

## 1.1 文档说明

本文档用于描述了一卡通对第三方应用提供的通用支付api接口文档，供第三方系统（前置和应用）对接参考。

## 1.2 约定

1. 传输编码统一为utf-8
2. RSA 加密算法为本接口采用的非对称加密算法，`SIGN_ALGORITHMS ="SHA1WithRSA"`;
3. RSA公钥由本系统提供，本系统返回信息的签名数据，第三方系统得到后通过公钥校验算法，验证返回数据的合法性
4. HMAC加密算法为用户请求本系统采用的加密算法，`SIGN_ALGORITHMS ="HMAC-SHA1"`;
5. 具体签名生成参考后面的签名章节
6. retcode等于0表示成功，非0表示失败，失败具体信息查看retmsg
7. 密码加密的原始串为timestamp:pwd

## 2.1 通用账户查询

请求URL:

- `https://ip:port/epayapi/services/thirdparty/common/accountquery`

请求方式/格式:

- POST
- application/x-www-form-urlencoded

请求参数:

参数名	类型	必选	说明
partner_id	String	是	本系统分配给各个接入应用的合作伙伴id号
cardphysid	String	二选一	查询对象的卡物理id
stuempno	String	二选一	查询对象的学号
timestamp	String	是	时间戳格式为yyyyMMddhh24miss
sign	String	是	签名
sign_method	String	是	参数的加密方法选择，可选值是：HMAC 加密方式为HMAC-SHA1

请求内容示例:

```
URI: https://ip:port/epayapi/services/thirdparty/common/accountquery

partner_id:10000
stuempno:09893092
timestamp:20150119130901
```

```
sign:5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method:HMAC
```

#### 返回示例:

- 成功

```
{
  "retcode": "0",
  "retmsg": "query success",
  "stuempno": "09893092",
  "username": "王二小",
  "onceTimeLimit": "5000",
  "dayTotalLimit": "10000",
  "expiredate": "20191010",
  "balance": 4850,
  "status": "normal",
  "timestamp": "20160606150902",
  "sign": "hjAtBGygGjFy0KEZmWdxi1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUsHnu5o3kAZ4U32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkeINHjyHsnxBr44G2UhyNxpUj+s7s0a7ggjbiMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSWd7bsf0J6HYw2spvu3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze//SvCqpCJfYt0rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhl+HwZJcag==",
  "sign_method": "RSA"
}
```

- 失败

```
{
  "retcode": "1",
  "retmsg": "account not exsit",
  "timestamp": "20160606150902",
  "sign": "hjAtBGygGjFy0KEZmWdxi1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUsHnu5o3kAZ4U32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkeINHjyHsnxBr44G2UhyNxpUj+s7s0a7ggjbiMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSWd7bsf0J6HYw2spvu3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze//SvCqpCJfYt0rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhl+HwZJcag==",
  "sign_method": "RSA"
}
```

#### 返回参数说明:

参数名	类型	说明
retcode	String	返回码 (0=成功, 其他为失败)

参数名	类型	说明
retmsg	String	返回消息
stuempno	String	学号
username	String	姓名
balance	Integer	余额(分)
expiredate	String	卡有效期
status	String	状态(正常)

## 2.2 通用支付

请求URL:

- <https://ip:port/epayapi/services/thirdparty/common/pay>

请求方式/格式:

- POST
- application/x-www-form-urlencoded

请求参数:

参数名	类型	必选	说明
partner_id	String	是	本系统分配给各个接入应用的合作伙伴id号
cardphyid	String	二选 —	查询对象的卡物理id,需要单独校验实体卡
stuempno	String	二选 —	发起消费对象的学号
tradenno	String(32)	是	第三方系统唯一流水号
tradenname	String(60)	是	交易名称
amount	Integer	是	消费金额 (分)
calcmanagefee	String	否	是否计算搭伙费T计算, F或不传不计算
limitflag	String	否	是否打开消费限额, on/off
encryptpwd	Sting	否	卡片密码, 消费超限额时需要密码校验, 加密算法 AES/ECB/PKCS5Padding
termid	Integer	否	终端号, 实体卡消费时需要
cardmac	Sting	否	卡片mac校验值, 实体卡消费时需要
posmac	Sting	否	终端mac校验值, 实体卡消费时需要
termdate	Sting	否	终端交易日期yyyyMMdd, 实体卡消费时需要

