S88PAY PAYMENT INTEGRATION DOCUMENTATION V 2.3

CONTENTS

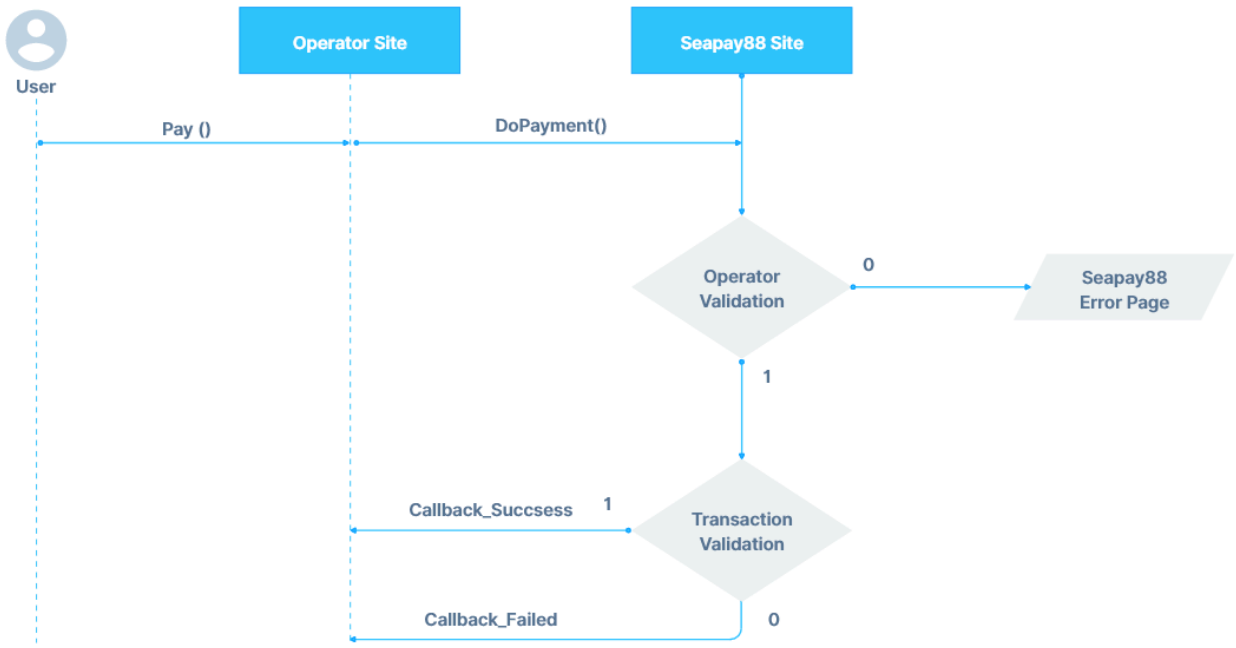
1. INTRODUCTION

This document provides technical specs for our Operator to perform integration and integration tests with S88pay Payment Gateway. This documentation includes functions, parameters, results, and error responses.

2. TERMS & DEFINITION

Name	Description
Operator	Partners who want to integrate with S88pay payment gateway
Payment Provider	Provider API Payment for Operator
BO Site	Backoffice site, used for Operator to verify transaction
Member Site	A website for member to log in and create transaction through S88Pay Payment Gateway
User/Member	Customer who create transaction through S88Pay Payment Gateway

3. WORKFLOW



4. API FEATURES

4.1 Pre Requirement Provide by Provider

- Server info & API URL
- Credential api : api key, secret key, merchant code
- Credential BO : username , password

4.2 Pre Requirement Provide by Operator :

- API Callback URL (mandatory). Used to forward transaction status, shown at point 7
- Call back success url page (optional), shown at point 8
- Call back failed url page (optional), shown at point 8

4.3 Api Security

- Token and transaction key (Need to create token and key for each transaction)
- IP White list for Back Office login

4. API FUNCTIONS

4.1 Payment Request

To be able to send the payment request to S88pay, the operator has to create a payment URL that contains the token and parameter request:

{base_url}/{merchant_code}/v2/dopayment?key={key}

- base_url : Provided by provider
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	merchant_code	Y	Provided by provider
2	merchant_api_key	Y	Provided by provider
3	transaction_code	Y	Generated by the operator, use string data type. It must be unique for each transaction
4	transaction_timestamp	Y	Generated by the operator. Integer data type. This parameter describes the transaction request time. The more detailed information regarding timestamps, please visit https://www.epochconverter.com/. Please note that we only process the time stamp on these time limits: **min: 1 hour before now and max: 5 minutes after now**

5	transaction_amount	Y	Using double data type
6	payment_code	Y	Use string value, for example, *P001*. Please contact the administrator to get your payment code.
7	user_id	Y	• Use string value. • For JPY, enter a user name (English or Japanese).
8	currency_code	Y	Please refer to the merchant's main currency list (point 13)
9	bank_code	N	Required just on VND, THB, IDR, MYR, MMK and PHP online bank payment method (points 15, 16)
10	callback_url	N	url callback beside url set from BO
11	return_url	N	Dynamic return URL to redirect back to the merchant page after completing transactions. **Use URL encoding when passing redirect URLs.**
12	random_bank_code	N	Only for VND (Use value *BankQR* for E-Wallet and *OBT* for bank transfer)
13	identity_id	N	required for Turkey (TRY)
14	phone	N	Required for: • MMK (WAVEPAY only: fill with Wavepay Account Number)
15	depositor_name	N	Mandatory for KRW
16	depositor_account_number	Y	Mandatory for KRW and THB Bank Transfer

17	depositor_bank	Y	Mandatory for THB Bank Transfer. List of depositor_bank on Section 17. THB Bank Code List.
18	card_serial	N	Only for VND. This value is required for Card PC. Must be provided in numeric format.
19	card_pin	N	Only for VND. This value is required for Card PC. Must be provided in numeric format.

Combine all of the parameters above into one string, separated by "&" character, and then encrypt it using encrypt_decrypt algorithms.

For example :

$str = merchant_code=xxx&merchant_api_key=xxx&transaction_code=xxx&transaction_timestamp=xxx&payment_code=xxx&transaction_amount=xxx&user_id=xxx&currency_code=xxx

Encryption sample :

$key = encrypt_decrypt('encrypt',$str, '{your_api_key}', '{your_secret_key}')

Refer to point 5 for the encryption/decryption function explanation.

Request example:

https://{base_url}/{Merchant_Code}/v2/dopayment?key=3eX%2Bf%2BMoVECXxSkKqV7aBRYIbyWg3DxdPdgZyG%2B377a7dR1OBBDNnU%2B%2Fvtn7hUyjP7WWdZ7gCsPF0J%2BJOiSxb1BFueIyRX3rxbSMa%2B%2FAyFvhz4L%2F2wJmSJKcNQn4whIL1sc1cfj7E1smQFAiYjfLXdY1Ev6Pnoit8Vouex3%2BupnZjJS8t44XRx5wugB5GuybZWPtlPhiN%2FP7P4uJW3RlFlo%2BtYrnHQ6GwqwRkoLrdv3qZXUzaatT8EWdztr973KWFDof2rVD%2B56SMAVrRHQZcYICU8RcjpyvJUaCtXpOKKg%3D

After the URL payment is created, then redirect to the URL.

