电子凭证打印</TITLE></HEAD>
<script language=javascript>
    function printResult()
    {
        window.print();
        window.close();
    }
    function ChangeToBig(value){
        if (value == null) {
            return "错误金额";
        }

        value = "" + value;

        var intFen,i;
        var strArr,strCheck,nstrCheck,strFen,strDW,strNum,strBig,strNow;

        var v = -1;
        try {
            v = parseFloat(value);
        }
        catch (ex) {
        }
    }
  
```



```
if (v == 0 ) {
    return "零元";
}
if (isNaN(v) || v <= 0 || v > 999999999999.99) {
    return "错误金额";
}
else if (v >= 1 && value.charAt(0) == '0') {
    return "错误金额";
}
else if (value.charAt(0) == '+') {
    return "错误金额";
}

if (value.indexOf("e") >=0 || value.indexOf("E") >=0) {
    return "错误金额";
}

var strArray = value.split(".");

if (strArray[1] && strArray[1].length > 2) {    //小数点后面超过两位提示
    return "错误金额";
}

try
{
    i = 0;
    strBig = "";
    intFen = parseInt(value*100+0.000999999999);    //转换为以分为单位的数值
    strFen = intFen.toString();
    strArr = strFen.split(".");
    strFen = strArr[0];
    intFen = strFen.length;    //获取长度
    strArr = strFen.split("");    //将各个数值分解到数组内
    while(intFen!=0)    //分解并转换
    {
        i = i+1;
        switch(i)    //选择单位
        {
            case 1:strDW = "分";break;
            case 2:strDW = "角";break;
            case 3:strDW = "元";break;
            case 4:strDW = "拾";break;
            case 5:strDW = "佰";break;
            case 6:strDW = "仟";break;
            case 7:strDW = "万";break;
            case 8:strDW = "拾";break;
            case 9:strDW = "佰";break;
            case 10:strDW = "仟";break;
            case 11:strDW = "亿";break;
            case 12:strDW = "拾";break;
            case 13:strDW = "佰";break;
            case 14:strDW = "仟";break;
        }

        switch (strArr[intFen-1])    //选择数字
        {
            case "1":strNum = "壹";break;
            case "2":strNum = "贰";break;
            case "3":strNum = "叁";break;
            case "4":strNum = "肆";break;
            case "5":strNum = "伍";break;
            case "6":strNum = "陆";break;
            case "7":strNum = "柒";break;
            case "8":strNum = "捌";break;
            case "9":strNum = "玖";break;
            case "0":strNum = "零";break;
        }

        //处理特殊情况
    }
}
```



```
strNow = strBig.split("");
//分为零时的情况
if((i==1)&&(strArr[intFen-1]=="0"))
{
    strBig = "整";
}
//角为零时的情况
else if((i==2)&&(strArr[intFen-1]=="0"))
{
    //角分同时为零时的情况
    if(strBig!="整")
        strBig = "零"+strBig;
}
//元为零的情况
else if((i==3)&&(strArr[intFen-1]=="0"))
{
    strBig = "元"+strBig;
}
//拾一仟中一位为零且其前一位（元以上）不为零的情况时补零
else if((i<7)&&(i>3)&&(strArr[intFen-1]=="0")&&(strNow[0]!="零
"))&&(strNow[0]!="元"))
{
    strBig = "零"+strBig;
}
//拾一仟中一位为零且其前一位（元以上）也为零的情况时跨过
else if((i<7)&&(i>3)&&(strArr[intFen-1]=="0")&&(strNow[0]=="零")){}
//拾一仟中一位为零且其前一位是元且为零的情况时跨过
else if((i<7)&&(i>3)&&(strArr[intFen-1]=="0")&&(strNow[0]=="元")){}
//当万为零时必须补上万字
else if((i==7)&&(strArr[intFen-1]=="0"))
{
    strBig = "万"+strBig;
}
//拾万一千万中一位为零且其前一位（万以上）不为零的情况时补零
else if((i<11)&&(i>7)&&(strArr[intFen-1]=="0")&&(strNow[0]!="零
"))&&(strNow[0]!="万"))
{
    strBig = "零"+strBig;
}
//拾万一千万中一位为零且其前一位（万以上）也为零的情况时跨过
else if((i<11)&&(i>7)&&(strArr[intFen-1]=="0")&&(strNow[0]=="万")){}
//拾万一千万中一位为零且其前一位为万位且为零的情况时跨过
else if((i<11)&&(i>7)&&(strArr[intFen-1]=="0")&&(strNow[0]=="零")){}
//万位为零且存在仟位和十万以上时，在万仟间补零
else if((i<11)&&(i>8)&&(strArr[intFen-1]!="0")&&(strNow[0]=="万
"))&&(strNow[2]=="仟"))
{
    strBig = strNum+strDW+"万零"+strBig.substring(1,strBig.length);
}
//单独处理亿位
else if(i==11)
{
    //亿位为零且万全为零存在仟位时，去掉万补为零
    if((strArr[intFen-1]=="0")&&(strNow[0]=="万")&&(strNow[2]=="仟
"))
    {
        strBig = "亿"+"零"+strBig.substring(1,strBig.length);
    }
    //亿位为零且万全为零不存在仟位时，去掉万
    else if((strArr[intFen-1]=="0")&&(strNow[0]=="万
"))&&(strNow[2]!="仟"))
    {
        strBig = "亿"+strBig.substring(1,strBig.length);
    }
    //亿位不为零且万全为零存在仟位时，去掉万补为零
    else if((strNow[0]=="万")&&(strNow[2]=="仟"))
    {
        strBig = strNum+strDW+"零
"+strBig.substring(1,strBig.length);
    }
}
```