参数名	类型	必选	说明
termtime	String	否	终端交易时间hh24miss，实体卡消费时需要
timestamp	String	是	时间戳格式为yyyyMMddhh24miss
sign	String	是	签名
sign_method	String	是	参数的加密方法选择，可选值是：HMAC 加密方式为HMAC-SHA1

请求内容示例:

```
URI: https://ip:port/epayapi/services/thirdparty/common/pay

partner_id:10000
stuempno:09893092
tradenno:20160607000001
trandename:print fee
amount: 2000
timestamp:20150119130901
sign:5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method:HMAC
```

返回示例:

- 成功

```
{
  "retcode":"0",
  "retmsg":"success",
  "tradenno":"20160607000001",
  "balance":4850,
  "refno":"20160605190200000001",
  "needpwdconfirm":false,
  "nexttermseqno":2,
  "timestamp":"20160606150902",
  "sign":"hjAtBGygGjFy0KEZmWdxi1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUs
Hnu5o3kAZ4U32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkEINHjyHsnxBr44G2UhyNx
pLUj+s7s0a7ggjbiMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSWd7bsf
0J6HYw2spvu3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze/
/SvCqpCJfYt0rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhL+HwZJcag==",
  "sign_method": "RSA"
}
```

- 失败

```
{
  "retcode":"1",
```

```

    "retmsg": "账户余额不足",
    "timestamp": "20160606150902",
    "needpwdconfirm": false,
    "sign":
    "hjAtBGygGjFy0KEZmWdxi1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUsHnu5o3kAZ4U
    32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkeINHjyHsnxBr44G2UhyNxpUj+s7s0a7
    ggjbIMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSwD7bsf0J6HYw2spvu
    3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze//SvCqpCJfYt
    00rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhl+HwZJcag==",
    "sign_method": "RSA"
  }

```

```

{
  "retcode": "1",
  "retmsg": "消费超限, 请输入密码",
  "needpwdconfirm": true,
  "timestamp": "20160606150902",
  "sign":
  "hjAtBGygGjFy0KEZmWdxi1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUsHnu5o3kAZ4U
  32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkeINHjyHsnxBr44G2UhyNxpUj+s7s0a7
  ggjbIMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSwD7bsf0J6HYw2spvu
  3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze//SvCqpCJfYt
  00rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhl+HwZJcag==",
  "sign_method": "RSA"
}

```

#### 返回参数说明:

参数名	类型	说明
retcode	String	返回码 (0=成功, 其他为失败)
retmsg	String	返回消息
tradenno	String	第三方流水号
refno	String	本系统生成流水号
balance	Integer	余额(分)
needpwdconfirm	Boolean	是否需要输入密码
nexttermseqno	Integer	下次终端流水号

密码加密算法为 AES/ECB/PKCS5Padding 秘钥线下约定

## 2.3 通用支付冲正

请求URL:

- <https://ip:port/epayapi/services/thirdparty/common/payreverse>

#### 请求方式/格式:

- POST
- application/x-www-form-urlencoded

#### 请求参数:

参数名	类型	必选	说明
partner_id	String	是	本系统分配给各个接入应用的合作伙伴id号
cardphyid	String	二选 —	查询对象的卡物理id
stuempno	String	二选 —	发起消费对象的学号
termid	Integer	否	终端号, 实体卡消费冲正时需要
tradeno	String(32)	是	第三方系统唯一流水号
reversetradeno	String(32)	是	第三方系统需要冲正的流水号
timestamp	String	是	时间戳格式为yyyyMMddhh24miss
sign	String	是	签名
sign_method	String	是	参数的加密方法选择, 可选值是: HMAC 加密方式为HMAC-SHA1

