

Tài liệu kết nối API Thẻ cào

Whypay, February, 2016
Authors: ThanhTl@whypay.vn

Hà Nội, 2016

Table of content

Quy trình kết nối	5
Gạch thẻ cào	5
Mô tả	5
Địa chỉ API	5
Bản tin Request	5
Bản tin Response	6
Ví dụ	6
Truy vấn giao dịch gạch thẻ	7
Mô tả	7
Địa chỉ API	7
Bản tin Request	7
Bản tin Response	7
Ví dụ	8
Mua mã thẻ cào	10
Mô tả	10
Địa chỉ API	10
Bản tin Request	10
Bản tin Response	11
Ví dụ	11
Truy vấn giao dịch mua thẻ	13
Mô tả	13
Địa chỉ API	13
Bản tin Request	13
Bản tin Response	13
Ví dụ	14
Topup	16
Mô tả	16
Địa chỉ API	16
Bản tin Request	16
Bản tin Response	16
Ví dụ	17
Truy vấn giao dịch Topup	18

Mô tả	18
Địa chỉ API	18
Bản tin Request	18
Bản tin Response	19
Ví dụ	19
Đặt hạn mức GD / ngày	21
Mô tả	21
Địa chỉ API	21
Bản tin Request	21
Bản tin Response	21
Ví dụ	21
Lấy số dư ví	23
Mô tả	23
Địa chỉ API	23
Bản tin Request	23
Bản tin Response	23
Ví dụ	23
Lưu ý:	24
Samples	25
PHP	25
C#	26
Java	28
Sandbox	30
Địa chỉ API	30
Thông tin đối tác	30
Ví dụ	30
Phụ lục	31
Bảng mapping mã thẻ	31
Bảng mapping mã telco	32
Bảng phân biệt các mã lỗi trả về	32

Lịch sử thay đổi

Ngày	Thay đổi	Nội dung	Ghi chú
1/2016	Tạo mới tài liệu	Tạo mới tài liệu	thanhtl
7/2017	Thêm Sample Java	Sample Java	thanhtl
8/2017	Lưu ý về trường hợp timeout	Lưu ý về trường hợp timeout	thanhtl
11/2017	Update địa chỉ sandbox	Add Sandbox	thanhtl
1/2018	Mua thẻ, truy vấn mua thẻ	Bổ sung mô tả phần giải mã trước khi parse json	thanhtl

1. Quy trình kết nối

Bước 1: Đối tác đăng ký tài khoản trên trang <https://santhecao.com>

Bước 2: Đối tác gửi thông tin tài khoản, địa chỉ ip cho Admin Santhecao

Bước 3: Admin Santhecao gửi lại cho đối tác thông tin ID và KEY dùng để kết nối

Bước 4: Đối tác thực hiện kết nối và nghiệm thu trên hệ thống

2. Gạch thẻ cào

2.1. Mô tả

Gạch thẻ cào là hàm cho phép đối tác chuyển thẻ cào thành tiền mặt. Tỷ lệ chuyển đổi quý đối tác tham khảo bản mới nhất tại trang web <https://santhecao.com>. Tài liệu này chỉ mô tả phương thức kết S2S

2.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

URL API: <http://gw.santhecao.com/api/user/sync>

2.3. Bản tin Request

KEY	VALUE	DESC
function	PaymentCard	Giá trị mặc định để gọi hàm PaymentCard
card_code	12454512454	Mã thẻ
card_serial	11545154451	Số serial
card_type	VTT	Loại thẻ: VTT (Viettel), VMS (Mobiphone), VNP (Vinaphone), FPT (Gate), VCOIN (VTC)
partner_rid	12456789abc	Request ID của đối tác gửi sang (để trace giao dịch).

2.4. Bản tin Response

KEY	VALUE	DESC
amount	10000	Số tiền
response_status	0	Mã lỗi trả về. Trường hợp response_status = null có nghĩa là gd Timeout, cần tra cứu lại GD
status	1	trạng thái giao dịch, 1 là thành công, <> 1 là các lỗi khác
desc	Giao dịch thành công	Mô tả lỗi
trans_id	12345	Mã giao dịch trên hệ thống santhecao

2.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => 'PaymentCard', 'card_code' => '1234564444',
'card_serial' => '1245154221', 'card_type' => 'VTT',
'partner_rid'=>'1234abc'];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```
$data = json_encode($data);
$data = Encrypt ($data, KEY);
$data = urlencode($data);
```

- Bước 3: Gọi đến trang API

```
$data = file_get_contents(
    "http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .
    $data);
```

- Bước 4: Lấy ra dữ liệu trả về