```
//亿位不为零且万全为零不存在仟位时, 去掉万
else if ((strNow[0]=="万") && (strNow[2]!="仟"))
{
    strBig =
strNum+strDW+strBig.substring(1,strBig.length);
}
//其他正常情况
else
{
    strBig = strNum+strDW+strBig;
}
}
//拾亿一仟亿中一位为零且其前一位(亿以上)不为零的情况时补零
else if ((i<15) && (i>11) && (strArr[intFen-1]=="0") && (strNow[0]!="零
"&& (strNow[0]!="亿"))
{
    strBig = "零"+strBig;
}
//拾亿一仟亿中一位为零且其前一位(亿以上)也为零的情况时跨过
else if ((i<15) && (i>11) && (strArr[intFen-1]=="0") && (strNow[0]=="亿")) {}
//拾亿一仟亿中一位为零且其前一位为亿位且为零的情况时跨过
else if ((i<15) && (i>11) && (strArr[intFen-1]=="0") && (strNow[0]=="零")) {}
//亿位为零且不存在仟万位和十亿以上时去掉上次写入的零
else if ((i<15) && (i>11) && (strArr[intFen-1]!="0") && (strNow[0]=="零
"&& (strNow[1]=="亿") && (strNow[3]!="仟"))
{
    strBig = strNum+strDW+strBig.substring(1,strBig.length);
}
//亿位为零且存在仟万位和十亿以上时, 在亿仟万间补零
else if ((i<15) && (i>11) && (strArr[intFen-1]!="0") && (strNow[0]=="零
"&& (strNow[1]=="亿") && (strNow[3]=="仟"))
{
    strBig = strNum+strDW+"亿零"+strBig.substring(2,strBig.length);
}
else
{
    strBig = strNum+strDW+strBig;
}

strFen = strFen.substring(0,intFen-1);
intFen = strFen.length;
strArr = strFen.split("");
}
return strBig;
}
catch(err)
{
    return ""; //若失败则返回原值
}
}
</script>
<BODY BGCOLOR='#FFFFFF' TEXT='#000000' LINK='#0000FF' VLINK='#0000FF' ALINK='#FF0000'>
<CENTER>电子凭证打印<br>
<%
//1、取得交易查询请求所需要的信息
String tMerchantTrnxNo = request.getParameter("MerchantTrnxNo" );