#### 请求内容示例:

```
URI: https://ip:port/epayapi/services/thirdparty/common/payreverse

partner_id:10000
stuempno:09893092
tradeno:20160607000002
tradeno:20160607000001
timestamp:20150119130901
sign:5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method:HMAC
```

#### 返回示例:

- 成功

```
{
  "retcode":"0",
  "retmsg":"success",
  "tradeno":"20160607000002",
```

```

    "nexttermseqno":2,
    "timestamp":"20160606150902",
    "sign":"hjAtBGygGjFy0KEZmWdxi1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUsHnu5o3kAZ4U32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkEINHjyHsnxBr44G2UhyNxpUj+s7s0a7ggjbIMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSWd7bsf0J6HYw2spvu3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze//SvCqpCJfYt00rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhL+HwZJcag==",
    "sign_method": "RSA"
}

```

- 失败

```

{
  "retcode":"1",
  "retmsg":"账户余额不足",
  "tradenno":"20160607000002",
  "timestamp":"20160606150902",
  "sign":
  "hjAtBGygGjFy0KEZmWdxi1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUsHnu5o3kAZ4U32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkEINHjyHsnxBr44G2UhyNxpUj+s7s0a7ggjbIMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSWd7bsf0J6HYw2spvu3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze//SvCqpCJfYt00rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhL+HwZJcag==",
  "sign_method":"RSA"
}

```

返回参数说明:

参数名	类型	说明
retcode	String	返回码（0=成功，其他为失败）
retmsg	String	返回消息
tradenno	String	第三方流水号

## 2.4 扣款流水查询

请求URL:

- <https://ip:port/epayapi/services/thirdparty/common/payquery>

请求方式/格式:

- POST
- application/x-www-form-urlencoded

请求参数:

参数名	类型	必选	说明
partner_id	String	是	本系统分配给各个接入应用的合作伙伴id号
stuempno	String	是	发起消费对象的学号
tradeno	String(32)	是	第三方系统唯一流水号
timestamp	String	是	时间戳格式为yyyyMMddhh24miss
sign	String	是	签名
sign_method	String	是	参数的加密方法选择，可选值是：HMAC 加密方式为HMAC-SHA1

请求内容示例:

```
URI: https://ip:port/epayapi/services/thirdparty/common/payquery
```

```
partner_id:10000
stuempno:09893092
tradeno:20160607000001
timestamp:20150119130901
sign:5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method:HMAC
```

返回示例:

- 成功

```
{
  "retcode":"0",
  "retmsg":"success"
  "tradeno":"20160607000001",
  "balance":4850,
  "refno":"20160605190200000001",
  "tradestatus": "success",
  "paytime":"20160606150902",
  "timestamp":"20160606150902",
  "sign":"hjAtBGygGjFy0KEZmWdxil1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUs
Hnu5o3kAZ4U32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzkEINHjyHsnxBr44G2UhyNx
pLUj+s7s0a7ggjbIMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSWd7bsf
0J6HYw2spvu3ptAo2RTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze/
/SvCqpCJfYt00rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhL+HwZJcag==",
  "sign_method": "RSA"
}
```

- 失败

```
{
  "retcode": "0",
  "retmsg": "成功",
  "tradestatus": "fail",
  "tradeno": "20160607000001",
  "timestamp": "20160606150902",
  "sign":
    "hjAtBGygGjFy0KEZmWdx1tk51/xqLYTr5tm2+zQVfD1RHRzGVC9GNm4ZaXnUsHnu5o3kAZ4U
    32m0YNiSFpncs0pl1AnnCf1n2hhn39ruABsvD//GzKEINHjyHsnxBr44G2UhyNxpLUj+s7s0a7
    ggjbiMk+BYC3u7Vku9/l+kZ0TkEyqs+wZKumfByMZGvKVkY2c2Nr67RaSWd7bsf0J6HYw2spvu
    3ptAo2RTTrfCxZ0Kn5DcSDm4cagzPMZSHxB8isqqNGrdUaAU0wzzC7704iA5zze//SvCqpCJfYt
    00rH0iJEN5kw1d4KESrKGMG6bezLXzAp0LbNlhl+HwZJcag==",
  "sign_method": "RSA"
}
```

