**网关交易文档**

**Version: 1.0.0**

目录

[1概述 2](#_Toc510168339)

[1.1 阅读对象 2](#_Toc510168340)

[2.开发规范及描述 2](#_Toc510168341)

[2.1 通讯协议 2](#_Toc510168342)

[2.2 报文返回 2](#_Toc510168343)

[2.3 签名机制 3](#_Toc510168344)

[2.4 报文规范说明 5](#_Toc510168345)

[2.4.1 符号说明 5](#_Toc510168346)

[2.4.2 接口数据类型定义 6](#_Toc510168347)

[4.接口说明 6](#_Toc510168348)

[4.1网银直连接口 6](#_Toc510168349)

[附录一 返回码描述 8](#_Toc510168350)

[附录二 10](#_Toc510168351)

# 1概述

## 1.1 阅读对象

本文档面向具备一定WEB网页开发能力，对加密算法有一定的了解，最好具备网银支付开发能力的人才

# 2.开发规范及描述

地址：http://103.97.229.106/trans-api/trans/api/back.json

## 2.1 通讯协议

所有报文均以 HTTPS/HTTP 通讯协议 POST 形式提交，字符集需采用**UTF8格式编码** 连接类型使用 application/x-www-form-urlencoded 后台通讯模式通讯

## 2.2 报文返回

支付接口允许用户自定义返回结果，定义方式

通过通讯地址 http://103.97.229.106/trans-api/trans/api/back.json 添加后缀 如：

http://103.97.229.106/trans-api/trans/api/back.json

| 格式类型 | 格式示例 |
| --- | --- |
| json | {"resultCode":"0000","resultMsg":"交易成功"} |
| xml | <root>  <resultCode>0000</resultCode>  <resultMsg>交易成功</resultMsg>  </root> |
| html（默认） | resultCode=0000&resultMsg=交易成功 |

## 2.3 签名机制

为保证数据在传输过程中，数据的真实性，保证数据在传输过程中不被结果或篡改，接口需要对接收的数据进行签名及验证签名校验，我方对数据签名使用算法为：

SHA1WithRSA 算法 提交给我方的数据需要通过私钥文件对交易报文数据进行签名。我方返回的数据需要商户使用 公钥文件对报文验证签名。

**签名/验证签名报文:**

请将报文中所有非空参数(signature)参数除外按 “key1=value1&key2=value2“ 格式拼接，每个键值对之间以 符号“&”相连

**PS: 键值对顺序必选 键字母 ASCII 编码排序**

DEMO (JAVA 示例)

Map<String, String>transData = **new** TreeMap<String, String>();

transData.put("keyC", "CC");

transData.put("keyB", "");

transData.put("keyA", "AA");

StringBuilder stringBuilder = **new** StringBuilder();

**for** (Entry<String, String> item : transData.entrySet()) {

**if**(StringUtil.*notEmpty*(item.getValue())){

stringBuilder.append(item.getKey()).append('=')

.append(item.getValue()).append('&');

}

}

**if**(stringBuilder.length() > 0){

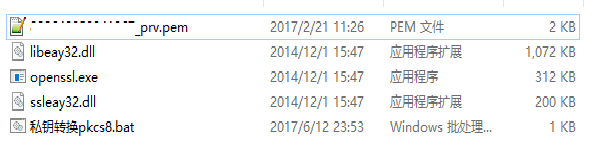
stringBuilder.setLength(stringBuilder.length()-1);

}

带签名明文：keyA=AA&keyC=CC

**签名算法(JAVA示例):**

1. 将商户密钥文件(\*\*\*\*\*\*\*\_prv.pem) 密钥文件放置于openssl 根目录



2.双击 私钥转换pkcs8.bat 文件生成 merchant\_pkcs8\_prv.pem 文件

3.代码示例（JAVA版本）

File privateFile = **new** File("/merchant\_pkcs8\_prv.pem");

FileInputStream fileInputStream = **new** FileInputStream(privateFile);

**byte**[]privateKeyContent = **new** **byte**[fileInputStream.available()];

fileInputStream.read(privateKeyContent);

fileInputStream.close();