```
$data = json_decode($data, true);
```

3. Truy vấn giao dịch gạch thẻ

3.1. Mô tả

Truy vấn lại giao dịch gạch thẻ cào trước đó. Sử dụng trong trường hợp đối tác muốn kiểm tra lại giao dịch gạch thẻ cào đã thành công hay chưa?

3.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

URL API: <http://gw.santhecao.com/api/user/sync>

3.3. Bản tin Request

KEY	VALUE	DESC
function	CheckTransaction	Giá trị mặc định để gọi hàm CheckTransaction
trans_id	XXXX	Mã giao dịch của hệ thống santhecao
partner_rid	12456789abc	Request ID của đối tác gửi sang (Trường hợp ko gửi trans_id)

3.4. Bản tin Response

KEY	VALUE	DESC
amount	10000	Số tiền

response_status	0	Mã lỗi trả về. Trường hợp response_status = null có nghĩa là gd Timeout, cần tra cứu lại GD
status	1	trạng thái giao dịch, 1 là thành công, <> 1 là các lỗi khác
desc	Giao dịch thành công	Mô tả lỗi
trans_id	12345	Mã giao dịch trên hệ thống santhecao
ref		Ref từ kênh đối tác
partner_rid		Request ID của đối tác gửi sang

3.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => 'CheckTransaction', 'trans_id' => '12312',  
'partner_rid' => '12132'];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```
$data = json_encode($data);  
$data = Encrypt ($data, KEY);  
$data = urlencode($data);
```

- Bước 3: Gọi đến trang API

```
$data = file_get_contents(  
"http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .  
$data);
```

- Bước 4: Lấy ra dữ liệu trả về

```
$data = json_decode($data, true);
```

- Bước 5: Lưu ý trường hợp timeout: Khi gd bị Timeout (Không biết được kết quả cuối), mặc định hệ thống sẽ trả về trạng thái giao dịch thành công (status = 1), đồng thời trạng thái mã lỗi Gw là NULL (response_status = null). Trường hợp này đối tác cần kiểm tra kỹ và gọi lại hàm check giao dịch để lấy kết quả cuối cùng (thành công hay thất bại). Tổng hợp các trạng thái gd được mô tả bằng bảng như sau

status	response_status	Mô tả
1	Khác NULL	Thành công
1	NULL	Timeout, cần truy vấn
Khác 1	Khác NULL	Thất bại
Khác 1	NULL	Timeout, cần truy vấn

4. Mua mã thẻ cào

4.1. Mô tả

API mua thẻ cào cho phép đối tác mua mã thẻ cào các nhà mạng, hoặc thẻ game. Chiết khấu mua thẻ cào được quy định trong hợp đồng của đối tác, phạm vi tài liệu này chỉ đề cập đến việc kết nối API giữa 2 hệ thống.

4.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

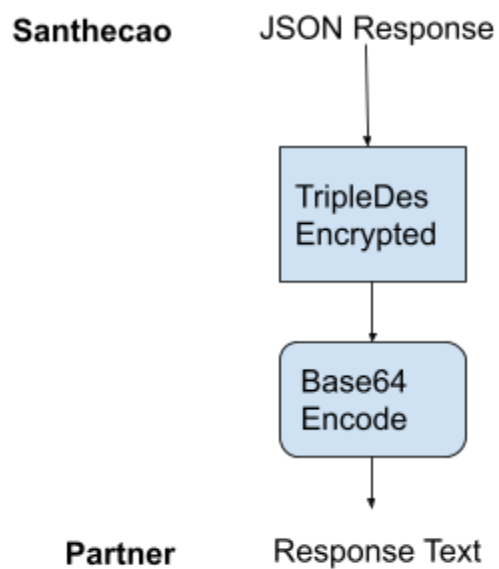
URL API: <http://gw.santhecao.com/api/user/sync>

4.3. Bản tin Request

KEY	VALUE	DESC
<code>function</code>	<code>BuyCardCode</code>	Giá trị mặc định để gọi hàm <code>BuyCardCode</code>
<code>telco</code>	VTT	Loại thẻ: VTT (Viettel), VMS: Mobifone, VNP: vinaphone (xem bảng mapping các nhà mạng ở dưới)
<code>amount</code>	20000	Mệnh giá thẻ. Hiện tại đang hỗ trợ các mệnh giá 10K, 20K, 50K, 100K, 200K, 500K
<code>quantity</code>	1	Số lượng thẻ
<code>partner rid</code>	12456789abc	Request ID của đối tác gửi sang (để trace giao dịch).