//2、生成交易查询请求对象
PrintTrnxVoucherRequest tPrintTrnxVoucherRequest = new PrintTrnxVoucherRequest();
tPrintTrnxVoucherRequest.setMerchantTrnxNo(tMerchantTrnxNo); //设定商户交易编号
(必要信息)
//3、传送交易查询请求并取得支付网址
TrxResponse tTrxResponse = tPrintTrnxVoucherRequest.postRequest();
if (tTrxResponse.isSuccess()) {
//4、交易查询请求提交成功
out.println("电子交易凭证号码 = [" + tTrxResponse.getValue("VoucherNo") + "<br>");
out.println("付款人户名 = [" + tTrxResponse.getValue("AccountName" ) + "<br>");
out.println("付款人账号 = [" + tTrxResponse.getValue("AccountNo" ) + "<br>");
out.println("付款人开户银行 = [" + tTrxResponse.getValue("AccountBank" ) + "<br>");
out.println("收款人户名 = [" + tTrxResponse.getValue("AccountDBName" ) + "<br>");
out.println("收款人账号 = [" + tTrxResponse.getValue("AccountDBNo" ) + "<br>");
```



```
out.println("收款人开户银行    = [" + tTrxResponse.getValue("AccountDBBank" ) + "]<br>");
out.println("金额(小写) = [" + tTrxResponse.getValue("TrnxAMT") + "]<br>");
out.println("金额(大写) = [<script language=javascript>document.write(ChangeToBig(' +
tTrxResponse.getValue("TrnxAMT") + "'))</script>]<br>");
out.println("用途          = []<br>");
out.println("交易流水号 = [" + tTrxResponse.getValue("TrnxSN") + "]<br>");
out.println("订单号      = [" + tTrxResponse.getValue("MerchantTrnxNo") + "]<br>");
out.println("交易日期 = [" + tTrxResponse.getValue("TrnxDate") + "]<br>");
out.println("时间戳      = [" + tTrxResponse.getValue("TrnxTime") + "]<br>");
out.println("备注 = []<br>");
out.println("打印日期      = [" + tTrxResponse.getValue("PrtTime") + "]<br>");
}
else {
//5、交易查询请求提交失败，商户自定义后续动作
out.println("ReturnCode    = [" + tTrxResponse.getReturnCode () + "]<br>");
out.println("ErrorMessage = [" + tTrxResponse.getErrorMessage() + "]<br>");
}
%>
<a class=top href="javascript:printResult()">打印</a>&nbsp;
<a href='Merchant.html'>回商户首页</a></CENTER>
</BODY></HTML>
```

附录二、响应码一览表

响应码	响应类型	说明
0000	网上支付平台返回	交易成功
1000	本地返回	无法读取商户端配置文件
1001	本地返回	商户端配置文件中参数设置错误
1002	本地返回	无法读取商户证书文档
1003	本地返回	无法读取商户私钥
1004	本地返回	无法写入交易日志文档
1005	本地返回	证书过期
1006	本地返回	证书格式错误
1100	本地返回	商户提交的交易资料不完整
1101	本地返回	商户提交的交易资料不合法
1102	本地返回	签名交易报文时发生错误
1103	本地返回	无法连线签名服务器
1104	本地返回	签名服务器返回签名错误
1201	本地返回	无法连线网上支付平台
1202	本地返回	提交交易时发生网络错误
1203	本地返回	无法接收到网上支付平台的响应
1204	本地返回	接收网上支付平台响应报文时发生网络错误
1205	本地返回	无法辨识网上支付平台的响应报文
