**寄售接口**

**开发文档**

**版本（1.0）**

**文档历史**

|  |  |  |
| --- | --- | --- |
| **修订日期** | **修订内容** | **修订版本** |
| 2024-11-22 | 创建 | 1．0 |
|  |  |  |
|  |  |  |

**目录**

[1. 引言 4](#_Toc85757015)

[1.1. 编写目的 4](#_Toc85757016)

[2. 协议概述 5](#_Toc85757017)

[2.1. 协议列表 5](#_Toc85757018)

[3. 接口商户准备文档 5](#_Toc85757019)

[3.1. 商户准备工作 5](#_Toc85757020)

[4. 接口调用说明 6](#_Toc85757021)

[4.1. 报文结构 6](#_Toc85757022)

[4.1.1. 报文结构说明 6](#_Toc85757023)

[4.1.2. 请求报文(API公共报文) 6](#_Toc85757024)

[4.1.3. 响应报文(API公共报文) 6](#_Toc85757025)

[4.2. 支付接口调用（消费） 7](#_Toc85757026)

[4.2.1. 请求报文 7](#_Toc85757027)

[4.2.2. 响应报文 8](#_Toc85757028)

[4.3. 查询接口说明 9](#_Toc85757029)

[4.3.1. 请求报文 9](#_Toc85757030)

[4.3.2. 响应报文 9](#_Toc85757031)

[4.4. 接口通知参数说明 10](#_Toc85757032)

# 引言

## 编写目的

本系统采用公开标准Web方式由合作伙伴调用卡寄售接口地址，实现将各种市面上一卡通的卡进行兑现或者结算。

# 协议概述

## 协议列表

|  |  |
| --- | --- |
| **协议名称** | **说明** |
| 寄售协议 | 寄售等开通服务的协议 |

协议的交互逻辑

客户端

服务端

商家ID、卡号、卡密、账号等参数

结果代码、充值金额等参数

卡支付接口流程图

# 售卡接口商户准备文档

## 商户准备工作

1. 注册到成为商户，联系商务合作人员。
2. 联系技术中心相关技术人员进行调试。
3. 对产品敏感信息双方协商进行对称加密处理。

# 寄售接口调用说明

## 报文结构

### 报文结构说明

平台提供的 API 接口有统一的明文报文，开发人员将接口字段信息按照要求签名加密后放入报文以form方式提交到服务器，服务器收到请求返回报文。

MD5在线校验地址：<https://www.sojson.com/encrypt_md5.html>

**商户订单号是唯一的，商户要做验证处理，防止重复充值，否则造成的损失商户自己承担。**

### 响应报文(API公共报文)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 中文名称 | 字段名 | 长度限制 | 必输 | 说明 |
| 返回码 | code | 4 | Y | 详见返回码说明 1成功 其它失败 |
| 返回信息 | msg | NA | Y | 返回描述 |
| 业务数据 | data | NA | Y | 业务具体数据 |
| 签名 | sign | NA | N | 业务数据的全包校验MD5值，具体生成方式4.1.3 |

### MD5生成方式

1. 除sign字段外，所有请求参数按照文档提供的排序进行签名。
2. 生成签名原始串后使用的字符集为UTF-8，加密算法为MD5 签名转MD5大写。

# 4.1.4 AES加密卡号卡密

使用商户加密密钥生成卡号卡密参数，保护卡密信息不被泄露，使用AES算法加密，ECB工作模式，PKCS5Padding方式填充，将加密结果转为16进制值。

签名时字符串使用的字符集为UTF-8。

**4.1.5 测试地址**

**Php 加密代码示例**

$data = openssl\_encrypt(“加密字符串”, 'AES-128-ECB', “加密秘钥”, 0);

$datakey= bin2hex($data);//把结果base64解码转16进制 得到加密后的数据

**JAVA 加密示例**

public static String Encrypt(String sSrc, String sKey) {

 try {

 if (sKey == null) {

 System.*out*.print("Key为空null");

 return null;

 }

 // 判断Key是否为16位

 if (sKey.length() != 16) {

 System.*out*.print("Key长度不是16位");

 return null;

 }

 byte[] raw = sKey.getBytes(StandardCharsets.*UTF\_8*);

 SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");

 Cipher cipher = Cipher.*getInstance*("AES/ECB/PKCS5Padding");//"算法/模式/补码方式"

 cipher.init(Cipher.*ENCRYPT\_MODE*, skeySpec);

 byte[] encrypted = cipher.doFinal(sSrc.getBytes(StandardCharsets.*UTF\_8*));

 String newstr= new Base64().encodeToString(encrypted);//此处使用BASE64做转码功能，同时能起到2次加密的作用。

 return *str2HexStr*(newstr).toLowerCase();

 }catch (Exception e){

 *log*.error("加密错误：{}",sSrc);

 return "";

 }

}

*/\*\**

 *\* 字符串转换成为16进制(无需Unicode编码)*

 *\*/*

public static String str2HexStr(String str) {

 char[] chars = "0123456789ABCDEF".toCharArray();

 StringBuilder sb = new StringBuilder("");

 byte[] bs = str.getBytes();

 int bit;

 for (int i = 0; i < bs.length; i++) {

 bit = (bs[i] & 0x0f0) >> 4;

 sb.append(chars[bit]);

 bit = bs[i] & 0x0f;

 sb.append(chars[bit]);

 }

 return sb.toString().trim();