4.4. Bản tin Response

Không giống như các API khác. API này trả về dữ liệu text đã được mã hóa. Đối tác cần giải mã trước khi parse json



Giá trị JSON sau khi giải mã như bên dưới

KEY	VALUE	DESC
<code>status</code>	1	trạng thái giao dịch, 1 là thành công, <> 1 là các lỗi khác
<code>desc</code>	Thành công	Mô tả lỗi
<code>trans_id</code>	112323	Mã giao dịch trên hệ thống santhecao
<code>response_status</code>	1	Lỗi trả về từ GW mua thẻ (tham khảo). Trường hợp <code>response_status = null</code> có nghĩa là gd Timeout, cần tra cứu lại GD

<code>cards</code>	<pre>[{"code":"03XXXXX420661", "serial":"3XXXXX74", "expired_date":1502163767, "price":"20000", "type":"VTT"}]</pre>	Danh sách thẻ, trả bằng chuỗi json có cấu trúc như sau: Cards = [THE1, THE2, THE3 .., THEN] THE{1..n} = {code, serial, expired_date, price, type}
--------------------	--	--

4.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => 'BuyCardCode', 'telco'=> 'VTT', 'amount' =>
'20000', 'quantity' => '1', 'partner_rid'=>'1234abc'];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```
$data = json_encode($data);
$data = Encrypt ($data, KEY);
$data = urlencode($data);
```

- Bước 3: Gọi đến trang API

```
$data = file_get_contents(
"http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .
$data);
```

- Bước 4: Lấy ra dữ liệu trả về

```
$data = Decrypt ($data, KEY);
$data = json_decode($data, true);
```

- Bước 5: Lưu ý trường hợp timeout: Khi gd bị Timeout (Không biết được kết quả cuối), mặc định hệ thống sẽ trả về trạng thái giao dịch thành công (status = 1), đồng thời trạng thái mã lỗi Gw là NULL (response_status =

null). Trường hợp này đối tác cần kiểm tra kỹ và gọi lại hàm check giao dịch để lấy kết quả cuối cùng (thành công hay thất bại). Tổng hợp các trạng thái gd được mô tả bằng bảng như sau

status	response_status	Mô tả
1	Khác NULL	Thành công
1	NULL	Timeout, cần truy vấn
Khác 1	Khác NULL	Thất bại
Khác 1	NULL	Timeout, cần truy vấn

5. Truy vấn giao dịch mua thẻ

5.1. Mô tả

Truy vấn lại giao dịch mua thẻ cào trước đó. Sử dụng trong trường hợp đối tác muốn kiểm tra lại giao dịch mua thẻ cào đã thành công hay chưa, đồng thời có thể lấy được danh sách mã thẻ đã mua trước đó

5.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

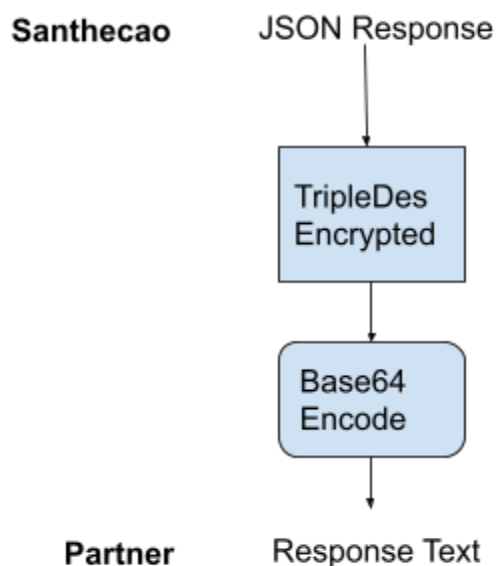
URL API: <http://gw.santhecao.com/api/user/sync>

5.3. Bản tin Request

KEY	VALUE	DESC
function	<code>CheckBuyCard</code>	Giá trị mặc định để gọi hàm <code>CheckBuyCard</code>
partner_rid	12456789abc	Request ID của đối tác gửi sang

5.4. Bản tin Response

Không giống như các API khác. API này trả về dữ liệu text đã được mã hóa. Đối tác cần giải mã trước khi parse json



Giá trị JSON sau khi giải mã như bên dưới

KEY	VALUE	DESC
<code>status</code>	1	trạng thái giao dịch, 1 là thành công, <> 1 là các lỗi khác
<code>desc</code>	Thành công	Mô tả lỗi
<code>trans_id</code>	112323	Mã giao dịch trên hệ thống santhecao
<code>response_status</code>	1	Lỗi trả về từ GW mua thẻ (tham khảo). Trường hợp response_status = null có nghĩa là gd Timeout, cần tra cứu lại GD
<code>cards</code>	[{"code": "03XXXXX420661", "serial": "3XXXXX74", "expire	Danh sách thẻ, trả bằng chuỗi json có cấu trúc như sau: Cards = [THE1, THE2, THE3 .., THEN]

	d_date":15021637 67,"price":"20000", "type":"VTT"]	THE{1..n} = {code,serial,expired_date,price,type} Expired_date: theo dạng unix timestamp
--	--	---

5.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => 'CheckBuyCard', 'trans_id' => '12312',  
        'partner_rid' => '12132'];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```
$data = json_encode($data);  
$data = Encrypt ($data, KEY);  
$data = urlencode($data);
```