4.2 Transaction status

This API is used to check the transaction status. It requires 1 post parameter (key), which is included by merchant_api_key and transaction_code.

method: POST

url : *{base_url}/api/v1/{merchant_code}/transactions/status*

parameter (form-data/x-www-form-urlencoded) :

a) key

The token contains these parameters :

No	Parameters	Description
1	merchant_api_key	Provided by provider
2	transaction_code	String data type.The transaction code.

Combine all of the parameters above into one string, separated by “&” character, and encrypt it using encrypt_decrypt algorithms.

For Example :

merchant_api_key=xxxxxxxxby9tUGdNbXI4U2o2Vml3dz09&transaction_code=DP0001

Encryption Example :

encrypt_decrypt('encrypt', 'merchant_api_key=xxxxxxxxby9tUGdNbXI4U2o2Vml3dz09&transaction_code=DP0001', '{your_api_key}', '{your_secret_key}')

Response Api Transaction status Example :


```
{
  "success": true,
  "data": [
    {
      "transaction_no": "DP1678780607892",
      "transaction_code": "TEST-DP-1678780607",
      "transaction_calculated_amount": "485.00000000",
      "transaction_actual_amount": "485.00000000",
      "amount": "500.00000000",
      "transaction_fee": "15.00000000",
      "transaction_ref": null,
      "transaction_ref2": null,
      "status": "pending",
      "type": "deposit",
      "note": null,
      "method_name": "UPI 3",
      "currency_name": "Indian Rupee",
      "currency_code": "INR",
      "datetime": "2023-03-14 15:56:47"
    }
  ],
  "message": "request transaction list successful"
}
```


4.3 Payout Request

This API is used for payout request submission. It requires some parameters to submit payouts. Check the transaction status. It requires an encrypted request with a "secret key" and "merchant key"

Method : POST

url : {baseurl}/api/v1/payout/{merchant_code}

Type : json

Json :

```
{
  "merchant_code" : "required",
   "transaction_code" : "required",
    "transaction_timestamp" : "required",
    "payout_code" : "required",
     "transaction_amount" : "required|numeric",
     "user_id" : "required",
     "currency_code" : "required",
     "bank_account_number" : "required",
      "ifsc_code": "required",
      "bank_code" : "required",
       "bank_name" : "required",
       "account_name": "required"
}
```

Decrypted request Example :

{"merchant_code":"SKU20220406065553","transaction_code":"WD00008","transaction_timestamp":"1652339606","payout_code":"WC01","transaction_amount":"100","user_id":"1","currency_code":"CNY","address":"gdgrgrgrewgewgewgrewg","account_name":"jhon"}

Encrypted Request Example :

{"key":"aHO52zIzSNR6wSQhSlS%2B9CSrEPWlEdsfKOL5YuWGSdPLny3Rgeo%2BKhglyFL2wgbOzG56iVWVeqMDqTZZSa0UIo9rKMtNJzY5zVnlRfBQaONMy5LVNyAmJEgtXxOmO30Za8PZLWcMGBUPk4fq95831U9iu6mh6DkwWk%2Blbn2aTUzqlj6LUA%2BlX8FIh2k%2BZaKvHCZw4ls0wfmwUQ9sr%2FjtgTwh1d69EDa96Ko7euFP1xjmxSFXD0mlCp%2B0%2FFNU0CBs3ZO1THnKjkbPI%2BCwkyI7qMUrIHGxBluE2laeryWUEaTHgaD1MsRd%2FFGLwHh%2F9pkHqlKoBtpoKOE%2BYW9sB2cWoJcMYCsBKRYFLKvbteutQBSZBaxY%2B0gqW99xIY6vqUHu"}

Success Response Sample :

```
{
   "success": true,
   "message": "request withdraw successful"
}
```

Failed Response Sample :


```
{
  "success": false,
  "message": "Invalid Transaction Code ( not unique )!, error code 111"
}
```


Payout Parameters :

No	Parameters Name	Required	Description
1	merchant_code	Y	Provided by provider
2	transaction_code	Y	Use string data type. It must be unique for each transaction
3	transaction_timestamp	Y	Generated by the operator. Integer data type. This parameter describes the transaction request time. The more detailed information regarding timestamps, please visit https://www.epochconverter.com/. Please note that we only process the time stamp on these time limits: min: 1 hour before now and max: 5 minutes after now
4	payout_code	Y	Use string value, for example, *W001*. Please contact the administrator to get your payout code.
5	transaction_amount	Y	Using double data type
6	user_id	Y	Use string values.
7	currency_code	Y	Use string value, Please refer to merchant's main currency list (point 14)
9	address	Y (For Crypto Payout)	This is the user's crypto wallet address.

			Use string type.
10	bank_account_number	Y (For INR, CNY, VND, THB, KRW, BDT, JPY,BRL, MMK Payout)	• Use string values. • For BRL used the ID CARD with 11 digits.
11	ifsc_code	Y (For INR Payout)	Use string values.
12	bank_code	Y (Not required for INR)	Use string values. Available bank codes for each currency are displayed at the bank code list section on this document.
13	bank_name	Y (Not required for INR)	Use string values. • Available bank names for each currency are displayed at the bank code list section on this document. • For BRL use the account value or PIX value.
14	branch_code	N	Required only for JPY (Using string data type).
15	account_name	Y	Bank Account / Wallet name. Using string data type.
16	callback_url	N	url callback beside url set from BO
17	phone_number	N	Required for BDT, 11 digits numbers without country code (ex. 01812345678)
18	account_type	N	Required for: • JPY (1 for saving; 2 for checking)
19	branch_name	N	Required for JPY

4.4 Get Balance

This API is used to check the merchant balance amount. It requires 1 post parameter (key), which is included by merchant_code

Method: GET

GET {base_url}/api/v1/balance/{merchant_code}

a) key

The token contains these parameters :

No	Parameters	Description
1	merchant_code	String data type.The merchant code.

Parameter json :

{"key":"encrypted merchant_code"}

sample

GET https://s88pay.net/api/v1/balance/SKU211111111

parameter json :

{"key":"ev0FFVerPMdbrFcXYLIwkyuL%2FY1v7%2F3GlKH46MJlNSI%3D"}

Sample response :


```
{
    "currency_name": "India Rupee",
    "currency_code": "INR",
    "balance": "100000",
    "frozen_balance": "1000",
     "available_balance": "99000"
}
```


4.5 Get Active Channels for VND

This API is used to check the available channels on VND. It requires 1 post parameter (key), which is included by merchant_code.

