TÀI LIỆU ĐẶC TẢ
API KẾT NỐI ĐẾN HỆ THỐNG CTG

MỤC LỤC

MERCHANT kết nối vào hệ thống được cung cấp các tham số:

STT	Tham số	Mô tả
1	PartnerID	ID cho MERCHANT do CTG cung cấp
2	MPIN	MPIN cho MERCHANT CTG cung cấp
3	User	User để truy cập hệ thống do CTG cung cấp
4	password	Mật khẩu người dùng truy cập hệ thống

1. Thuật toán mã hóa sử dụng

- Thuật toán TripleDES với Cipher Mode là ECB và Padding là PKCS5/PKCS
- Thuật toán mã hóa SHA

2. Các bước thực hiện kết nối

Thực hiện lệnh nghiệp vụ:

- Sau khi login thành công hê thống CTG sẽ trả về thông báo và kèm theo một session ID, client sẽ sử dụng session ID đó để mã hóa và 1 số các thông tin gửi đi.
- Nếu quá một thời gian cho phép của session (exprite) thì client cần phải đăng nhập lại bằng lệnh login
- Không nên Login logout nhiều lần để tăng performance cho hệ thống. Khi gửi lệnh CardCharge CTG sẽ trả về thông báo nếu như hết hạn session, lúc đó client phải login lại.
- Logout khỏi hệ thống: Client gửi lệnh logout đến CTG. Sau thời điểm này, session ID của client sẽ không còn hiệu lực.
- Ngoài ra bên phía MERCHANT được cung cấp thêm 2 hàm:
    - ChangePass : để đổi password đăng nhập hệ thống
    - ChangeMpin : để đổi Mpin gọi hàm charge

3. Mô tả các hàm giao tiếp

Description	Details	Remark
Server IP	IP	Địa chỉ IP của máy chủ chạy ứng dụng.
Server Port	TCP Port	Cổng mở cho phép truy cập
Protocol	WSDL	
Server URL	http://123.30.146.197:8080/webservice/TelcoAPI?wsdl	Đường dẫn giao tiếp

4. Đặc tả các hàm

Đăng nhập hệ thống

Tên hàm: login

Mục đích đăng nhập hệ thống để lấy kiểm tra và lấy mã phiên làm việc (sessionid)

Parameter	Description	Data Type
Username	Tên truy nhập cung cấp	Kí tự
Password	Mật khẩu của người sử dụng được mã hóa theo thuật toán SHA (mật khẩu do CTG cung cấp, bên MERCHANT có thể thay đổi)	Kí tự
ParnerID	ID của MERCHANT được khai báo cho user người sử dụng.	alphanumberic

Thứ tự tham số của hàm như sau:

Public LoginResponse logIn(String username, String SHApass, int partnerID)

Kết quả trả về

Parameter	Description	Data Type
Status	1: Truy nhập không thành công Khác 1: không thành công	alphanumberic
Sesion_id	Mã truy cập do hệ thống trả về để người dùng sử dụng	Kí tự
Message	Thông báo trả về tự hệ thống	Kí tự

Thoát hệ thống

Tên hàm: logout

Mục đích hủy mã phiên giao dịch đảm bảo độ bảo mật

Parameter	Description	Data Type
Username	Tên truy nhập cung cấp	Kí tự
Partner ID	Id của merchant được cung cấp	Kí tự
Sesion_id	Mã truy cập mà đã được hệ thống trả về khi truy nhập thành công.	Kí tự

Kết quả trả về

Parameter	Description	Data Type
Status	1: thành công	alphanumberic
Message	Câu môt tả	alphanumberic
TransId	Mã giao dịch	alphanumberic

Thứ tự tham số:

public LogoutResponse logOut(String username,int partnerID, String Sesion_id)

Đổi mật khẩu người sử dụng

Tên hàm: change_password

Thực hiện thay đổi mật khẩu của người sử dụng.

Parameter	Description	Data Type
Username	Tên truy nhập được cung cấp	Kí tự
ParnerID	ID của merchant	Kí tự
OldPassword	Mật khẩu cũ của người sử dụng đã được mã hóa bằng thuật toán SHA	Kí tự
NewPassword	Mật khẩu mới của người sử dụng đã được mã hóa bằng thuật toán SHA	Kí tự

Thứ tự tham số

public ChangeResponse ChangePassword(String username, int partnerID, String oldPass, String newPass)

Kết quả trả về

Parameter	Description	Data Type
Status	1 :Thành công	alphanumberic
Message	Câu môt tả	alphanumberic
TransId	Mã giao dịch	alphanumberic

Đổi MPIN

Tên hàm: change_mpin

Thực hiện thay đổi Mpin của người sử dụng.