- Bước 3: Gọi đến trang API

```
$data = file_get_contents(  
    "http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .  
    $data);
```

- Bước 4: Lấy ra dữ liệu trả về

```
$data = Decrypt ($data, KEY);  
$data = json_decode($data, true);
```

- Bước 5: Lưu ý trường hợp timeout: Khi gd bị Timeout (Không biết được kết quả cuối), mặc định hệ thống sẽ trả về trạng thái giao dịch thành công (status = 1), đồng thời trạng thái mã lỗi Gw là NULL (response_status = null). Trường hợp này đối tác cần kiểm tra kỹ và gọi lại hàm check giao

dịch để lấy kết quả cuối cùng (thành công hay thất bại). Tổng hợp các trạng thái gd được mô tả bằng bảng như sau

status	response_status	Mô tả
1	Khác NULL	Thành công
1	NULL	Timeout, cần truy vấn
Khác 1	Khác NULL	Thất bại
Khác 1	NULL	Timeout, cần truy vấn

6. Topup

6.1. Mô tả

API Topup cho phép đối tác thực hiện nạp tiền trực tiếp vào số điện thoại cho khách hàng. Chiết khấu topup được quy định trong hợp đồng của đối tác.

6.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

URL API: <http://gw.santhecao.com/api/user/sync>

6.3. Bản tin Request

KEY	VALUE	DESC
<code>function</code>	<code>Topup</code>	Giá trị mặc định để gọi hàm <code>Topup</code>
<code>msisdn</code>	0912345678	Số điện thoại theo định dạng 09, 016,
<code>mobile type</code>	1	1: Trả trước 0: Trả sau
<code>amount</code>	20000	Mệnh giá topup. (Hỗ trợ các mệnh giá mặc định sau, 10K, 20K, 50K, 100K, 200K, 500K)
<code>telco</code>	VTT	Nhà mạng - trong trường hợp chuyển mạng giữ số (VTT VMS VNP VNM GMOBILE) xem bảng mã mapping telco
<code>partner rid</code>	12456789abc	Request ID của đối tác gửi sang (để trace giao dịch).

6.4. Bản tin Response

KEY	VALUE	DESC
<code>status</code>	1	trạng thái giao dịch, 1 là thành công, <> 1 là các lỗi khác
<code>desc</code>	Thành công	Mô tả lỗi
<code>trans_id</code>	112323	Mã giao dịch trên hệ thống santhecao
<code>response_status</code>	1	Lỗi trả về từ GW mua thẻ (tham khảo). Trường hợp <code>response_status</code> = null có nghĩa là gd Timeout, cần tra cứu lại GD

6.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => Topup, msisdn=> 0977798600, 'amount' =>
'20000', mobile_type=> '1', 'partner_rid'=>'1234abc'];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```
$data = json_encode($data);
$data = Encrypt ($data, KEY);
$data = urlencode($data);
```

- Bước 3: Gọi đến trang API

```
$data = file_get_contents(
"http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .
$data);
```

- Bước 4: Lấy ra dữ liệu trả về

```
$data = json_decode($data, true);
```

- Bước 5: Lưu ý trường hợp timeout: Khi gd bị Timeout (Không biết được kết quả cuối), mặc định hệ thống sẽ trả về trạng thái giao dịch thành công (status = 1), đồng thời trạng thái mã lỗi Gw là NULL (response_status = null). Trường hợp này đối tác cần kiểm tra kỹ và gọi lại hàm check giao dịch để lấy kết quả cuối cùng (thành công hay thất bại). Tổng hợp các trạng thái gd được mô tả bằng bảng như sau

status	response_status	Mô tả
1	Khác NULL	Thành công
1	NULL	Timeout, cần truy vấn
Khác 1	Khác NULL	Thất bại
Khác 1	NULL	Timeout, cần truy vấn

7. Truy vấn giao dịch Topup

7.1. Mô tả

Truy vấn lại giao dịch Topup đã gọi trước đó. Sử dụng trong trường hợp đối tác muốn kiểm tra lại giao dịch Topup hoặc trường hợp Topup trả về Timeout

7.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

URL API: <http://gw.santhecao.com/api/user/sync>

7.3. Bản tin Request

KEY	VALUE	DESC
-----	-------	------

function	CheckTopup	Giá trị mặc định để gọi hàm CheckTopup
partner_rid	12456789abc	Request ID của đối tác gửi sang

7.4. Bản tin Response

KEY	VALUE	DESC
status	1	trạng thái giao dịch, 1 là thành công, <> 1 là các lỗi khác
desc	Thành công	Mô tả lỗi
trans_id	112323	Mã giao dịch trên hệ thống santhecao
response_status	1	Lỗi trả về từ GW mua thẻ (tham khảo). Nếu response_status có giá trị là null, nghĩa là gd bị timeout, cần truy vấn lại kết quả

7.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => CheckTopup, 'trans_id' => '12312',  
'partner_rid' => '12132'];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```
$data = json_encode($data);  
$data = Encrypt($data, KEY);  
$data = urlencode($data);
```