Method: POST

POST {base_url}/api/v2/channel/{merchant_code}

The token contains these parameters :

No	Parameters	Description
1	merchant_code	String data type.The merchant code.
2	currency_code	Use string value, Please refer to merchant's main currency list (point 14)

Parameter JSON :

key contain these encrypted parameters

merchant_code = string

currency_code = string

for example

merchant_code=SKU123456789&currency_code=vnd

Encrypted key

m75LzwzdUcTTZ8naXXqWa6BbM24kV9%2Fm3tMSO0gqMoH%2FR1KqwRVNKXzhFy9SLYi2efqZRcBBNhj38RRLUK%2BDrw%3D%3D

Sample :

POST https://s88pay.net/api/v2/channel/SKU123456789

Body

{
"key": "m75LzwzdUcTTZ8naXXqWa6BbM24kV9%2Fm3tMSO0gqMoH%2FR1KqwRVNKXzhFy9SLYi2efqZRcBBNhj38RRLUK%2BDrw%3D%3D"
}

Sample Response :


```
{
  "status": "success",
  "message": "success",
  "data": [
    {
      "code": "momo",
      "name": "Momo"
    },
    {
      "code": "vietcombank",
      "name": "VietcomBank"
    },
    {
      "code": "acbbank",
      "name": "ACBBank"
    }
  ],
  "error": null
}
```


5. ENCRYPT / DECRYPT FUNCTION

We combine some encryption methods to create the token.

The Sample function in PHP code :

```php
public function encrypt_decrypt($action, $string, $apikey =
'{your_api_key}', $secretkey = '{your_secret_key}') {
    $output = false;
    $encrypt_method = "AES-256-CBC";
    $secret_key = $apikey;
```

```
    $secret_iv = $secretkey;
        // hash
    $key = substr(hash('sha256', $secret_key, true), 0, 32);

    $iv = substr(hash('sha256', $secret_iv), 0, 16);
    if ( $action == 'encrypt' ) {
        $output = openssl_encrypt($string, $encrypt_method, $key,
OPENSSL_RAW_DATA, $iv);
        $output = base64_encode($output);
        $output = urlencode($output);
    } else if( $action == 'decrypt' ) {
     $output = openssl_decrypt(base64_decode(urldecode($string)),
$encrypt_method, $key, OPENSSL_RAW_DATA, $iv);
 }
 return $output;
}
```

The Sample function in C# code :

```
<?php

using System;
using System.Security.Cryptography;
using System.Text;
using System.IO;
public class EncryptDecrypt
{
    public static void Main()
    {
        string source =
"currency_code=INR&merchant_code=213213&merchant_api_key=3213212ewfewqfw
qf&transaction_code=TEST-DP-163158903432&transaction_timestamp=163158901
2&bank_code=1003&transaction_amount=1000";
        string apiKey =  "3213212ewfewqfwqf";
        string secretKey =  "jdo383f1d2021ehd1dj2di32";
        string result = encrypt_decrypt("encrypt", source, apiKey,
secretKey);
        Console.WriteLine("The key of  " + source + " is: " + result);
```

```
        Console.WriteLine("Decoded of key " + result + " is: " +
encrypt_decrypt("decrypt", result, apiKey, secretKey));
    }

    public static string encrypt_decrypt(string action, string payload,
string apikey, string secretkey){
        string secret_iv = secretkey;

        string iv = hashSha256(secret_iv).Substring(0, 16);

        SHA256 mySHA256 = SHA256Managed.Create();

        string output = "";
        if ( action == "encrypt" ) {
            output = EncryptString(payload,
mySHA256.ComputeHash(Encoding.ASCII.GetBytes(apikey)),
Encoding.ASCII.GetBytes(iv));
            } else if( action == "decrypt" ) {
                output = DecryptString(payload,
mySHA256.ComputeHash(Encoding.ASCII.GetBytes(apikey)),
Encoding.ASCII.GetBytes(iv));
            }
            return output;
        }

        public static string hashSha256(string payload){
            string hash = "";
            using (SHA256 sha256Hash = SHA256.Create()){
                byte[] sourceBytes = Encoding.UTF8.GetBytes(payload);
                byte[] hashBytes = sha256Hash.ComputeHash(sourceBytes);
                hash = BitConverter.ToString(hashBytes).Replace("-",
String.Empty).ToLower();
            }
            return hash;
        }

        public static string EncryptString(string plainText, byte[] key,
byte[] iv)
        {
```

```
            Aes encryptor = Aes.Create();
            encryptor.Mode = CipherMode.CBC;
            encryptor.Key = key;
            encryptor.IV = iv;
            MemoryStream memoryStream = new MemoryStream();
            ICryptoTransform aesEncryptor = encryptor.CreateEncryptor();
            CryptoStream cryptoStream = new CryptoStream(memoryStream,
aesEncryptor, CryptoStreamMode . Write);
            byte[] plainBytes = Encoding.ASCII.GetBytes(plainText);
            cryptoStream.Write(plainBytes, 0, plainBytes . Length);
            cryptoStream.FlushFinalBlock();
            byte[] cipherBytes = memoryStream.ToArray();
            memoryStream.Close();
            cryptoStream.Close();
            string cipherText = Convert.ToBase64String(cipherBytes, 0,
cipherBytes.Length);
            return Uri.EscapeDataString(cipherText);
        }

        public static string DecryptString(string cipherText, byte[]
key, byte[] iv)
        {
            Aes encryptor = Aes.Create();
            encryptor.Mode = CipherMode.CBC;
            encryptor.Key = key;
            encryptor.IV = iv;
            MemoryStream memoryStream = new MemoryStream();
            ICryptoTransform aesDecryptor = encryptor.CreateDecryptor();
            CryptoStream cryptoStream = new CryptoStream(memoryStream,
aesDecryptor, CryptoStreamMode . Write);
            string plainText = String.Empty;
            try {
                byte[] cipherBytes =
Convert.FromBase64String(Uri.UnescapeDataString(cipherText));
                cryptoStream.Write(cipherBytes, 0, cipherBytes .
Length);
                cryptoStream.FlushFinalBlock();
                byte[] plainBytes = memoryStream.ToArray();
                plainText = Encoding.ASCII.GetString(plainBytes, 0,
```

```
plainBytes.Length);
            } finally {
                memoryStream.Close();
                cryptoStream.Close();
            }
            return plainText;
        }
    }
```