String privateKeyStr = **new** String(privateKeyContent)

.replaceAll("-.\*", "");

privateKeyContent =

Base64.*getDecoder*().decode(privateKeyStr.replace("\n", ""));

PKCS8EncodedKeySpec keySpec = **new** PKCS8EncodedKeySpec(privateKeyContent);

KeyFactory keyFactory = KeyFactory.*getInstance*("RSA");

PrivateKey privateKey = keyFactory.generatePrivate(keySpec);

String context = "keyA=AA&keyC=CC";

java.security.Signature signature =

java.security.Signature.*getInstance*("SHA1WithRSA");

signature.initSign(privateKey);

signature.update(context.getBytes(Charset.*forName*("UTF8")));

String sign = Base64.*getEncoder*().encodeToString(signature.sign());

System.***out***.println("signature = " + sign);

**验证签名算法(JAVA示例):**

String context = "keyA=AA&keyC=CC";

String signatureStr = "签名字符串";

File publicFile = **new** File("/XXXXXXXXXXXXXXX\_pub.pem");

FileInputStream fileInputStream = **new** FileInputStream(publicFile);

**byte**[]publicKeyContent = **new** **byte**[fileInputStream.available()];

fileInputStream.read(publicKeyContent);

fileInputStream.close();

String publicKeyStr = **new** String(publicKeyContent)

.replaceAll("-.\*", "");

**byte**[] buffer = java.util.Base64.*getDecoder*().decode(publicKeyStr.replace("\r\n", "").trim());

KeyFactory keyFactory = KeyFactory.*getInstance*("RSA");

X509EncodedKeySpec keySpec = **new** X509EncodedKeySpec(buffer);

PublicKey publicKey = keyFactory.generatePublic(keySpec);

java.security.Signature signature = java.security.Signature.*getInstance*("SHA1WithRSA");

signature.initVerify(publicKey);

signature.update(context.getBytes(Charset.*forName*("UTF8")));

**boolean** bverify = signature.verify(Base64.*getDecoder*().decode(signatureStr));

System.***out***.println(bverify);

## 2.4 报文规范说明

### 2.4.1 符号说明

| 符号 | 含义 |
| --- | --- |
| M | 描述参数字段值必须上送至接口且值不能为空 |
| C | 描述参数字段在某些场景下可以选择为空或上送该字段 |

### 2.4.2 接口数据类型定义

| 数据类型 | 类型描述 |
| --- | --- |
| 基础数据类型： | |
| N(x) | 固定长度的数组 如 N4 允许上送 2017 |
| N(x,y) | 长度在 x~y 范围之间的数字 如 N1..2 允许上送 1 或 10 |
| A(x) | 固定长度位的字母 如 A5 允许上送 SALES |
| A(x,y) | 长度在 x~y 范围之间的字母 如 A1..2 允许上送 A 或 AB |
| ASC(x) | 固定长度的任意字符 |
| ASC(x,y) | 长度在x~y 范围之内的任意字符 |
| 时间类型： | |
| dd | 日 |
| MM | 月 |
| yyyy | 年 |
| HH | 时(24小时进制) 0-23时 |
| mm | 分 |
| ss | 秒 |

# 4.接口说明

### 4.1网银直连接口

| 参数名称 | 参数编码 | 属性 | 数据描述 | 数据类型 |
| --- | --- | --- | --- | --- |
| version | 版本号 | M | 固定值1.0.0 | ASC(5) |
| transType | 业务类型 | M | 固定值 SALES | A(4) |
| productId | 产品类型 | M | 0001 网关消费| 0121 微信H5 支付 0122 微信收银台扫码 | N(4) |
| merNo | 商户号 | M | 商户号 | N(15) |
| orderDate | 订单日期 | M | 订单交易日期 yyyyMMdd | N(8) |
| orderNo | 订单号 | M | 商户平台订单号 | ASC(40) |
| notifyUrl | 后台通知地址 | M | 用户完成支付后,服务器后台通知地址 | ASC(255) |
| returnUrl | 前台跳转地址 | M | 前台跳转地址 | ASC(255) |
| transAmt | 交易金额 | M | 分为单位如 100 代表 1.00元 | N(64) |
| bankCode | 银行编码 | O | 跳转目标银行银行编码 参考 附录二 网银直连时必填 | ASC(20) |
| connectType | 终端类型 | C | IOS\_WAP IOS系统浏览器支付 AND\_WAP ANDROID 系统浏览器支付 如果不上送**会影响成功率及可能被风控** | ASC(40) |
| commodityName | 产品名称 | C | 产品名称 | ASC(64) |
| commodityDetail | 产品详情 | C | 产品详情 真实交易的订单详情 如  交易金额 3.50元 commodityDetail 上送 可口可乐 如果改参数不上送 **会影响成功率及可能被风控** | ASC(255) |
| signature | 签名字段 | M | 参考 目录3.3 |  |