#### 返回参数说明:

参数名	类型	说明
retcode	String	返回码 (0=成功, 1 - 流水不存在)
retmsg	String	返回消息
tradeno	String	第三方流水号
refno	String	本系统生成流水号
tradestatus	String	交易状态, success - 成功, fail - 失败
paytime	String	支付时间, 交易成功才会返回
balance	Integer	余额(分)

## 2.5 交易明细下载

#### 请求URL:

- [https://ip:port/epayapi/services/thirdparty/common/query\\_bill\\_list](https://ip:port/epayapi/services/thirdparty/common/query_bill_list)

#### 请求方式/格式:

- POST
- application/x-www-form-urlencoded

#### 请求参数:

参数名	类型	必选	说明
partner_id	String	是	本系统分配给各个接入应用的合作伙伴id号
accdate	String	是	账单日期
pageno	String	否	页码, 默认1

参数名	类型	必选	说明
pagesize	String	否	每页行数,默认10, 范围10-500
timestamp	String	是	时间戳格式为yyyyMMddhh24miss
sign	String	是	签名
sign_method	String	是	HMAC 加密方式为HMAC-SHA1

请求内容示例:

```
URI: https://ip:port/epayapi/services/thirdparty/common/query_bill_list

partner_id:10000
accdate:20190701
pageno:1
pagesize:10
timestamp:20150119130901
sign:5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method:HMAC
```

返回示例:

- 成功

```
{
  "retcode": "0",
  "retmsg": "查询成功",
  "data": {
    "totalCount": 11,
    "pageSize": 10,
    "pageNo": 1,
    "list": [
      {
        "refno": "20170809160440168058",
        "tradeno": "20170809160440168058",
        "paytime": "20170809160449",
        "billname": "IC卡消费",
        "amount": 1,
        "billtype": "联机刷卡消费",
        "billstatus": 2,
        "tradetype": "2",
        "trade": "6630",
        "termname": "第一食堂POS机1",
        "aftbal": 65.21
      },
      {
        "refno": "20170809103401168050",
        "tradeno": "20170809103401168050",
        "paytime": "20170809103413",
```

```
        "billname": "IC卡消费",
        "amount": 1,
        "billtype": "联机刷卡消费",
        "billstatus": 2,
        "tradetype": "2",
        "trade": "6630",
        "termname": "第一食堂POS机3",
        "aftbal": 66.21
    },
    {
        "refno": "20170809101226168047",
        "trade": "20170809101226168047",
        "paytime": "20170809101238",
        "billname": "IC卡消费",
        "amount": 1,
        "billtype": "联机刷卡消费",
        "billstatus": 2,
        "tradetype": "2",
        "trade": "6630",
        "termname": "第一食堂POS机2",
        "aftbal": 67.21
    },
    {
        "refno": "20170809101217168046",
        "trade": "20170809101217168046",
        "paytime": "20170809101229",
        "billname": "IC卡消费",
        "amount": 1,
        "billtype": "联机刷卡消费",
        "billstatus": 2,
        "tradetype": "2",
        "trade": "6630",
        "termname": "第一食堂POS机1",
        "aftbal": 68.21
    },
    {
        "refno": "20170809100854168045",
        "trade": "20170809100854168045",
        "paytime": "20170809100905",
        "billname": "IC卡消费",
        "amount": 1,
        "billtype": "联机刷卡消费",
        "billstatus": 2,
        "tradetype": "2",
        "trade": "6630",
        "termname": "第一食堂POS机1",
        "aftbal": 69.21
    },
    {
        "refno": "20170809100845168043",
        "trade": "20170809100845168043",
        "paytime": "20170809100824",
        "billname": "IC卡消费",
        "amount": 1,
```

```

        "billtype": "联机刷卡消费",
        "billstatus": 2,
        "tradetype": "2",
        "tradeCode": "6630",
        "termname": "第一食堂POS机1",
        "aftbal": 70.21
    },
    {
        "refno": "20170809100845168044",
        "traden": "20170809100845168044",
        "paytime": "20170809100824",
        "billname": "IC卡消费",
        "amount": 1,
        "billtype": "联机刷卡消费撤销",
        "billstatus": 2,
        "tradetype": "1",
        "tradeCode": "6632",
        "termname": "第一食堂POS机1",
        "aftbal": 71.21
    },
    {
        "refno": "20170807145900002705",
        "traden": "20170807145900002705",
        "paytime": "20170807145958",
        "billname": "现金充值",
        "amount": 70,
        "billtype": "钱包充值",
        "billstatus": 2,
        "tradetype": "1",
        "tradeCode": "6500",
        "termname": "第一食堂充值机1",
        "aftbal": 70.21
    }
],
"firstResult": 0,
"totalPage": 2,
"firstPage": true,
"lastPage": false,
"nextPage": 2,
"prePage": 1
}
}