Java Encrypt Steps

```
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.util.Arrays;
import java.util.Base64;
import java.util.Base64.Decoder;
import java.net.URLDecoder;
import java.net.URLEncoder;
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.security.NoSuchAlgorithmException;
import java.math.BigInteger;

public class Playground {
    public static void main(String[] args) throws Exception  {
        String key_api =  "FjgHGLxnJO1qjlhHcK4KHg%3D%3D";
        String secret_api = "vqw330bxVfX9TEGMF1BDUKlVDm4DFejA0STb0WDarkU%3D";

        //For Encrypt Sample Data
        String data =
"currency_code=THB&merchant_code=SKU20220913051108&merchant_api_key=FjgHGLxnJO1qjlhH
cK4KHg%3D%3D&transaction_code=D996574709141&transaction_timestamp=1663729322&payment
_code=PTHB07&bank_code=104&transaction_amount=500.00&user_id=4";

        //For Descrypt Sample Data
```

```
        String encryptData =
"6mWOao2m34hP%2FZHQ%2F5skq5kTDuzWkHePH74pJsc4FTgmXqhidXtY4VLq9eW0N7%2FWDfY4gH2gHEVw8
FU5d0LKpqc6mOc1gy8dhxyeT%2FlAyzaI0gnkXzRWwMm7pXlwNpoNG2v7%2FSXMCioAJMf3nLWEQji%2ByRT
dBxietyMP8XArnQ2KJ%2F7IoOwQgxl6rPPGuHHCoi74POTI5lEzHbHCNDgGc4XjF89BQM0ROtlAyTpG%2BST
CtEYjE90crt%2FGaZwYkwCcaqgtFo8o%2FBNRZTDs4WcQOQCTyu2AnHtxdJvcPUgXAW%2BebPv91vNrwt68A
mxcLhic";

        //Hash Key 32 Length
        final MessageDigest md_key = MessageDigest.getInstance("SHA-256");
        byte[] key_hash =
Arrays.copyOfRange(md_key.digest(key_api.getBytes("UTF-8")),0,32);

        //Hash Secret 16 Length
        String sha_secret = hashsha256(secret_api).substring(0,16);
        byte[] iv_hash = sha_secret.getBytes("UTF-8");

        //Execute Test Scenario
        // String result = EncryptDecrypt("encrypt",data,key_hash,iv_hash);
        String result = EncryptDecrypt("decrypt",encryptData,key_hash,iv_hash);
        System.out.println(result);

    }

    static String EncryptDecrypt(String action, String data, byte[] key_hash, byte[]
iv_hash) throws Exception {
        //Process OpenSSL EncryptDecrypt
        byte[] cipherText;
        String output;
        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
        final SecretKeySpec key = new SecretKeySpec(key_hash, "AES");
        final IvParameterSpec iv = new IvParameterSpec(iv_hash, 0,
cipher.getBlockSize());

        if(action == "encrypt"){
            //Encrypt condition
            cipher.init(Cipher.ENCRYPT_MODE, key, iv);
            cipherText = cipher.doFinal(data.getBytes("UTF-8"));
            output = Base64.getEncoder().encodeToString(cipherText);
            String result = URLEncoder.encode(output,
StandardCharsets.UTF_8.toString());
            return result;
        }else{
```

```
            //Decrypt condition
            String decodeX = URLDecoder.decode(data,
StandardCharsets.UTF_8.toString());
            byte[] decodeData = Base64.getDecoder().decode(decodeX);
            cipher.init(Cipher.DECRYPT_MODE, key, iv);
            cipherText = cipher.doFinal(decodeData);
            output = new String(cipherText,StandardCharsets.UTF_8);
            return output;
        }

    }

    static String hashsha256(String key) throws NoSuchAlgorithmException {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        md.update(key.getBytes(StandardCharsets.UTF_8));
        byte[] digest = md.digest();
        String key_string = String.format("%064x", new BigInteger(1, digest));
        return key_string;
    }
}
```

NodeJS Code:

```
var crypto = require('crypto')


const apikey = "myapikey"

const secretkey = "mysecretkey"


const encrypt_decrypt = (action, data) => {

   const encryptionMethod = "AES-256-CBC"


   const key = crypto.createHash('sha256').update(apikey).digest()

   const iv = crypto.createHash('sha256').update(secretkey,
'utf8').digest('hex').substring(0,16)
```

```
    if (action == "encrypt") {
        const cipher = crypto.createCipheriv(encryptionMethod, key, iv)
        const res = encodeURIComponent(Buffer.from(
            cipher.update(data, 'utf8', 'base64') + cipher.final('base64')
        ).toString())
        console.log(res)
        return res
    } else if (action == "decrypt") {
        const buff = decodeURIComponent(Buffer.from(data))
        const decipher = crypto.createDecipheriv(encryptionMethod, key, iv)
        const res = (
          decipher.update(buff.toString('utf8'), 'base64', 'utf8') +
          decipher.final('utf8')
        )
        console.log(res)
        return res
    }
}


// example to encrypt data
encrypt_decrypt("encrypt",
"currency_code=INR&merchant_code=SKU20230101012023&merchant_api_key=myapikey&transacti
on_code=TEST-DP-123&transaction_timestamp=1677495605&payment_code=PAY01D&transaction_a
mount=1000&user_id=test01")


// example to decrypt data
encrypt_decrypt("decrypt",
"QqqF5QD9NdtM1O5JCpySZXyFT0gmXvEgWUEgoW19xajeviLAAlwzdDJmD7sgE2laIp7iEt%2F1SzUpquHykjf
QP2eTTQGyR3Jw60iVniAayGxBOQRoPW91n%2FT4DzQkZL1eqapgcum%2FyGKLErYJ0v1WedA2nYZ%2Fd64vZIS
Gh3eA2PqDGJdLZWYKbAP7uGHzMGBslmx8CcBCFbjrKvfA5VGam6LHi1ZWTfv8eeHmlBv4CSI6pXzhb43UZ22uB
Qj%2FN8rc6oJQd7l14FK2A4sZhpUhZQ%3D%3D")
```

6. DEPOSIT CALL BACK API (Provide by Operator , Required)

After the operator sends a payment request on dopayment (see point 5.1), the provider will process it. Once the provider gets the transaction status (success or failed), the provider will call this API to forward the transaction status.

- Callback Payment API Url: can be set on BO
- Method : POST
- Parameter : key, transaction_code, transaction_no

Provider will send the parameter on key format, then operator need to decrypt it using encrypt/decrypt function as shown at point 6. For key, you need to decrypt with *apikey* and *apisecret*. Besides the key, we also send the transaction code in the transaction_code parameter and the transaction number in the transaction_no parameter that is not encrypted.

After decrypting the parameter you will find that the parameters are:

No	Parameter	Description
1	transaction_code	String data type. The transaction code that is sent by the operator on dopayment.
2	transaction_status	Integer data type. The status of the transaction as shown on point 9
3	transaction_amount	Double Data type. The amount of the transaction
4	transaction_fee	Double type. The amount of fee transaction
5	currency_code	String Data Type.. Please refer to currency list (point 11)
6	transaction_no	String data type. The transaction code on the provider's database
7	transaction_actual_amount	Double Data type. The actual amount paid by the member.