- Bước 3: Gọi đến trang API

```
$data = file_get_contents(
    "http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .
    $data);
```

- Bước 4: Lấy ra dữ liệu trả về

```
$data = json_decode($data, true);
```

- Bước 5: Lưu ý trường hợp timeout: Khi gd bị Timeout (Không biết được kết quả cuối), mặc định hệ thống sẽ trả về trạng thái giao dịch thành công (status = 1), đồng thời trạng thái mã lỗi Gw là NULL (response_status = null). Trường hợp này đối tác cần kiểm tra kỹ và gọi lại hàm check giao dịch để lấy kết quả cuối cùng (thành công hay thất bại). Tổng hợp các trạng thái gd được mô tả bằng bảng như sau

status	response_status	Mô tả
1	Khác NULL	Thành công
1	NULL	Timeout, cần truy vấn
Khác 1	Khác NULL	Thất bại
Khác 1	NULL	Timeout, cần truy vấn

8. Đặt hạn mức GD / ngày

8.1. Mô tả

Hàm này cho phép đối tác đặt được hạn mức giao dịch trên 1 ngày.

8.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

URL API: <http://gw.santhecao.com/api/user/sync>

8.3. Bản tin Request

KEY	VALUE	DESC
function	<code>SetQuota</code>	Giá trị mặc định để gọi hàm <code>SetQuota</code>
quota	500000	Hạn mức cần set. = 0 nếu như ko muốn giới hạn hạn mức Null hoặc < 0 nếu muốn trả về hạn mức hiện tại

8.4. Bản tin Response

Hạn mức hiện tại

8.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => SetQuota, quota=> '12312'];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```

$data = json_encode($data);

$data = Encrypt ($data, KEY);

$data = urlencode($data);

```

- Bước 3: Gọi đến trang API

```

$data = file_get_contents(

"http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .

$data);

```

- Bước 4: Lấy ra dữ liệu trả về

```

$data = json_decode($data, true);

```

9. Lấy số dư ví

9.1. Mô tả

Hàm này cho phép đối tác lấy số dư ví tại thời điểm gọi.

9.2. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

URL API: <http://gw.santhecao.com/api/user/sync>

9.3. Bản tin Request

KEY	VALUE	DESC
function	<code>GetBalance</code>	Giá trị mặc định để gọi hàm <code>GetBalance</code>

9.4. Bản tin Response

Số dư hiện tại

9.5. Ví dụ

- Bước 1: Đóng gói dữ liệu cần gửi đến gw

```
$data = ['function' => GetBalance];
```

- Bước 2: Mã hóa dữ liệu đã đóng gói

```
$data = json_encode($data);
```

```
$data = Encrypt($data, KEY);
```

```
$data = urlencode($data);
```

- Bước 3: Gọi đến trang API

```
$data = file_get_contents(  
    "http://gw.santhecao.com/api/user/sync?partner=PARTNER_ID&info=" .  
    $data);
```

- Bước 4: Lấy ra dữ liệu trả về

```
$data = json_decode($data, true);
```

Lưu ý:

10. Samples

10.1. PHP

```
<?php
function Encrypt($data, $secret) {
    //Generate a key from a hash
    $key = md5(utf8_encode($secret), true);

    //Take first 8 bytes of $key and append them to the end of $key.
    $key .= substr($key, 0, 8);

    //Pad for PKCS7
    $blockSize = mcrypt_get_block_size('tripledes', 'ecb');
    $len = strlen($data);
    $pad = $blockSize - ($len % $blockSize);
    $data .= str_repeat(chr($pad), $pad);

    //Encrypt data
    $encData = mcrypt_encrypt('tripledes', $key, $data, 'ecb');

    return base64_encode($encData);
}

//And here is the new decrypt function.

function Decrypt($data, $secret) {

    //Generate a key from a hash
    $key = md5(utf8_encode($secret), true);

    //Take first 8 bytes of $key and append them to the end of $key.
    $key .= substr($key, 0, 8);

    $data = base64_decode($data);