Parameter	Description	Data Type
Username	Tên truy nhập được cung cấp	Kí tự
AgentID	ID của MERCHANT	Kí tự
OldMPIN	Mật khẩu của MERCHANT đã được mã hóa bằng thuật toán tripledes với key là sessionid mà khi đăng nhập hệ thống trả	Kí tự
NewMPIN	Mật khẩu đã được mã hóa bằng thuật toán tripledes với key là sessionid mà khi đăng nhập hệ thống trả về.	Kí tự

Thứ tự tham số

public ChangeResponse ChangeMPIN(String username, int partnerID, String oldMPIN, String newMPIN)

Kết quả trả về

Parameter	Description	Data Type
Status	1 :Thành công	alphanumberic
Message	Câu môt tả	alphanumberic
TransId	Mã giao dịch	

Charging Function:

Tên hàm: CardCharge

Thực hiện các giao dịch nạp tiền cho khách hàng từ thẻ trả trước VMS/VNP

Parameter	Description	Data Type
Username	Tên truy nhập đã được cung cấp	Kí tự
ParnerID	ID cung cấp cho MERCHANT	alphanumberic

MPIN	Mpin của MERCHANT đã được mã hóa bằng thuật toán tripledes với key là sessionid mà khi đăng nhập hệ thống trả về.	Kí tự
Target	Tài khoản của user nhận (chuỗi username của khách hàng bên merchant)	Kí tự

dataCard	Thông tin bí mật trên thẻ, thông tin được mã hoá bằng thuật toán TribleDess với KEY là SessionID. Thông tin trước khi mã hoá: mã thẻ:mã nhà cung cấp. Ví dụ: Mã thẻ: 13 Mã nhà cung cấp: VNP Dữ liệu trước khi mã hoá: 13:VNP Sau khi mã hoá mảng byte được chuyển về định dạng chuỗi hexa. Trong đấy danh sách mã nhà cung cấp do CTG gửi cho đối tác. Trong trường hợp cần gửi serial thẻ lên hệ thống, carddata có định dạng như sau: Seri của thẻ: mã thẻ: mệnh giá: mã nhà cung cấp. Ví dụ: Seri: 12 Mã thẻ: 13 Mệnh giá: 10000 Mã nhà cung cấp: VNP Carddata=12:13:1000:VNP Trường hợp không gửi mệnh giá lên, thì thay bằng ký tự trống: Ex: 12:13::VNP	Kí tự
Target_email	Là kiểu String email đúng định dạng mà bên phía merchant truyền sang cho hệ thống CTG	String

Target_mobile	Là kiểu String nhưng ở dạng chuỗi số mà bên phía merchant truyền sang cho hệ thống CTG	String

Thứ tự tham số:

public ChargeResponse CardCharge(String username,long partnerID,String mpin_encrypt, String encr_dataCard,String target, String Target_email, String Target_mobile)

Kết quả trả về

Parameter	Description	Data Type
Status	1 :Thành công	alphanumberic
Trans_id	Mã giao dịch	Numberic
amount	Số tiền nạp thành công. Đã được mã hóa, merchant cần giải mã bằng session_key đã cung cấp	alphanumberic
Message	Câu thông báo	alphanumberic

Bảng mã trả về khi thực hiện lệnh:

Giá trị	Thông báo lỗi
1	Lệnh thực hiện thành công
-1	Thẻ đã sử dụng (mã dành cho thẻ mobifone)
-2	Thẻ đã khoá (mã dành cho thẻ mobifone)
-3	Thẻ đã hết hạn sử dụng (mã dành cho thẻ mobifone)
-4	Thẻ chưa được kích hoạt (mã dành cho thẻ mobifone)
-10	Mã thẻ không đúng định dạng (mã dành cho thẻ mobifone)
-12	Thẻ không tồn tại (mã dành cho thẻ mobifone)
-14	Không đúng định dạng (Mô tả trả về tương ứng)
-99	Lỗi hệ thống nạp bên mobifone (mã dành cho thẻ mobifone
0	Lỗi khác (mã dành cho thẻ mobifone)
2	Không login sử dụng các hàm charge