交易返回：

如果交易成功，返回银行跳转HTML

交易失败返回失败报文

| 参数名称 | 参数编码 | 属性 | 数据描述 | 数据类型 |
| --- | --- | --- | --- | --- |
| version | 版本号 | M | 固定值1.0.0 | ASC(5) |
| transType | 业务类型 | M | 根据上送交易类型返回 | A(4) |
| productId | 产品类型 | M | 根据上送交易类型返回 | N(4) |
| merNo | 商户号 | M | 商户号 | N(15) |
| orderDate | 订单日期 | M | 订单交易日期 yyyyMMdd | N(8) |
| orderNo | 订单号 | M | 商户平台订单号 | ASC(40) |
| notifyUrl | 后台通知地址 | M | 用户完成支付后,服务器后台通知地址 | ASC(255) |
| returnUrl | 前台跳转地址 | M | 前台跳转地址 | ASC(255) |
| transAmt | 交易金额 | M | 分为单位如 100 代表 1.00元 | N(64) |
| bankCode | 银行编码 | M | 跳转目标银行银行编码 参考 附录二 | ASC(20) |
| respCode | 返回码 | M | 见附录 一 | ASC(4) |
| respDesc | 返回描述 | C | 见附录 一 | ASC(1,64) |
| signature | 签名字段 | M | 参考 目录3.3 |  |

**异步回调通知**

当用户在支付界面完成支付后，通过商户上送报文请求时提供的 notifyUrl 字段向商户发送交易结果信息。

补发通知机制，如果发送失败后一定时间后会在此发送通知，当商户成功接收到发送的回调信息，请在response 报文体中 回复一下内容 “success”

当服务器接收到 success 内容后 将不会继续对商户发送该订单的后台服务通知

**PS:请正视交易异步回调通讯，如发生多次超时或无法访问情况，该通讯地址可能被拉入通讯黑名单。**

| 参数名称 | 参数编码 | 属性 | 数据描述 | 数据类型 |
| --- | --- | --- | --- | --- |
| transType | 业务类型 | M | 根据上送交易类型返回 | A(4) |
| productId | 产品类型 | M | 根据上送交易类型返回 | N(4) |
| merNo | 商户号 | M | 商户号 | N(15) |
| orderDate | 订单日期 | M | 订单交易日期 yyyyMMdd | N(8) |
| orderNo | 订单号 | M | 商户平台订单号 | ASC(40) |
| transAmt | 交易金额 | M | 分为单位如 100 代表 1.00元 | N(64) |
| serialId | 流水号 | M | 平台产生的交易流水号 | ASC(32) |
| respCode | 返回码 | M | 见附录 一 | ASC(4) |
| respDesc | 返回描述 | C | 见附录 一 | ASC(1,64) |
| signature | 签名数据 | M | 通过 RSA 公钥验证签名 | ASC(1024) |