Note:

- For the deposit data, Parameters are separated with "&" :

merchant_code=xxx&merchant_api_key=xxx&transaction_code=xxx&transaction_timestamp=xxx&payment_code=xxx&transaction_amount=xxx&user_id=xxx&currency_code=xxx

7. PAYOUT CALL BACK API (Provide by Operator , Required)

After the operator sends payout requests on point 5.3, then the provider will process it. Once the provider gets the transaction status (success or failed), the provider will call this API to forward the transaction status.

- Callback Payout API Url : can be set on BO
- Method : POST
- Parameter : key, transaction_code, transaction_no

Provider will send the parameter on key format, then operator need to decrypt it using encrypt/decrypt function as shown at point 6. For key, you need to decrypt with *apikey* and *apisecret.* Besides the key, we also send the transaction code in the transaction_code parameter and the transaction number in the transaction_no parameter that is not encrypted.

After decrypting the parameter, you will find that the parameters are:

No	Parameter	Description
1	transaction_no	String data type. The transaction code recorded by the provider
2	transaction_code	String data type. The transaction code sent by the operator on payout request
3	transaction_status	Integer data type. The status of the transaction as shown on point 10
4	transaction_amount	Double Data type. The amount of the transaction
5	transaction_fee	Double type. The amount of fee transaction
6	currency_code	Integer value. Please refer to currency code list (point 13)
7	transaction_ref	Transaction reference of payout request

Note:

- For the payout, the data params format are Json type:

{"merchant_code":"SKU20220406065553","transaction_code":"WD00008","transaction_timestamp":"16523 39606","payout_code":"WC01","transaction_amount":"100","user_id":"1","currency_code":"CNY","address":" gdgrgrgrewgewgewgrewg","account_name":"jhon"}

8. CALL BACK PAGE URL (Optional)

- Call back success page and failed URL: can be set on BO
- Method : Get

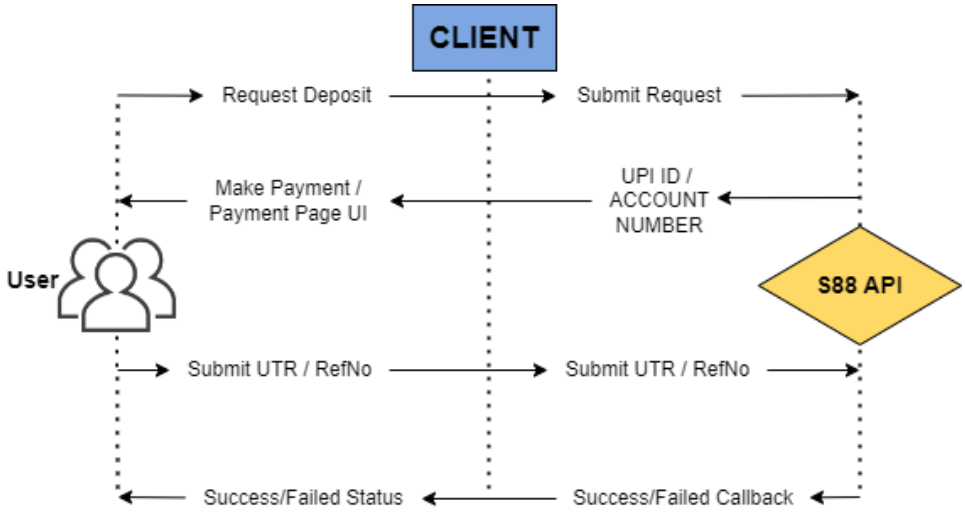
It is possible to set a success callback page. When the transaction gets a success status, the page will be redirected to the success URL page. When the transaction gets failed status, the page will be redirected to the failed URL page.

Sample callback success url format set on the Back Office: "http://www.test.com/success"

Then it will be redirected to this form: http://www.test.com/success

9. Payment Request v3 (for INR, BDT, VND, JPY, KRW, BRL, TRY, MMK, THB, KHR, MXN, MYR merchant only)

Besides payment request v2, s88pay has payment request v3 with API POST method. The payment request v3 flowchart:



9.1 Transaction Request API

{base_url}/api/{merchant_code}/v3/dopayment ***(for INR, BDT, VND, JPY, BRL, IDR, MMK, THB, KHR, MXN, MYR)***

{base_url}/api/{merchant_code}/v3/krw-payment ***(for KRW only)***

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	merchant_code	Y	Provided by provider
2	merchant_api_key	Y	Provided by provider
3	transaction_code	Y	Generated by operator, use string data type. Must be unique for each transactions
4	transaction_timestamp	Y	Generated by the operator. Integer data type. This parameter describes the transaction request time. The more detailed information regarding timestamps, please visit : https://www.epochconverter.com/. Please note that we only process the time stamp on these time limit: min: 1 hour before now and max: 5 minutes after now
5	transaction_amount	Y	Using double data type

6	payment_code	Y	Use string value for example *P001*. Please contact the administrator to get your payment code.
7	user_id	Y	• Use string value. • For JPY, enter a user name (English or Japanese).
8	currency_code	Y	Please refer to merchant main currency list (point 13)
9	bank_code	N	Required just on BDT, VND, THB, IDR, MYR, MMK, PHP online bank payment method (points 15, 16)
10	depositor_name	N	Mandatory for JPY, KRW and BDT Bank Transfer
11	depositor_account_number	Y	Mandatory for KRW and THB Bank Transfer
12	callback_url	N	url callback beside url set from BO
13	identity_id	Y	required for Turkey (TRY)
14	phone	N	Required for: • MMK (WAVEPAY only: fill with Wavepay Account Number)
15	card_serial	N	Only for VND. This value is required for Card PC. Must be provided in numeric format.
16	card_pin	N	Only for VND. This value is required for Card PC. Must be provided in numeric format.

combine all of the parameters above into one string, separate by “&” character and then encrypt it using encrypt_decrypt algorithms.