1206	本地返回	网上支付平台服务暂时停止
1301	本地返回	网上支付平台的响应报文不完整
1302	本地返回	网上支付平台的响应报文签名验证失败
1303	本地返回	无法辨识网上支付平台的交易结果
1999	本地返回	系统发生无法预期的错误
2000	网上支付平台返回	无法读取网上支付平台系统配置文件
2001	网上支付平台返回	网上支付平台系统配置文件中参数设置错误
2002	网上支付平台返回	无法读取网上支付平台证书
2003	网上支付平台返回	无法读取网上支付平台私钥
2004	网上支付平台返回	数据库错误
2006	网上支付平台返回	证书格式错误
2100	网上支付平台返回	商户提交的交易资料不完整
2101	网上支付平台返回	商户提交的交易资料不合法
2102	网上支付平台返回	签名响应报文时发生错误
2201	网上支付平台返回	无法连线银行后台系统

2202	网上支付平台返回	接收商户交易请求时发生网络错误
2205	网上支付平台返回	无法辨识商户提交的交易请求报文
2301	网上支付平台返回	商户提交的交易请求报文不完整
2302	网上支付平台返回	商户提交的交易请求报文签名验证失败
2303	网上支付平台返回	商户提交的商户号与签名所用的证书不匹配
2304	网上支付平台返回	商户状态不允许交易
2305	网上支付平台返回	商户不存在
2308	网上支付平台返回	商户无可用的支付方式
2309	网上支付平台返回	无法取得商户证书
2310	网上支付平台返回	网上支付平台未开放此种类的交易
2311	网上支付平台返回	商户未开放指定的商品种类
2312	网上支付平台返回	商户交易编号重复
2400	网上支付平台返回	后台系统响应交易失败
2500	网上支付平台返回	所有交易已测试通过，请通知银行开放可以进行正式交易
2501	网上支付平台返回	此种交易种类测试已通过，请按照网上支付平台所指示的进行其他交易测试
2600	网上支付平台返回	指定的交易不存在
2601	网上支付平台返回	原成功的冻结交易不支持此交易,交易金额超过限制
2602	网上支付平台返回	指定的交易记录不存在
2700	网上支付平台返回	网银企业客户状态非正常，无法进行测试
2701	网上支付平台返回	网银企业客户不存在
2702	网上支付平台返回	指定的网银企业客户与指定的企业工商客户号不匹配
2703	网上支付平台返回	交易金额不能超过企业客户该类业务的单笔交易额度限制
2704	网上支付平台返回	网银客户付款方帐户不存在或者状态非正常
2705	网上支付平台返回	网银客户收款方帐户不存在或者状态非正常
2706	网上支付平台返回	商户手续费帐户不存在或者状态非正常
2707	网上支付平台返回	资金转出或冻结账户所属分行未开通实时跨行交易，未能执行请求的交易
2708	网上支付平台返回	资金转入账户所属分行未开通实时跨行交易，未能执行请求的交易
2710	网上支付平台返回	网银发生无法预期的错误
2711	网上支付平台返回	买方企业帐户所属开户行未开通实时跨行转帐功能
2712	网上支付平台返回	商户手续费结算帐户所属开户行未开通实时跨行转帐功能
2713	网上支付平台返回	卖方企业帐户所属开户行未开通实时跨行转帐功能
2714	网上支付平台返回	商户手续费结算帐户所属开户行未开通实时跨行转帐功能
2715	网上支付平台返回	企业客户没有可用于支付的帐户



2716	网上支付平台返回	买方企业帐户所属开户行未开通实时跨行转帐功能
2717	网上支付平台返回	账户没有权限进行对外支付
2999	网上支付平台返回	系统发生无法预期的错误

附录三、TrustPay Client API

本章中所有对象的属性皆可用 `setAttribute()` 来设定相对应的属性，并用 `getAttribute()` 来取得属性的值。如要取得 `TrxResponse` 对象的 `ReturnCode` 属性，可以使用 `getReturnCode()` 来取得；如要设定 `TrxResponse` 对象的 `ReturnCode` 属性，可以使用 `setReturnCode()`。

com.hitrust.b2b.trustpay.client.TrxResponse

■ 说明

商户型接口软件包实体类，代表网上支付平台对商户提交交易的响应。

■ 属性

属性	属性名称	属性值	类型	说明
RC_SUCCESS	交易状态代码	0000	String	交易请求成功
STATUS_ORIGINAL	交易状态代码	0	String	原始请求
STATUS_CHECK	交易状态代码	1	String	复核中
STATUS_SUCCESS	交易状态代码	2	String	交易成功
STATUS_FAILURE	交易状态代码	3	String	交易失败
STATUS_REJECT	交易状态代码	4	String	交易被驳回
STATUS_CANCEL	交易状态代码	5	String	交易作废
STATUS_NO_RESPONSE	交易状态代码	9	String	交易无响应
ReturnCode	响应码		String	响应码请参考《附录二、响应码一览表》的说明
ErrorMessage	错误信息		String	交易错误原因的本文说明