}

## 接口调用

### 请求报文

通过此接口将要寄售的卡信息以post方式提交。

调用方式：POST

**接口调用地址:** [http://47.92.163.52:9896/api/consignmentOrders](http://47.92.108.245:9093/api/consigmentOrders)

固定签名排序：customerId={customerId}&timestamp={timestamp}&orderId={orderId}&batchno={batchno}&productCode={productCode}&cardNumber={cardNumber}&cardPassword={cardPassword}&faceMoney={faceMoney}&notifyUrl={notifyUrl}&key={签名秘钥}

**接口输入参数说明:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | **字段名称** | **类型** | **是否必填** | **说明** |
| **商户号** | customerId | long | **必填** | 商家在平台的商户号 |
| **校验时间戳** | timestamp | long | **必填** | Unix timestamp |
| **商户订单号** | orderId | string | **必填** | 商户订单编号不大于50位长度,必须唯一 |
| **提交类型** | productType | int | **必填** | 0:卡号卡密 1：图片 2：兑换链接 |
| **批次订单号** | batchno | string | **必填** | 批次订单号,可以重复，便于对整个批次订单进行管理,单个批次最多不能超过2000张卡 |
| **产品代码** | productCode |  string | **必填** | 登陆商户端查看 |
| **自定义费率** | customizeRate | Float | **可选** | 当产品支持自定义费率的时候可以传 最多小数点后1位，比如98.6 折 传98.6 |
| **卡号** | cardNumber | string | **可选** | **无卡号不用填 不用加密** |
| **卡密码** | cardPassword | string | **必填** | 点卡卡密，以AES方式加密，参照4.1.4. |
| **CVV码** | seCode | string | **可选** | 部分产品可能需要传cvv码，没有可以不传（不参与签名） |
| **卡面值** | faceMoney | int | **必填** | 卡面的金额 |
| **通知地址** | notifyUrl | string | **可选** | 单据处理后用来通知商户的地址**,**URL参数是以http://开头的完整URL地址(后台处理)不填写则不通知； |
| **拓展字段** | custom | string | **可选** | 商户自定义参数或扩展参数，接口按原值返回（不参与签名） |
| **签名** | sign | string | **必填** | 参照4.1.3 |

### 响应报文

|  |  |  |  |
| --- | --- | --- | --- |
| **中文名称** | **字段名称** | **类型** | **说明** |
| **返回码** | **code** | int | 代码1=寄售成功，单据已被接收。 |
| **返回信息** | **msg** | string | 返回消息 |
| **收单订单号** | **sysOrderId** | string | 寄售成功才会有该参数 |

**code=1时， 单据才合法被接收；**

**code=-1时， 单据接收失败，同时msg将返回具体失败信息；**

## 查询接口说明

### 请求报文

**接口调用地址:**  http://47.92.163.52:9896/api/searchOrders

固定签名排序：customerId={customerId}&timestamp={timestamp}&orderId={orderId}&key={签名秘钥}

**接口输入参数说明：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **中文名称** | **字段名称** | **类型** | **是否必填** | **说明** |
| **商户号** | customerId | string | **必填** | 商家在平台的商户号 |
| **校验时间戳** | timestamp | int | **必填** | Unix timestamp |
| **商户订单号** | orderId | string | **必填** | 提交订成功时返回的系统订单号**sysOrderId** |
| **签名** | sign | string | **必填** | 参照4.1.3 |

### 响应报文

|  |  |  |  |
| --- | --- | --- | --- |
| **中文名称** | **字段名称** | **类型** | **说明** |
| **返回码** | code | String | 1=查询成功 2=订单不存在 其他查询失败 |
| **返回信息** | msg | string | 返回消息，如果查询失败返回失败消息 |
| **返回数据** | data | Array | 以下为data数据内容 |
| **商户号** | customerId | string | 商户号 |
| **订单号** | sysOrderId | string | 该订单在平台的寄售订单号 |
| **商户订单号** | orderId | string | 商户上传的订单号 |
| **订单状态** | status | string | 1处理中2成功3失败 |
| **备注** | message | string | 订单信息 |
| **卡面金额** | faceMoney | int | 商户上传卡面值金额 |
| **结算面值** | realMoney | int | 卡的真实面值 |
| **结算金额** | amount | float | 卡的结算金额 |
| **成功时间** | successTime | long | Unix timestamp |
| **拓展字段** | custom | string | 商户自定义参数原样返回 |

## 异步通知参数说明

当单据在平台已经处理完成后，系统自动调用合作商的通知接口(用户提供的notify\_url参数)。

固定签名排序：customerId={customerId}&sysOrderId={sysOrderId}&orderId={orderId}&status={status}&faceMoney={faceMoney}&realMoney={realMoney}&amount={amount}&successTime={successTime}&key={签名秘钥}

收到异步消息需输出success 以便系统确认消息送达。如没有正确响应系统会再回掉2次，每次回掉间隔递增10秒

|  |  |  |  |
| --- | --- | --- | --- |
| **中文名称** | **字段名称** | **类型** | **说明** |
| **商户号** | customerId | string | 商户号 |
| **订单号** | sysOrderId | string | 该订单在平台的寄售订单号 |
| **商户订单号** | orderId | string | 商户上传的订单号 |
| **订单状态** | status | string | 1处理中2成功3失败 |
| **备注** | message | string | 订单信息（不参与签名） |
| **卡面金额** | faceMoney | string | 商户上传卡面值金额 |
| **结算面值** | realMoney | float | 卡的真实面值 |
| **结算金额** | amount | int | 卡的结算金额 |
| **成功时间** | successTime | int | Unix timestamp |
| **拓展字段** | custom | string | 商户自定义参数原样返回（不参与签名） |
| **签名** | sign | string | 参照4.1.3(不参与签名) |