
title: 一卡通v5二维码申码接口文档 date: 2019-10-21 tags:

1.1 文档说明

本文档用于描述了一卡通二维码申码相关接口，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. HMAC加密算法为用户请求本系统采用的加密算法，`SIGN_ALGORITHMS ="HMAC-SHA1"`;
3. 具体签名生成参考后面的签名章节
4. retcode等于0表示成功，非0表示失败，失败具体信息查看retmsg

2.1 二维码申码

请求URL:

- `https://ip:port/epay/thirdapp/precreateqrcode`

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 |
| nonce | String | 是 | 随机数，用于判重 |
| amount | Integer | 是 | 消费金额(分) |
| 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/epay/thirdapp/precreateqrcode
```

```
partner_id:10000
nonce:xzf902100af19
amount:2000
timestamp:20150119130901
sign:5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method:HMAC
```

返回示例:

- 成功

```
{
  "retcode": "0",
  "retmsg": "初始化成功",
  "qrcode": "https://abc.com/epay/thirdapp/qrpayscan?
q=ab39402bcc1029281c",
  "refno": "2019080910201002000001"
}
```

- 失败

```
{
  "retcode": "1",
  "retmsg": "account not exist"
}
```

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 |
| refno | String | 交易参考号 |
| qrcode | String | 二维码内容, 第三方自己转成二维码图片 |

2.2 二维码扫码支付查询

请求URL:

- <https://ip:port/epay/thirdapp/precreateqrcodequery>

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 |

| 参数名 | 类型 | 必选 | 说明 |
|-------------|---------|----|------------------------------------|
| refno | String | 是 | 初始化返回的交易参考号 |
| amount | Integer | 是 | 消费金额(分) |
| 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/epay/thirdapp/precreateqrcodequery

```
partner_id:10000
amount:2000
refno:201906101000001
timestamp:20150119130901
sign:5195f9b9116e4adf67eeebc9935d33dc683f677d
sign_method:HMAC
```

返回示例:

- 成功

```
{
  "retcode":"0",
  "retmsg":"查询成功",
  "refno":"201906101000001",
  "dtlstatus":"success",
  "amount":2100,
  "balance":5200,
  "stuempno":"03902910",
  "custname":"王二"
}
```

- 失败

```
{
  "retcode":"1",
  "retmsg":"流水不存在"
}
```

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 |
| refno | String | 交易参考号 |
| dtlstatus | String | 流水状态,init=初始化未支付， success=成功,paying=支付中（需要查询确认）,fail=失败 |
| amount | Integer | 交易金额 |
| balance | Integer | 余额 |
| stuempno | String | 用户学号 |
| custname | String | 用户姓名 |

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
qrcode:cS2nsBRzhW72lQgcGdI6s64YSaaWnxlWtIiU=
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：

```
partner_id=10000&qrcode=cS2nsBRzhW72lQgcGdI6s64YSaaWnxlWtIiU=
&sign_method=HMAC&timestamp=20150119130901
```

****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。