    $data = mcrypt_decrypt('tripledes', $key, $data, 'ecb');

    $block = mcrypt_get_block_size('tripledes', 'ecb');
    $len = strlen($data);
    $pad = ord($data[$len - 1]);

    return substr($data, 0, strlen($data) - $pad);
}
```

```
}
```

10.2. C#

```
private static string Encrypted(string data)
{
    //Generate a key from a hash
    MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider();
    // Create a Byte array to store the encryption to return.
    byte[] key;
    // Required UTF8 Encoding used to encode the input value to a usable
state.
    UTF8Encoding textencoder = new UTF8Encoding();

    // let the show begin.
    key = md5.ComputeHash(textencoder.GetBytes("SECRET_KEY"));

    // Destroy objects that aren't needed.
    md5.Clear();
    md5 = null;
    byte[] toEncryptArray = UTF8Encoding.UTF8.GetBytes(data);

    byte[] first8byte = new byte[8];
    for (int i = 0; i < 8; i++)
    {
        first8byte[i] = key[i];
    }
    //Take first 8 bytes of $key and append them to the end of $key.
    key = key.Concat(first8byte).ToArray();

    TripleDESCryptoServiceProvider tdes = new
TripleDESCryptoServiceProvider();
    //set the secret key for the tripleDES algorithm
    tdes.Key = key;
    //mode of operation. there are other 4 modes. We choose ECB(Electronic
code Book)
    tdes.Mode = CipherMode.ECB;
    //padding mode(if any extra byte added)
    tdes.Padding = PaddingMode.PKCS7;

    ICryptoTransform cTransform = tdes.CreateEncryptor();
    //transform the specified region of bytes array to resultArray
    byte[] resultArray = cTransform.TransformFinalBlock
    toEncryptArray, 0, toEncryptArray.Length;
}
```

```

        //Release resources held by TripleDes Encryptor
        tdes.Clear();
        //Return the encrypted data into unreadable string format
        return Convert.ToBase64String(resultArray, 0, resultArray.Length);
    }

private static string Decrypted(string data)
{
    //Generate a key from a hash

    byte[] key = KraGiE.TripleDES.HSSEncryption(SECRET_KEY);
    byte[] toDecryptArray = Convert.FromBase64String(data);

    byte[] first8byte = new byte[8];
    for (int i = 0; i < 8; i++)
    {
        first8byte[i] = key[i];
    }

    //Take first 8 bytes of $key and append them to the end of $key.
    key = key.Concat(first8byte.ToArray());

    TripleDESCryptoServiceProvider tdes = new TripleDESCryptoServiceProvider();
    //set the secret key for the tripleDES algorithm
    tdes.Key = key;
    //mode of operation. there are other 4 modes. We choose ECB(Electronic code
    Book)
    tdes.Mode = CipherMode.ECB;
    //padding mode(if any extra byte added)
    tdes.Padding = PaddingMode.PKCS7;

    ICryptoTransform cTransform = tdes.CreateDecryptor();
    //transform the specified region of bytes array to resultArray
    byte[] resultArray = cTransform.TransformFinalBlock
        (toDecryptArray, 0, toDecryptArray.Length);
    //Release resources held by TripleDes Encryptor
    tdes.Clear();
    //Return the decrypted data into unreadable string format
    return System.Text.Encoding.UTF8.GetString(resultArray);
}

```

10.3. Java

```

/**
 * Created by ThanhTL on 7/11/2017.
 */
import java.security.MessageDigest;
import java.util.Arrays;

import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

public class SanthecaoServices {

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

        String text = "thanhtl";

        String codedtext = encrypt(text, "xxxxxx");
        String decodedtext = decrypt(codedtext, "xxxxxx");

        System.out.println(codedtext); // this is a byte array, you'll just see a reference to an array
        System.out.println(decodedtext); // This correctly shows "kyle boon"
    }

    public static String encrypt(String message, String key) throws Exception {
        final MessageDigest md = MessageDigest.getInstance("md5");
        final byte[] digestOfPassword = md.digest(key.getBytes("utf-8"));
        final byte[] keyBytes = new byte[digestOfPassword.length + 8];
        for (int i = 0; i < keyBytes.length; i++) {
            int j = (i >= digestOfPassword.length ? i - digestOfPassword.length : i);
            keyBytes[i] = digestOfPassword[j];
        }

        final SecretKey skey = new SecretKeySpec(keyBytes, "DESede");
        final IvParameterSpec iv = new IvParameterSpec(new byte[8]);
        final Cipher cipher = Cipher.getInstance("DESede/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, skey);

        final byte[] plainTextBytes = message.getBytes("utf-8");
        final byte[] cipherText = cipher.doFinal(plainTextBytes);
        final String encodedCipherText = Base64.getEncoder().encodeToString(cipherText);
    }
}

```