```

- 失败

```

{
    "retcode": "304",
    "retmsg": "签名验证失败",
    "data": null
}

```

### 返回参数说明:

参数名	类型	说明
retcode	String	返回码 (0=成功, 其他为失败)
retmsg	String	返回消息
data	list	流水明细集合

### 流水明细说明:

参数名	类型	说明
refno	String	交易参考号
tradeno	String	第三方交易流水号
paytime	String	交易时间 yyyyMMddhh24miss
billname	String	交易名称
amount	Double	交易金额(元)
billtype	String	交易类型
billstatus	String	状态 2-成功 3-失败
tradetype	String	交易方向 1-充值 2-消费
tradeCode	String	交易码
termname	String	终端名称
aftbal	Double	交易后余额(元)

## 附录A-用户请求HMAC签名算法

- 签名方式: `hmac-sha1`
- 签名密钥由本系统统一线下提供
- 签名校验的通用步骤如下:

\*\*第一步, \*\*设所有发送或者接收到的数据为集合M, 将集合M内非空参数值的参数按照参数名ASCII码从小到大排序 (字典序), 使用URL键值对的格式 (即key1=value1&key2=value2...) 拼接成字符串stringA。

假设传送的参数如下:

```
partner_id:10000
stuempno:09893092
tradeno:20160607000001
trandename:printfee
```

```
amount: 2000
timestamp: 20150119130901
sign: 5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method: HMAC
```

对参数按照key=value的格式，并按照参数名ASCII字典序排序如下：

```
amount=2000&partner_id=10000&sign_method=HMAC&stuempno=09893092
&timestamp=20150119130901&tradeno=20160607000001&trandename=printfee
```

**\*\*特别注意以下重要规则：\*\***

- 参数名ASCII码从小到大排序（字典序）；
- 如果参数的值为空不参与签名；
- 参数名区分大小写；
- 传送的sign参数不参与签名，用该sign值作校验。

**\*\*第二步，\*\***用密钥secretkey对stringA字符串，进行hmac-sha1签名，得到sign值signValue。signValue最后采用十六进制小写hex编码生成签名字符串。

## 附录B-服务端返回数据RSA签名校验算法

- 签名方式：SHA1withRSA
- 签名校验的公钥key为本系统统一线下提供。
- 签名校验的通用步骤如下：

**\*\*第一步，\*\***设所有发送或者接收到的数据为集合M，将集合M内非空参数值的参数按照参数名ASCII码从小到大排序（字典序），使用URL键值对的格式（即key1=value1&key2=value2...）拼接成字符串stringA。

**特别注意以下重要规则：**

- 参数名ASCII码从小到大排序（字典序）；
- 如果参数的值为空不参与签名；
- 参数名区分大小写；
- 传送的sign参数不参与签名，用该sign值作校验。

**\*\*第二步，\*\***对sign值进行base64解码，用本系统提供的公钥key对sign签名值解码后的数据基于stringA字符串，进行SHA1withRSA签名验证

举例：

假设传送的参数如下：

```
retcode:1  
retmsg:账户余额不足  
timestamp:20160513155100  
sign_mehtod:RSA
```

对参数按照key=value的格式，并按照参数名ASCII字典序排序如下：

```
retcode=1&retmsg=账户余额不足&timestamp=20160513155100&sign_mehtod=RSA
```