3	Lỗi hệ thống
4	Thẻ không sử dụng được (lỗi chung cho thẻ vinaphone)
5	Thực hiện lệnh sai 10 lần liên tiếp.
6	Thẻ mới được sử dụng trước đó
7	Lỗi hệ thống VMS quá tải tạm thời dừng kênh nạp thẻ VMS
8	Charge thẻ bị lỗi hệ thống lỗi này cần ghi nhận lại để kiểm soát và đối soát (lỗi này ít xảy ra nhưng sẽ để đối soát xem
9	Sai thông tin partner
10	Sai format thông tin email , mobile gửi
11	Đang khóa Kênh VMS, VNP hoặc VTT
15	Thẻ đã được sử dụng
50	Card is used or card do not exist (Thẻ đã sử dụng hoặc không tồn tại)
51	Card serial is invalid (Seri thẻ không đúng)
52	Card serial and pin is not match (Mã thẻ và serial không khớp)
53	Card serial or pin is incorrects (Serial hoặc mã thẻ không đúng)
54	Card is waiting for activate (Card chưa được kích hoạt, liên hệ với nhà cung cấp)
55	Card is block for 24 hours (Các tạm thời bị block 24 h)
59	Card is not activate (Mã thẻ chưa được kích hoạt)
56	TargetAccount is locked (TargetAccount tạm thời bị khóa do Charging sai nhiều lần)
601	Card war used (Thẻ đã được sử dụng)
602	Card isn’t exist (Thẻ không tồn tại)
603	Card serial incorrect (Mã thẻ không đúng định dạng)
607	Lỗi hệ thống telco
608	Loss of connection to the core system
610	Thông tin giao dịch không chính xác

655	Vượt qua số gd cho phép
657	The partner was locked
658	Timeout
659	The database of Scratch card system is error
661	scratch catch payment is too failure to allow
697	Database of core is error
698	Loss of connection to the core system
699	Error is not in the list description

5. Bảng mã nhà cung cấp.

Nhà cung cấp	Mã nhà cung cấp	Độ dài mã thẻ	Độ dài serial	Cần phải gửi serial thẻ lên Server
Vinaphone	VNP	12		Có
MobiFone	VMS	14		Có
Viettel	VTT	13-15	11-15	Có

6. Các biện pháp bảo mật và chống gian lận.

- Thực hiện truy nhập hệ thống bằng username và password đã được cung cấp để lấy mã phiên làm việc.
- Tại tất cả các giao dịch thì mật khẩu được gửi đi bằng cách mã hóa bằng giải thuật tripleDes với key là mã phiên làm việc do hệ thống trả về lúc đăng nhập.
- Giới hạn số giao dịch trong khoảng thời gian:
    - Đảm bảo chống việc viết chương trình quét tìm mật khẩu đúng
    - Chống DDOS (tràn tải hệ thống).
    - Khóa user và đại lý nếu quá số lần đăng nhập sai cho phép.
- Yêu cầu bên phía Merchant phải có biện pháp giới hạn số lần nạp sai thẻ để tránh quét và tấn công DDOS.
    - Ví dụ: chỉ cho 1 user được nạp thẻ sai 5 lần liên tục trong 1 ngày.

7. Một số ví dụ về mã hoá và giải mã áp dụng

NET C#

```
      public string Encrypt(string key, string data)
        {
            data = data.Trim();
            byte[] keydata = Encoding.ASCII.GetBytes(key);
            string md5String = BitConverter.ToString(new
            MD5CryptoServiceProvider().ComputeHash(keydata)).Replace("-",
"").ToLower();
            byte[] tripleDesKey = Encoding.ASCII.GetBytes(md5String.Substring(0,
24));
            TripleDES tripdes = TripleDESCryptoServiceProvider.Create();
            tripdes.Mode = CipherMode.ECB; tripdes.Key = tripleDesKey;
tripdes.GenerateIV();
            MemoryStream ms = new MemoryStream();
            CryptoStream encStream = new CryptoStream(ms,
tripdes.CreateEncryptor(),
            CryptoStreamMode.Write);
encStream.Write(Encoding.ASCII.GetBytes(data), 0,
Encoding.ASCII.GetByteCount(data));
            encStream.FlushFinalBlock(); byte[] cryptoByte = ms.ToArray();
ms.Close();
            encStream.Close();
            return Convert.ToBase64String(cryptoByte, 0,
cryptoByte.GetLength(0)).Trim();
        }
        public static string Decrypt(string key, string dataen)
        {
            byte[] keydata = Encoding.ASCII.GetBytes(key);
            string md5String = BitConverter.ToString(new

            MD5CryptoServiceProvider().ComputeHash(keydata)).Replace("-",
"").ToLower();
            byte[] tripleDesKey =
            Encoding.ASCII.GetBytes(md5String.Substring(0, 24));
            TripleDES tripdes = TripleDESCryptoServiceProvider.Create();
            tripdes.Mode = CipherMode.ECB; tripdes.Key = tripleDesKey;
tripdes.GenerateIV();
            ICryptoTransform ict = tripdes.CreateDecryptor();
            byte[] cryptoByte =
            ict.TransformFinalBlock(Encoding.ASCII.GetBytes(dataen), 0, 8);

            string data = Convert.ToBase64String(cryptoByte, 0,
cryptoByte.GetLength(0)).Trim();
            return
            Encoding.ASCII.GetString(Convert.FromBase64String(data));
           }
```