```

        return encodedCipherText;
    }

    public static String decrypt(String base64Msg, String key) throws Exception {
        final MessageDigest md = MessageDigest.getInstance("md5");
        final byte[] digestOfPassword = md.digest(key.getBytes("utf-8"));
        final byte[] keyBytes = new byte[digestOfPassword.length + 8];
        for (int i = 0; i < keyBytes.length; i++) {
            int j = (i >= digestOfPassword.length ? i - digestOfPassword.length : i);
            keyBytes[i] = digestOfPassword[j];
        }

        final SecretKey skey = new SecretKeySpec(keyBytes, "DESede");
        final IvParameterSpec iv = new IvParameterSpec(new byte[8]);
        final Cipher decipher = Cipher.getInstance("DESede/ECB/PKCS5Padding");
        decipher.init(Cipher.DECRYPT_MODE, skey);

        // final byte[] encData = new
        // sun.misc.BASE64Decoder().decodeBuffer(message);
        final byte[] plainText = decipher.doFinal(Base64.getDecoder().decode(base64Msg));

        return new String(plainText, "UTF-8");
    }
}

```

10.4. Nodejs

```

var CryptoJS = require('crypto-js');
var forge    = require('node-forge');
var utf8     = require('utf8');

/**
 * TripleDES encryption
 */
* @method encrypt
* @param {String} key
* @param {String} text
*/
function encrypt (key, text){

    key    = CryptoJS.MD5(utf8.encode(key)).toString(CryptoJS.enc.Latin1);

```

```

    key    = key + key.substring(0, 8);
    var cipher = forge.cipher.createCipher('3DES-ECB', forge.util.createBuffer(key));
    cipher.start({iv:''});
    cipher.update(forge.util.createBuffer(text, 'utf-8'));
    cipher.finish();
    var encrypted = cipher.output;
    return ( forge.util.encode64(encrypted.getBytes()) );
}

/**
 * TripleDES decryption
 */
* @method decrypt
* @param {String} key
* @param {String} encrypted_text (should be a base64 encoded string)
*/
function decrypt(key, encrypted_text){

    key    = CryptoJS.MD5(utf8.encode(key)).toString(CryptoJS.enc.Latin1);
    key    = key + key.substring(0, 8);
    var decipher = forge.cipher.createDecipher('3DES-ECB', forge.util.createBuffer(key));
    encrypted_text = forge.util.decode64(encrypted_text);
    decipher.start({iv:''});
    decipher.update(forge.util.createBuffer(encrypted_text, 'utf-8'));
    decipher.finish();
    var decrypted = decipher.output;
    return decrypted.toString();
}

const key = "t6twy5-S11_75=5qy42ER6Y84r84494p";

console.log(encrypt(key,
JSON.stringify({"function":"Topup","msisdn":"84977798600","mobile_type":"1","amount":10000,"partner_id":"test5a1d1834c4ff0"})));

console.log(decrypt(key,"CXEvO1n" +
"KxZR14/43HVoz4VeXFe/oZOPpuFylrKzx6xlyUeBXWGNsSeZEq7Oc3qgAc1oLyXeCzNq/4/mTyD9eMLICfTkYt5
72JtJGhSyXrhacKPKIqynewt3nkeK" +
"mldgC12rxyH7blzI4WtjaCTdng=="))

```

Python 3.x

```

from Crypto.Hash import MD5
from Crypto.Cipher import DES3
import base64
import requests
import json
from urllib.parse import quote

def encrypt(data, secret):
    md5 = MD5.new()
    md5.update(secret.encode("utf-8"))
    key = md5.digest()
    # print("Md5 is", md5, md5, len(md5), "\n", md5[0:8])
    # //Take first 8 bytes of $key and append them to the end of $key.
    key += key[0:8]
    # key = key.decode('ascii')

    cipher = DES3.new(key, DES3.MODE_ECB)

    pad_len = 8 - len(data) % 8 # length of padding
    padding = chr(pad_len) * pad_len # PKCS5 padding content
    data += padding

    print("length data", len(data), data)
    msg = cipher.encrypt(data)
    encoded = base64.b64encode(msg)
    # print(encoded)
    return encoded

def decrypt(data, secret):
    md5 = MD5.new()
    md5.update(secret.encode("utf-8"))
    key = md5.digest()
    # print("Md5 is", md5, md5, len(md5), "\n", md5[0:8])
    # //Take first 8 bytes of $key and append them to the end of $key.
    key += key[0:8]
    # key = key.decode('ascii')

    cipher = DES3.new(key, DES3.MODE_ECB)

    msg = cipher.decrypt(base64.b64decode(data))
    # print(encoded)
    return msg

# stc-endpoint
URL = "http://gw.santhecao.com/api/user/sync"
PARTNER_ID = 999 #stc cung cap cho doi tac
PARTNER_KEY = 'KEY'# stc cung cap cho doi tac