■ 方法

◆ `public TrxResponse(XMLDocument aXMLDocument)`

`TrxResponse` 的构造函数，使用 `XMLDocument` 初始对象的属性。

参数：`aXMLDocument` – 网上支付平台响应的 XML 报文。

◆ `public bool isSuccess()`

回传交易是否成功。`true` 表示交易成功；`false` 表示交易失败。

◆ `public String getValue(String aTag)`

回传指定参数名的参数值。

com.hitrust.b2b.trustpay.client.b2b.TrnxInfo

■ 说明

商户端接口软件包实体类，代表企业客户在商户网站购买的交易详细信息。

■ 属性

属性	属性名称	类型	说明
TrnxOpr	交易人员	String 最大长度 20 个字符	交易时必须设定
TrnxItems	交易商品明细	为 TrnxItem 对象的 ArrayList 集合	交易时可设定
TrnxRemarks	交易信息明细	为 TrnxRemark 对象的 ArrayList 集合	交易时可设定

■ 方法

◆ public TrnxInfo ()

TrnxInfo 的构造函数，生成一个空的交易信息对象。

◆ public TrnxInfo(XMLDocument aXMLDocument)

构造 TrnxInfo 对象，并使用 XML 文件初始对象的属性。

参数: aXMLDocument – 网上支付平台响应的交易请求结果



com.hitrust.b2b.trustpay.client.b2b.TrnxItems

■ 说明

商户端接口软件包实体类，代表企业客户在商户网站购买的交易商品。

■ 属性

属性	属性名称	类型	说明
无			

■ 方法

- ◆ public TrnxItems ()
- ◆ TrnxItems 的构造函数，生成一个空的交易信息对象。
- ◆ public TrnxItems (XMLDocument aXMLDocument)

构造 TrnxItems 对象，并使用 XML 文件初始对象的属性。

参数：aXMLDocument – 网上支付平台响应的交易请求结果



`com.hitrust.b2b.trustpay.client.b2b.TrnxItem`

■ 说明

商户端接口软件包实体类，代表企业客户在商户网站购买的商品明细。

■ 属性

属性	属性名称	类型	说明
ProductID	产品代码	String 最大长度 20 个字符	交易时可设定
ProductName	产品名称	String 最大长度 50 个字符	交易时可设定
UnitPrice	产品单价	Double 单位为人民币元 小数点后最多两位	交易时可设定
Qty	购买数量	int	交易时可设定

■ 方法

◆ `public TrnxItem()`

`TrnxItem` 的构造函数，生成一个空的订单明细对象。

◆ `public TrnxItem(String aProductID, String aProductName, double aUnitPrice, int aQty)`

构造 `TrnxItem` 对象，并使用参数初始对象的属性。

参数: `aProductID` - 产品代码

`aProductName` - 产品名称

`aUnitPrice` - 产品单价

`aQty` - 购买数量

com.hitrust.b2b.trustpay.client.b2b.TrnxRemarks

■ 说明

商户端接口软件包实体类，代表企业操作员在商户网站的交易信息。

■ 属性

属性	属性名称	类型	说明
无			

■ 方法

- ◆ public TrnxRemarks ()
- ◆ TrnxRemarks 的构造函数，生成一个空的交易信息对象。
- ◆ public TrnxRemarks (XMLDocument aXMLDocument)

构造 TrnxRemarks 对象，并使用 XML 文件初始对象的属性。

参数：aXMLDocument – 网上支付平台响应的交易请求结果



com.hitrust.b2b.trustpay.client.b2b.TrnxRemark

■ 说明

商户端接口软件包实体类，代表企业操作员在商户网站的交易信息明细。

■ 属性

属性	属性名称	类型	说明
Key	交易信息名称	String 最大长度 100 个字符	交易时可设定
Value	交易信息值	String 最大长度 100 个字符	交易时可设定

■ 方法

◆ public TrnxRemark()

TrnxRemark 的构造函数，生成一个空的交易信息明细对象。

◆ public TrnxRemark (String aKey, String aValue)

构造 TrnxRemark 对象，并使用参数初始对象的属性。

参数: aKey - 交易信息名称

a Value -交易信息值



com.hitrust.b2b.trustpay.client.b2b.FundTransferRequest