For example :

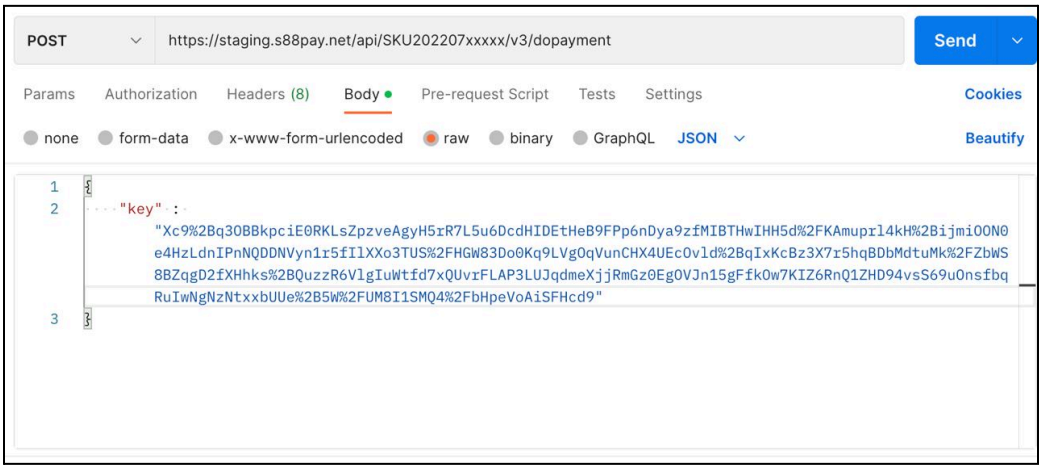
$str = merchant_code=xxx&merchant_api_key=xxx&transaction_code=xxx&transaction_timestamp=xxx&payment_code=xxx&transaction_amount=xxx&user_id=xxx&currency_code=xxx

Encryption sample :

$key = encrypt_decrypt(‘encrypt’,$str, ‘{your_api_key}’, ‘{your_secret_key}’)

Refer to point 6 for the encryption/decryption function explanation.

Request example :



Response example for INR:

{

"status": "success",

"message": "Submit Transaction Success!",

"transaction_no": "DP16873387xxxxx",

"transaction_code": "TEST-DP-16873xxxxx",

"amount": "510.00",

"upi_id": "example@upi"

"qr":"https://s88pay.net/imageencoded=true&url=aHR0cHM6Ly9hcGkucGF5cHJvc3R1ZG

vLmNvbS9hc3NldC91cGkvY2FjaGVkLzgzMmE1NmZlLTBkYzktNDc0Ni05NDhkLWFlYmY

3E5NDlkMw==

"expired_at":"2023-06-22 15:06:04"

"expired_timezone":"GMT+05:30"

}

Response example for BDT:

{

"status": "success",

"message": "Submit Transaction Success!",

"transaction_no": "DP16873387xxxxx",

"transaction_code": "TEST-DP-16873xxxxx",

"amount": "510.00",

"wallet_number": "01861843585"

"wallet_number_type": "Agent"

"wallet_name": "rocket"

"wallet_code": "1001"

"expired_at":"2023-06-22 15:06:04"

"expired_timezone":"GMT+05:30"

}

Response example for VND:

{

"status": "success",

"message": "Submit Transaction Success!",

"transaction_no": "DP16873387xxxxx",

"transaction_code": "TEST-DP-16873xxxxx",

"amount": "510.00",

"wallet_number":"0797748156",

"wallet_name":"Pham Chi Cuong",

"wallet_code":"momo",

"match_code":"685532",

"expired_at":"2023-06-22 15:06:04"

"expired_timezone":"GMT+05:30"

}

Response example for JPY:

{

"status": "success",

"message": "[DP] Submit Transaction Success!",

"transaction_no": "DP17xxxx25175326",

"amount": "1111111.00",

"bank_code": "0310",

"branch": "Natsxxxe",

"branch_code": "559",

"virtual_account": "964xx39",

"bank_name": "GMxxxらネット銀行",

"bank_account_name": "合同会社xxxASE",

"expired_at": "2025-06-19 18:41:15",

"expired_timezone": "GMT+09:00"

}

Response example for KRW:

{

"status": "success",

"message": "[DP] Submit Transaction Success!",

"transaction_no": "DP17549xxxxxxx8",

"transaction_code": "TEST-DP-1xxxxx74765",

"amount": "100000.00",

"name": "Juna",

"bank_name": "케이뱅크 K-Bank",

"bank_account_number": "2023030xxxxxxx",

"bank_account_name": "TEST"

}

Response example for IDR:

{

"status":"success",

"message":"Submit Transaction Success!",

"transaction_no":"DP1710917153xxx",

"transaction_code":"TEST-DP-1710917xxx",

"amount":"100003.00",

"qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29tL3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY29sb3I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJBTksuV1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYyNDAzMDNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzEzNDAzMDNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM1NCVTYwMTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwNDI2NTcwNTUwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X092MGFmYTRMZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",

"bank_name":"QRIS",

"account_number":"",

"expired_at":"2024-03-20 14:00:53",

"expired_timezone":"GMT+07:00"

}

Response example for MMK:

{

"status": "success",

"message": "[DP] Submit Transaction Success!",

"transaction_no": "DP1722324938xxx",

"transaction_code": "TEST-DP-172232xxxx",

"amount": "1000.00",

"bank_name": "AYA Bank",

"account_number": "test",

"name": "test",

"reference": "OS-0065785",

"expired_at": "2024-07-30 14:20:38",

"expired_timezone": "GMT+06:30"

}

Response example for THB:

b2b:


```
{
 "status": "success",
 "message": "[DP] Submit Transaction Success!",
 "transaction_no": "DP17xxxx251xxx2",
 "transaction_code": "TEST-DP17xxxx251xxx2",
 "amount": "500.00",
 "bank_account_number": "1234567890",
 "bank_account_name": "John Doe",
 "bank_name": "Dummy Bank",
 "expired_at": "2025-12-09 12:00:00",
 "expired_timezone": "GMT+07:00"
}
```


PROMPTPAY:


```
{
 "status": "success",
 "message": "[DP] Submit Transaction Success!",
 "transaction_no": "DP17xxxx251xxx2",
 "transaction_code": "TEST-DP17xxxx251xxx2",
 "amount": "500.00",
 "qr": "https://dummy.qr/code.png",
 "expired_at": "2025-12-09 12:00:00",
 "expired_timezone": "GMT+07:00"
}
```


9.2 Submit UTR API (for INR only)

To complete payment request, operator need to submit UTR to S88pay use this API:

{base_url}/api/{merchant_code}/v3/submit-utr

- base_url : Provided by provider
- Method : POST

- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	transaction_code	Y	Exact value that is sent from the payment request API before.
2	utr	Y	Unique 12-digits number.

Combine all of the parameters above into one string, separated by "&" character, and then encrypt it using encrypt_decrypt algorithms.