# location given here
data = {"function": "CheckTransaction", "partner_id": "1231231"}

info = json.dumps(data)

```

```

# print(info)

info = encrypt(info,PARTNER_KEY)
# print(info)

# print("decode", decrypt(info,PARTNER_KEY))

# defining a params dict for the parameters to be sent to the API
params = {'partner': PARTNER_ID,'info': info}

r = requests.get(url = URL, params = params)

# data = r.json()

print(r.json())

```

11. Sandbox

11.1. Địa chỉ API

Đối tác thực hiện kết nối vào địa chỉ sau

URL API: <http://sandbox.santhecao.com:8090/api/user/sync>

11.2. Thông tin đối tác

PARTNER_ID	999
SECKEY	t6twy5-S11_75=5qy42ER6Y84r84494p

11.3. Ví dụ

- Bước 1: Tạo JSON string

```

{"function":"Topup","msisdn":"0977798600","mobile_type":"1","amount":10000,"partner_rid":"test5a1d1834c4ff0"}

```

- Bước 2: Mã hóa dữ liệu

```
CXEvO1n
KxZR14/43HVoz4VeXFe/oZOPpuFylrKzx6xlyUeBXWGNsSeZEq7Oc3qgAc1oLy
XeCzNq/4/mTyD9eMLlCfTkhYT572JtJGhSyXrhacKPKIqynewt3nkeK
mIdgC12rxyH7blzl4WtjaCTdng==
```

- Bước 3: URL_ENCODE dữ liệu đã được mã hóa

```
CXEvO1nKxZR14%2F43HVoz4VeXFe%2FoZOPpuFylrKzx6xlyUeBXWGNsSeZEq7
Oc3qgAc1oLyXeCzNq%2F4%2FmTyD9eMLlCfTkhYT572JtJGhSyXrhac
KPKIqynewt3nkeKmIdgC12rxyH7blzl4WtjaCTdng%3D%3D
```

- Bước 4: Gọi đến URL sandbox và đợi kết quả trả về

```
http://sandbox.santhecao.com:8090/api/user/sync?partner=999&in
fo=CXEvO1nKxZR14%2F43HVoz4VeXFe%2FoZOPpuFylrKzx6xlyUeBX
WGNsSeZEq7Oc3qgAc1oLyXeCzNq%2F4%2FmTyD9eMLlCfTkhYT572JtJGhSyXr
hacKPKIqynewt3nkeKmIdgC12rxyH7blzl4WtjaCTdng%3D%3D
```

- Bước 5: Xử lý dữ liệu trả về

```
{"status":1,"desc":"Giao dịch thành công (gia
lap).","response_status":1,"trans_id":10004}
```

12. Phụ lục

12.1. Bảng mapping mã thẻ

Mã	Giải thích
VTT	Thẻ Viettel
VMS	Thẻ Mobi
VNP	Thẻ Vinaphone
VNM	Thẻ mạng Vietnam mobile
GMOBILE	Thẻ mạng GTel, Gmobile
ZING	Thẻ Zing, của vina game
GATE	Thẻ Gate, FPT
ONCASH	Thẻ ONCASH, VDC-Net2E
GARENA	Thẻ GARENA
VCOIN	Thẻ VCOIN của VTC
MGC	Thẻ Mega Card
VMSDATA	Thẻ data Mobifone

12.2. Bảng mapping mã telco

Mã	Giải thích
VTT	Viettel
VMS	Mobi
VNP	Vinaphone
VNM	Vietnam mobile
GMOBILE	GTel, Gmobile

12.3. Bảng phân biệt các mã lỗi trả về

status	response_status	Mô tả
1	Khác NULL	Thành công
1	NULL	Timeout, cần truy vấn
Khác 1	Khác NULL	Thất bại
Khác 1	NULL	Timeout, cần truy vấn

12.4. Bảng mã lỗi

Stt	Lỗi hệ thống	Mã lỗi
1	Hết thẻ	status = 99 và response_status = 95
2	Tài khoản hết tiền	status = 90 và response_status = 98
3	Quá hạn mức GD	status = 90 và response_status = 90
4	Không đủ điều kiện thực hiện GD	status = 90 và response_status = 91
5	Số lượng / Mệnh giá / Loại thẻ không hợp lệ	(status = 90 hoặc status = 99) và response_status = 97
6	Lỗi hệ thống	status = 99 và response_status = 500
7	Lỗi cổng thẻ	status = 0