Java

```
public static String getMD5(String sMessage) {
  byte[] defaultBytes = sMessage.getBytes();
  try {
      MessageDigest algorithm =
MessageDigest.getInstance("MD5");
      algorithm.reset();
      algorithm.update(defaultBytes);
```

```
      byte messageDigest[] = algorithm.digest();

      StringBuffer hexString = new StringBuffer();
      for (int i = 0; i < messageDigest.length; i++)
        String hex = Integer.toHexString(0xFF &
messageDigest[i]);
        if (hex.length() == 1) {
          hexString.append('0');
        }
        hexString.append(hex);
      }
      return hexString.toString();
    } catch (NoSuchAlgorithmException nsae) {
      return null;
    }
  }


public static String  Encrypt(String key,String data)  throws
Exception
{
  Cipher cipher=Cipher.getInstance("TripleDES");
String keymd5 =getMD5(key).substring(0,24);
      SecretKeySpec keyspec = new
      SecretKeySpec(keymd5.getBytes(),"TripleDES");
       cipher.init(Cipher.ENCRYPT_MODE,keyspec);
        byte[] stringBytes=data.getBytes();
        byte[] raw=cipher.doFinal(stringBytes);
        BASE64Encoder encoder = new BASE64Encoder();
        String base64 = encoder.encode(raw);
        return base64;
}

public static String  Decrypt(String key,String data)  throws
Exception
{
 Cipher cipher=Cipher.getInstance("TripleDES");
 String keymd5 =getMD5(key).substring(0,24);

       SecretKeySpec keyspec = new

 SecretKeySpec(keymd5.getBytes(),"TripleDES");

       cipher.init(Cipher.DECRYPT_MODE,keyspec);

   BASE64Decoder decoder = new BASE64Decoder();

   byte[] raw = decoder.decodeBuffer(data);

   byte[] stringBytes = cipher.doFinal(raw);

   String result = new String(stringBytes);

   return result;


   }
```

PHP

```
<?php

function Encrypt($input, $key_seed){

    $input = trim($input);

   $block = mcrypt_get_block_size('tripledes', 'ecb');

    $len = strlen($input);

    $padding = $block - ($len % $block);

    $input .= str_repeat(chr($padding),$padding);

    // generate a 24 byte key from the md5 of the seed

  $key = substr(md5($key_seed),0,24);

  $iv_size = mcrypt_get_iv_size(MCRYPT_TRIPLEDES,
MCRYPT_MODE_ECB);

  $iv = mcrypt_create_iv($iv_size, MCRYPT_RAND);

  // encrypt

  $encrypted_data = mcrypt_encrypt(MCRYPT_TRIPLEDES, $key,
$input,

MCRYPT_MODE_ECB, $iv);

  // clean up output and return base64 encoded

  return base64_encode($encrypted_data);

} //end function Encrypt()

?>

<?php function Decrypt($input, $key_seed)

{

      $input = base64_decode($input);

      $key = substr(md5($key_seed),0,24);

      $text=mcrypt_decrypt(MCRYPT_TRIPLEDES, $key, $input,
      MCRYPT_MODE_ECB,'12345678');
      $block = mcrypt_get_block_size('tripledes', 'ecb');

      $packing = ord($text{strlen($text) - 1});

if($packing and ($packing < $block)){

      for($P = strlen($text) - 1; $P >= strlen($text) - $packing; $P--){

       if(ord($text{$P}) != $packing){

 $packing = 0;
```

```
}

}

}

$text = substr($text,0,strlen($text) - $packing);

return $text;

}

?>
```