■ 说明

商户端接口软件包业务处理类，负责商户提交直接支付请求的处理。

■ 属性

属性	属性名称	类型	说明
TrnxInfo	交易信息	TrnxInfo 对象	必须设定
MerchantTrnxNo	商 户 交 易 编 号	String 最大长度 100 个字符	必须设定
TrnxDate	交易日期	String	必须设定
TrnxAmount	交易金额	Double	必须设定
TrnxTime	交易时间	String	必须设定
AccountDBNo	收 款 方 账 号	String 最大长度 20 个字符	必须设定
AccountDBName	收 款 方 账 户 名	String 最大长度 50 个字符	必须设定
AccountDBBank	收 款 方 账 户 开 户 联 行 号	String 最大长度 50 个字符	必须设定
MerchantRemarks	商户 备注信息	String 最大长度 200 个字符	网上支付平台的交易结果 信息将包含此备注信息。 商户可以用来传递所需的 信息。

■ 方法

◆ public TrxResponse postRequest()

发送交易请求至网上支付平台，并接收网上支付平台的响应。

回传值：交易响应对象。如果交易响应成功，则可以使用交易响应对象的
getValue(String aTag)方法取得下列信息。

参数	参数名称	说明
----	------	----



TrnxType	交易种类	FUND_TRANSFER	直接支付
TrnxAMT	交易金额		

com.hitrust.b2b.trustpay.client.b2b.TrnxResult

■ 说明

商户端接口软件包业务处理类，继承 com.hitrust.trustpay.client.TrxResponse，表示网上支付平台响应的交易结果。

■ 属性

■ 方法

◆ public TrnxResult(String aMessage)

构造 TrnxResult 对象，并使用参数初始交易结果对象的属性。

参数：aMessage – 网上支付平台发送到商户交易结果接收页面的 MSG 参数。商户交易结果接收页面可以使用 request.getParameter("MSG") 来取得。

返回值：交易结果对象。如果交易响应成功，则可以使用交易结果对象的 getValue(String aTag) 方法取得下列信息。

参数	参数名称	返回值长度限制	说明
MerchantID	商户代码	最大长度 20 个字符	
CorporationCustomerNo	企业客户代码	最大长度 100 个字符	
MerchantTrnxNo	商户交易编号	最大长度 100 个字符	
OrginalTrnxSN	交易流水号	最大长度 20 个字符	
TrnxType	交易类型	最大长度 20 个字符	
TrnxAMT	交易金额	无	
OrginalFreezeNo	冻结序号	最大长度 6 个字符	直接支付类型交易无此属性
AccountNo	付款方账号	最大长度 20 个字符	
AccountName	付款方名称	最大长度 50 个字符	
AccountBank	付款方账户行联行号	最大长度 50 个字符	
AccountDBNo	收款方账号	最大长度 20 个字符	
AccountDBName	收款方名称	最大长度 50 个字符	
AccountDBBank	收款方账户行联行号	最大长度 50 个字符	
TrnxTime	交易时间	无	



TrnxStatus

交易状态（2 为成功）

最大长度 10 个字符



com.hitrust.b2b.trustpay.client.b2b.QueryTrnxRequest

说明

商户端接口软件包业务处理类，负责商户提交交易查询请求的处理。

属性

属性	属性名称	类型	说明
MerchantTrnxNo	商户交易编号	String 最大长度 100 个字符	必须设定
MerchantRemarks	商户备注信息	String 最大长度 200 个字符	网上支付平台的交易结果信息将包含此备注信息。商户可以用来传递所需的信息。

方法

◆ public TrxResponse postRequest()

发送交易请求至网上支付平台，并接收网上支付平台的响应。

返回值：交易响应对象。如果交易响应成功，则可以使用交易响应对象的
getValue(String aTag)方法取得下列信息。

参数	参数名称	返回值长度限制	说明
MerchantID	商户代码	最大长度 20 个字符	
CorporationCustomerNo	付款方企业客户代码	最大长度 100 个字符	
MerchantTrnxNo	商户交易编号	最大长度 100 个字符	
OrginalTrnxSN	交易流水号	最大长度 20 个字符	
TrnxType	交易类型	最大长度 20 个字符	
TrnxAMT	交易金额	无	
OrginalFreezeNo	冻结序号	最大长度 6 个字符	
AccountNo	付款方账号	最大长度 20 个字符	
AccountName	付款方名称	最大长度 50 个字符	
