**TÀI LIỆU ĐẶC TẢ KẾT NỐI**

**API THANH TOÁN QUA THẺ CÀO QUA HỆ THỐNG SMS.VN**

Phiên bản 1.0

(Gửi đối tác)

Cập nhật nội dung

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Mô tả** | **Người cập nhật** |
| 28/03/2013 | 1.0 | Tạo tài liệu | TuTL |
| 19/02/2016 | 1.1 | Update | HaNM |

Để cung cấp cổng thanh toán qua thẻ cào cho các đối tác một cách đơn giản nhất, iNET đã cập nhật cổng kết nối Card Charging. Cổng kết nối này được dựng trên giao thức http, dữ liệu định dạng thuần văn bản (Plain Text)và có hai tùy chọn: xác thực hoặc không xác thực tính toàn vẹn dữ liệu

1. **API Kết nối**

Để sử dụng kênh thanh toán thẻ cào thông qua kết nối với sms.vn, đối tác cần xây dựng module request tới URL theo method **POST**

URL request:

http://sms.vn/card-charging-api

Method: POST

Request: Plain text

Response: Plain text

1. **Mô tả các hàm sử dụng**
   1. **Card Charging**

Thực hiện thanh toán cho sản phẩm

* + 1. **Request**

Ví dụ:

* Chuối tham số tùy chọn không xác thực tính toàn dữ liệu:

accesskey=6fe7a9a25c4e30e731c0bc1d18e9da28&serial=123456789&pin=12345678&type=VTT

* Chuỗi tham số tùy chọn xác thực tính toàn vẹn dữ liệu

accesskey=6fe7a9a25c4e30e731c0bc1d18e9da28&serial=123456789&pin=12345678&type=VTT&signature=8FA66D343C2586ACBA69B65D3621E8F496675585EAB1B4ADE71F51AEE2FA4DA563480D4B3F353125A8C83551AF650203010001

Chuỗi signature được tạo ra bằng cách mã hóa MD5 chuỗi các tham số**(4. Hướng dẫn kết nối)**

Các tham số:

|  |  |  |
| --- | --- | --- |
| **STT** | **Name** | **Description** |
| 1 | accesskey | Chuỗi mã hóa định danh sản phẩm & tài khoản |
| 2 | serial | Chuỗi serial thẻ cào |
| 3 | pin | Mã thẻ cào |
| 4 | type | Mã nhà cung cấp thẻ cào (xem bảng bên dưới) |
| 5 | signature | Chuỗi chữ ký để xác thực tính toàn vẹn dữ liệu. Bằng null nếu không yêu cầu xác thực |

* + 1. **Response**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã lỗi** | **Giải thích** | **Cách giải quyết** |
| 1 | -1985 | Invalid parameters request | Thiếu param request |
| 2 | -1984 | Wrong Username | Kiểm tra lại Username do iNET cung cấp |
| 3 | -1983 | Wrong Password | Kiểm tra lại Password do iNET cung cấp |
| 4 | -1982 | Invalid IP request | IP request chưa khai báo với iNet |
| 5 | -1981 | System busy | Re-try |
| 6 | 0 | Transaction fail | Giao dịch không thành công |
| 7 | 10000 | Tương ứng với giao dịch thành công và giá trị thẻ là 10.000 vnđ | Thành công |
| 8 | >10000 | Tương ứng với giao dịch thành công và giá trị thẻ là giá trị mã trả về (20000, 30000, 50000 …) | Thành công |
| 9 | -10 | Mã thẻ sai định dạng | Kiểm tra lại định dạng mã thẻ, không nhập vào các ký tự “-” trong phần mã thẻ(nếu có). |
| 10 | 4 | Thẻ không sử dụng được | Kiểm tra lại mã thẻ(mã lỗi với thẻ Vinaphone). |
| 11 | 5 | Nhập sai mã thẻ quá 5 lần. | Liên hệ với iNET để được trợ giúp |
| 12 | 6 | Thẻ chưa được kích hoạt | Liên hệ với nhà cung cấp thẻ |
| 13 | 9 | Tạm thời khóa kênh nạp VMS do quá tải. | Partner khóa các giao dịch nạp tiền và thực hiện nạp lại sau 2-3 phút. |
| 14 | 10 | Hệ thống nhà cung cấp gặp lỗi. | Partner khóa các giao dịch nạp tiền và thực hiện nạp lại sau 2-3 phút. |
| 15 | 11 | Kết nối với nhà cung cấp tạm thời bị gián đoạn | Liên hệ với iNET để được trợ giúp. |
| 16 | 13 | Hệ thống tạm thời bận | Hệ thống tạm thời bận, liên hệ với iNET để được trợ giúp. |
| 17 | -2 | Thẻ đã bị khóa | Kiểm tra lại mã thẻ với nhà cung cấp. |
| 18 | -3 | Thẻ hết hạn sử dụng | Kiểm tra lại hạn sử dụng của thẻ |
| 19 | 50 | Thẻ đã sử dụng hoặc không tồn tại | Kiểm tra lại mã thẻ đang sử dụng. |
| 20 | 51 | Seri thẻ không đúng | Kiểm tra lại serial cua thẻ đang sử dụng. |
| 21 | 52 | Mã thẻ và serial không khớp | Kiểm tra lại serial và mã thẻ |
| 22 | 53 | Serial hoặc mã thẻ không đúng | Kiểm tra lại serial và mã thẻ |
| 23 | 55 | Card tạm thời bị block 24 h. | Sau 24h thẻ sẽ đươc kích hoạt lại. |
| 24 | 59 | Mã thẻ chưa được kích hoạt. | Kiểm tra lại mã thẻ với nhà cung cấp(thẻ Viettel). |
| 25 | 99 | Giao dịch pending | Định kì 30 phút thực hiện request đến hệ thống iNET để lấy trạng thái giao dịch nếu error code vẫn trả về 99.  (Sử dụng hàm Lấy trạng thái giao dịch pending) |
| 26 | 100 | Không tồn tại access key |  |
| 27 | 101 | Sai chữ kỹ, dữ liệu không toàn vẹn |  |
| 28 | 102 | Tài khoản tạm thời bị khóa 15 phút vì nạp sai mã thẻ quá 5 lần lien tiếp. |  |
|  | **<Các giá trị khác>** |  | **Contact với đầu mối iNET để được trợ giúp** |