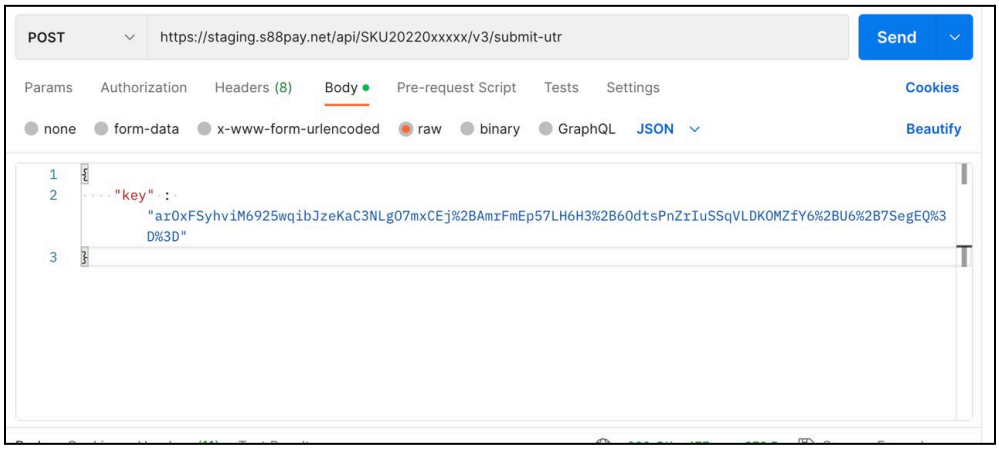
For example :

$str = transaction_code=TEST-DP-xxxxxxxxxx&utr=xxxxxxxxxxx

Encryption sample :

$key = encrypt_decrypt('encrypt',$str, '{your_api_key}', '{your_secret_key}')

Request example :



Response example :

{

"status": "success",

"message": "Success Submit Utr!"

}

9.3 Submit RefNo API (for BDT and MMK)

To complete payment request, operator need to submit RefNo to S88pay use this API:

{base_url}/api/{merchant_code}/v3/submit-refno

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	transaction_code	Y	Exact value that is sent from the payment request API before.
2	ref_no	Y	For MMK is Last 5 digits of the transaction.

Combine all of the parameters above into one string, separated by "&" character, and then encrypt it using encrypt_decrypt algorithms.

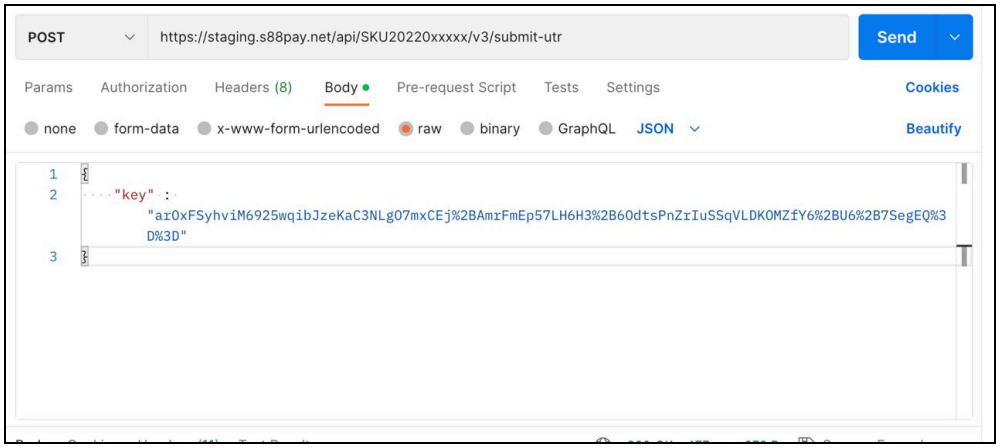
For example :

$str = transaction_code=TEST-DP-xxxxxxxxxx&ref_no=xxxxxxxxxxx

Encryption sample :

$key = encrypt_decrypt('encrypt',$str, '{your_api_key}', '{your_secret_key}')

Request example :



Response example :

{

"status": "success",

"message": "Success Submit RefNo!"

}

9.4 Submit Depositor Account (for MMK only)

To complete payment request, operator need to submit Depositor name to S88pay use this API:

{base_url}/api/{merchant_code}/v3/submit-depositor

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	transaction_code	Y	Exact value that is sent from the payment request API before.
2	depositor_name	Y	Depositor Name

Combine all of the parameters above into one string, separated by “&” character, and then encrypt it using encrypt_decrypt algorithms.

For example :

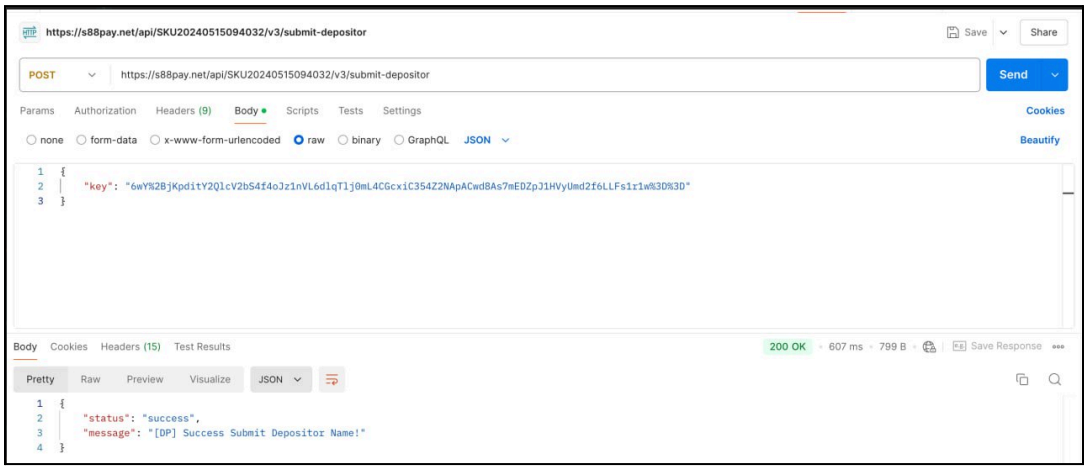
{

“transaction_code”: “TEST001”,

“depositor_name”: “”,

}

Request example :



Response example :

{

"status": "success",

"message": "[DP] Success Submitted Depositor Name!.",

}

10. Payment Request v4 (for INR, BDT, VND, JPY, BRL, IDR, THB, TRY, MMK, MXN, KHR, MYR, PHP, KRW merchant only)

In addition to Payment Request v2 and v3, S88Pay now offers Payment Request v4, which combines the features of both versions.

10.1 Transaction Request API V4

{base_url}/api/{merchant_code}/v4/dopayment ***(for INR, BDT, VND, JPY, BRL, IDR, THB, TRY, MMK, MXN, KHR, MYR, PHP, KRW)***

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	merchant_code	Y	Provided by provider
2	merchant_api_key	Y	Provided by provider
3	transaction_code	Y	Generated by operator, use string data type. Must be unique for each transactions
4	transaction_timestamp	Y	Generated by the operator. Integer data type. This parameter describes the transaction request time. The more detailed information regarding timestamps, please visit : https://www.epochconverter.com/. Please note that we only process the time stamp on these time limit: min: 1 hour before now and max: 5 minutes after now
5	transaction_amount	Y	Using double data type

6	payment_code	Y	Use string values for example *P001*. Please contact the administrator to get your payment code.
7	user_id	Y	● Use string values. ● For JPY, enter a user name (English or Japanese).
8	currency_code	Y	Please refer to merchant main currency list (point 13)
9	bank_code	N	Required just on BDT, VND, THB, TRY, IDR, MYR, MMK, PHP online bank payment method (points 15, 16)
10	depositor_name	N	Mandatory for JPY, KRW and BDT Bank Transfer
11	depositor_account_number	Y	Mandatory for THB Bank Transfer
12	callback_url	N	url callback beside url set from BO
13	identity_id	Y	required for Turkey (TRY)
14	phone	N	Required for: ● MMK (WAVEPAY only: fill with Wavepay Account Number)
15	card_serial	N	Only for VND. This value is required for Card PC. Must be provided in numeric format.
16	card_pin	N	Only for VND. This value is required for Card PC. Must be provided in numeric format.

combine all of the parameters above into one string, separate by “&” character and then encrypt it using encrypt_decrypt algorithms.