AccountBank	付款方账户行联行号	最大长度 50 个字符	
AccountDBNo	收款方账号	最大长度 20 个字符	
AccountDBName	收款方名称	最大长度 50 个字符	
AccountDBBank	收款方账户行联行号	最大长度 50 个字符	



TrnxTime	交易时间	无	
TrnxStatus	交易状态（2 为成功）	最大长度 10 个字符	

com.hitrust.b2b.trustpay.client.b2b. DownloadTrnxRequest

■ 说明

商户端接口软件包业务处理类，负责商户提交下载交易记录请求的处理。

■ 属性

属性	属性名称	类型	说明
MerchantTrnxDate	商户交易日期	String 格式如 (YYYY/MM/DD)	必须设定
MerchantRemarks	商户备注信息	String 最大长度 200 个字符	网上支付平台的交易结果信息将包含此备注信息。商户可以用来传递所需的信息。

■ 方法

◆ public TrxResponse postRequest()

发送交易请求至网上支付平台，并接收网上支付平台的响应。

回传值：交易响应对象。如果交易响应成功，则可以使用 **getValue()** 方法取得交易记录明细，每笔交易明细为字符串类型，使用逗号分隔不同的字段。字段信息依次如下：

- ◆ 交易类型 目前只提供直接支付 (FUND_TRANSFER) 一种交易类型
- ◆ 商户交易编号
- ◆ 交易金额
- ◆ 交易时间
- ◆ 交易状态 成功 (2)、失败 (3)、作废 (5)

com.hitrust.b2b.trustpay.client.b2b. QueryTrnxBatchRequest

■ 说明

商户端接口软件包业务处理类，负责商户提交交易批量查询请求的处理。

■ 属性

属性	属性名称	类型	说明
TrnxStatus	交易状态	String	选择设定

			不填为全部查询,0000 为成功状态,Other 为失败状态
StartAccDate	起始查询日期	String 格式为"MM/DD/YYYY"	必须设定 起始日期不大于终止日期
EndAccDate	终止查询日期	String 格式为"MM/DD/YYYY"	必须设定 终止日期不大于当前日期

■ 方法

◆ `public TrxResponse postRequest()`

发送交易请求至网上支付平台,并接收网上支付平台的响应。

返回值: 交易响应对象。如果交易响应成功,则可以使用交易响应对象的 `getResponseMessage()` 方法取得响应信息,再通过 `XMLDocument` 的 `getDocuments(String aTag)` 和 `getValueNotNull(String aTag)` 方法取得下列信息。

参数	参数名称	返回值长度限制	说明
MerchantTrnxNo	订单号	最大长度 100 个字符	
TrnxTime	交易时间	无	
PayAccountName	付款方账户名	最大长度 50 个字符	
PayAccount	付款方帐号	最大长度 20 个字符	
TrnxAmount	交易金额	无	
TrnxStatus	交易状态(0000 为成功)	最大长度 10 个字符	

`com.hitrust.b2b.trustpay.client.b2b.PrintTrnxVoucherRequest`

■ 说明

商户型接口软件包业务处理类,负责商户提交打印电子交易凭证请求的处理。

■ 属性

属性	属性名称	类型	说明
----	------	----	----

MerchantTrnxNo	订单号	String	必须设定
----------------	-----	--------	------

■ 方法

◆ `public TrxResponse postRequest()`

发送交易请求至网上支付平台，并接收网上支付平台的响应。

返回值：交易响应对象。如果交易响应成功，则可以使用交易响应对象的 `getResponseMessage()` 方法取得响应信息，再通过 `XMLDocument` 的 `getDocuments(String aTag)` 和 `getValueNotNull(String aTag)` 方法取得下列信息。

参数	参数名称	返回值长度限制	说明
VoucherNo	电子交易凭证号码	最大长度 100 个字符	
AccountName	付款人户名	最大长度 50 个字符	
AccountNo	付款人账号	最大长度 20 个字符	
AccountBank	付款人开户银行	最大长度 50 个字符	
AccountDBName	收款人户名	最大长度 50 个字符	
AccountDBNo	收款人账号	最大长度 20 个字符	
AccountDBBank	收款人开户银行	最大长度 50 个字符	
TrnxAMT	交易金额	无	
TrnxSN	交易流水号	最大长度 20 个字符	
MerchantTrnxNo	订单号	最大长度 100 个字符	
TrnxDate	交易日期	无	
TrnxTime	时间戳	无	
PrtTime	打印时间	无	