1. **Mã nhà cung cấp**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nhà cung cấp** | **Mã** | **Độ dài mã thẻ** | **Độ dài serial** |
| Vinaphone | VNP | 12-15 | 9 |
| MobiFone | VMS | 12-14 | 9-15 |
| VTC Vcoin | VTC | 12 | 12 |
| FPT Gate | FPT | 10 | 10 |
| Viettel | VTT | 13-15 | 11-15 |
| Megacard | MGC | 12 | 12 |
| Oncash | ONC | 12 | 10-12 |

1. **Hướng dẫn kết nối**
   1. **Quy tắc tạo chữ ký**

Chuỗi chữ ký được sinh ra bằng cách mã hóa MD5 từ chuỗi ghép các tham số theo tứ thự nêu dưới đây, cách nhau bằng dấu ‘&’, giữa các tham số và giá trị cách nhau bằng dấu ‘=’

Ví dụ:

accesskey=6fe7a9a25c4e30e731c0bc1d18e9da28&serial=123456789&pin=12345678&type=VTT&privatekey=7dbf0ee2739fa53d41b74424be99ec2c

Sau đó chuỗi tham số được mã hóa MD5 để sinh ra chữ ký.

signature = MD5(chuỗi\_tham\_số)

* 1. **Code mẫu JAVA**

**//Thư viện MD5**

**import** java.security.MessageDigest;

**import** java.security.NoSuchAlgorithmException;

**public** class MD5 {

public static String encrypt(String input) {

StringBuffer sb = new StringBuffer();

try {

MessageDigest md = MessageDigest.getInstance("MD5");

md.update(input.getBytes());

byte[] output = md.digest();

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

sb.append(Integer.toString((output[i] & 0xff) + 0x100, 16).substring(1));

}

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

}

return sb.toString();

}

}

**// Kết nối**

**import** java.io.BufferedReader;

**import** java.io.DataOutputStream;

**import** java.io.InputStreamReader;

**import** java.net.HttpURLConnection;

**import** java.net.URL;

**public class** HttpURLConnectionExample {

**public static void** main(String[] args) **throws** Exception {

HttpURLConnectionExample http = **new** HttpURLConnectionExample();

// Tao chu ky

String chuoiParam = accesskey=6fe7a9a25c4e30e731c0bc1d18e9da28&serial=123456789&pin=12345678&type=VTT&privatekey=7dbf0ee2739fa53d41b74424be99ec2c

String signature = MD5(chuoiParam);

http.sendPost(access\_key, type, pin, serial, signature);

}

// HTTP POST request

**privatevoid** sendPost(String access\_key, String type, String pin,

String serial, String signature) **throws** Exception {

String url = "http://sms.vn/card-charging-api

URL obj = **new** URL(url);

HttpURLConnection con = (HttpURLConnection) obj.openConnection();

// add reuqest header

con.setRequestMethod("POST");

String urlParameters = "accesskey="+access\_key+"&type="+type+"&pin="+pin+"&serial="+serial+"&signature="+signature;

// Send post request

con.setDoOutput(**true**);

DataOutputStream wr = **new** DataOutputStream(con.getOutputStream());

wr.writeBytes(urlParameters);

wr.flush();

wr.close();

**int** responseCode = con.getResponseCode();

System.*out*.println("\nSending 'POST' request to URL : " + url);

System.*out*.println("Post parameters : " + urlParameters);

System.*out*.println("Response Code : " + responseCode);

BufferedReader in = **new** BufferedReader(**new** InputStreamReader(con

.getInputStream()));

String inputLine;

StringBuffer response = **new** StringBuffer();

**while** ((inputLine = in.readLine()) != **null**) {

response.append(inputLine);

}

in.close();

// print result

System.*out*.println(response.toString());

}

}