# 附录一 返回码描述

PS：所有P开头的状态吗均为需要后续操作或未知状态

| 返回码 | 描述 | 返回码 | 数据类型 |
| --- | --- | --- | --- |
| **0000** | **交易成功** | T039 | 原交易不支持撤销 |
| 未知类： | | T040 | 上一笔撤销交易正在处理中 |
| P000 | 交易处理中 | T041 | 上一笔退货交易正在处理中 |
| P999 | 交易结果未知 | T042 | 余额不足 |
| 验证失败类 | | T043 | 生成代付单失败 |
| T001 | 请求参数非法 | T044 | 商品订单状态异常 |
| T002 | 商户号不能为空 | T045 | 支付订单状态异常 |
| T003 | 商户订单号不能为空 | T046 | 交易限额超限 |
| T004 | 银行预留手机号不能为空 | T047 | 交易金额与原交易不符 |
| T005 | 交易金额不能为空 | T048 | 联行号不存在 |
| T006 | 卡号不能为空 | T049 | 该时间段不允许交易 |
| T007 | CVN2不能为空 | T050 | 订单日期无效。 |
| T008 | 信用卡有效期不能为空 | T051 | 短信验证码请求过于频繁,请稍后再试 |
| T009 | 验签字段不能为空 | T052 | 手机号非法 |
| T010 | 商户信息不存在 | T053 | 短信验证码错误次数过多,请稍候再试 |
| T011 | 商户状态异常 | T054 | 未获取短信验证码或短信验证码为空 |
| T012 | 校验签名失败 | T055 | 短信验证码不能为空 |
| T013 | 商户密钥信息不存在 | T056 | 短信验证码错误 |
| T014 | 商户密钥已失效 | T057 | 文件获取日期格式错误 |
| T015 | 获取商户密钥ID失败 | T058 | 查询日期范围不能超过 |
| T016 | 产品信息不存在 | T059 | 获取卡号信息异常 |
| T017 | 产品信息状态异常 | T060 | 当前交易无需发送短信 |
| T018 | 未授权的交易 | T061 | 交易不支持退款 |
| T019 | 渠道约束信息为空 | T062 | 获取微信授权信息失败 |
| T020 | 不支持的卡类型 | T062 | 获取微信配置信息失败 |
| T021 | 不支持借记卡交易 | T064 | 微信对账ftp创建文件失败 |
| T022 | 不支持贷记卡交易 | T065 | 短信验证码超时,请重新获取 |
| T023 | 订单号重复 | T066 | 卡类型不匹配 |
| T024 | 订单信息保存失败 | T067 | 银行卡签约失败 |
| T027 | 未知交易类型" | T068 | 交易数据不正确 |
| T028 | 原交易不存在 | C001 | 验证银行结果签名失败 |
| T029 | 原交易状态异常 | C002 | 支付信息保存失败 |
| T030 | 交易金额超限 | C003 | CVN2错误或疑似作弊(详情请联系发卡行) |
| T031 | 路由筛选失败 | C004 | 卡有效期错误(详情请联系发卡行) |
| T032 | 交易金额非法 | C005 | 支付失败、业务类型受限(详情请联系发卡行) |
| T033 | 银行卡信息保存失败 | C006 | 卡信息或状态异常或功能受限(详情请联系发卡银行) |
| T034 | 当天交易不支持退款 | C007 | 交易金额超限(详情请联系发卡银行) |
| T035 | 原交易已撤销 | C008 | 该卡有作弊嫌疑或有相关限制(详情请联系发卡银行) |
| T036 | 非当天交易不支持撤销 | W001 | 交易金额或交易次数超限 |
| T037 | 原交易已发起撤销或退货 | W002 | 交易金额超限 |
| T038 | 不支持的银行卡 | W003 | 交易次数超限 |
| 9996 | 交易已退货 | 9999 | 系统异常 |
| 9998 | 交易失败 |  |  |

# 附录二

| 银行编码 | 银行名称 | 银行编码 | 银行名称 |
| --- | --- | --- | --- |
| PSBC | 邮储银行 | ABC | 农业银行 |
| ICBC | 工商银行 | BOC | 中国银行 |
| CCB | 建设银行 | BOCM | 交通银行 |
| CEBB | 光大银行 | CMBC | 民生银行 |
| GDB | 广发银行 | CMB | 招商银行 |
| BOS | 上海银行 | BOB | 北京银行 |
| CIB | 兴业银行 | CNCB | 中信银行 |
| GZCB | 广州银行 | HXB | 华夏银行 |
| BOB | 北京银行 | PAB | 平安银行 |
| BOS | 上海银行 | SPDB | 浦发银行 |
| GDB | 广发银行 |  |  |