For example :

$str = "merchant_code=xxx&merchant_api_key=xxx&transaction_code=xxx&transaction_timestamp=xxx&payment_code=xxx&transaction_amount=xxx&user_id=xxx&currency_code=xxx"

Encryption sample :

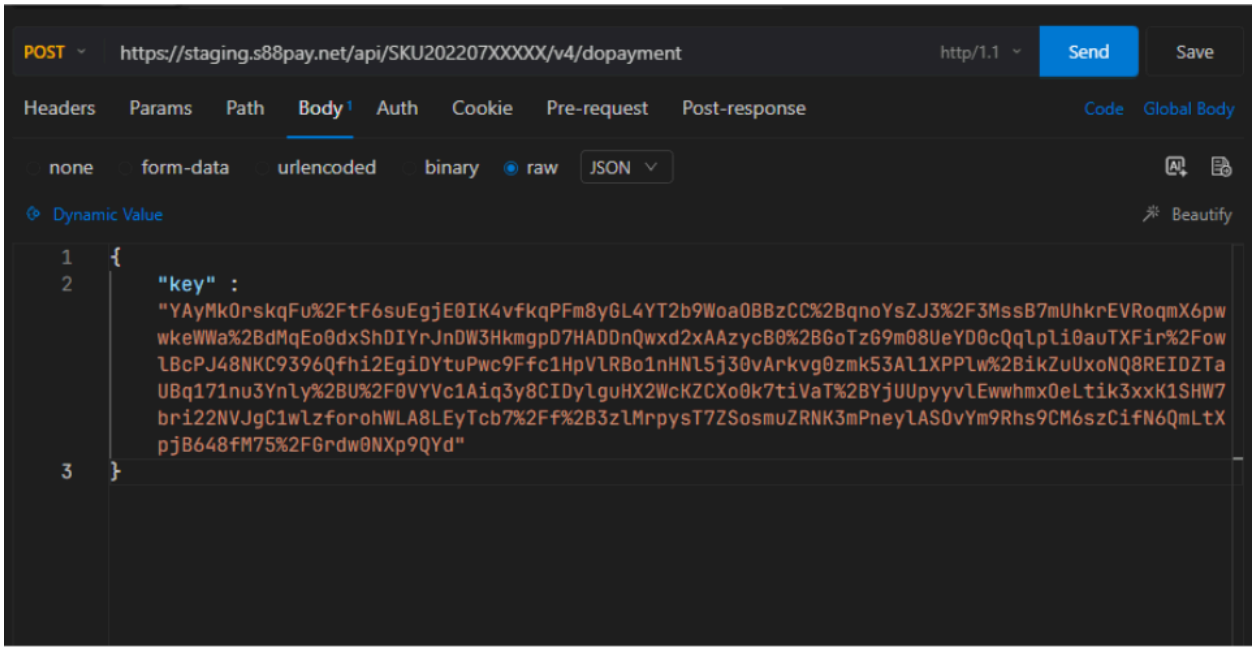
$key = encrypt_decrypt('encrypt',$str, '{your_api_key}', '{your_secret_key}')

Encryption result example :

"YAyMkOrskqFu%2FtF6suEgjE0IK4vfkqPFm8yGL4YT2b9WoaOBBzCC%2BqnoYsZJ3%2F3MssB7mUhkrEVRoqmX6pwwkeWWa%2BdMqEo0dxShDIYrJnDW3HkmgpD7HADDnQwxd2xAAzycB0%2BGoTzG9m08UeYD0cQqlpli0auTXFir%2FowlBcPJ48NKC9396Qfhi2EgiDYtuPwc9Ffc1HpVlRBo1nHNl5j30vArkvg0zmk53Al1XPPlw%2BikZuUxoNQ8REIDZTaUBq171nu3Ynly%2BU%2F0VYVc1Aiq3y8CIDylguHX2WcKZCXo0k7tiVaT%2BYjUUpyyvlEwwhmxOeLtik3xxK1SHW7bri22NVJgC1wlzforohWLA8LEyTcb7%2Ff%2B3zlMrpysT7ZSosmuZRNK3mPneylASOvYm9Rhs9CM6szCifN6QmLtXpjB648fM75%2FGrdw0NXp9QYd"

Refer to point 6 for the encryption/decryption function explanation.

Request example :



Response example for INR	**Status: 200**
{ *"status": "success",*	

```
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "upi_id": "example@upi"
   "qr":"https://s88pay.net/imageencoded=true&url=aHR0cHM6Ly9hcGkucGF5cHJvc3R1ZGvLmNvbS9hc3NldC91cGkvY2FjaGVkLzgzMmE1NmZlLTBkYzktNDc0Ni05NDhkLWFlYmY3E5NDlkMw==",
   "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+05:30"
}
```


Response example for BDT	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "wallet_number": "01861843585"
    "wallet_number_type": "Agent"
    "wallet_name": "rocket"
    "wallet_code": "1001"
   "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+05:30"
}
```

Response example for VND	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "wallet_number":"0797748156",
    "wallet_name":"Pham Chi Cuong",
    "wallet_code":"momo",
    "match_code":"685532",
    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+07:00"
}
```


Response example for JPY	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "bank_code": "0310",
    "branch": "Natsxxxe",
    "branch_code": "559",
    "virtual_account": "964xx39",
    "bank_name": "GMxxxらネット銀行",
    "bank_account_name": "合同会社xxxASE",
```

```
    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873387xxxx
        x&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+09:00"
}
```


Response example for BRL	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "pix_code": "000201265800xxxxx",
    "qr":"https://s88pay.net/imageencoded=true&url=aHR0cHM6Ly9hcGkucGF5cHJvc3R1ZGv
        LmNvbS9hc3NldC91cGkvY2FjaGVkLzgzMmE1NmZlLTBkYzktNDc0Ni05NDhkLWFlY
        mY3E5NDlkMw==",
    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT-03:00"
}
```


Response example for IDR	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
```

```
    "amount": "510.00",
    "bank_name":"QRIS",
    "account_number":"",
    "qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29
        tL3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY2
        9sb3I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJB
        TksuV1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYy
        NDAzMDNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzE
        zNDAzMDNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM
        1NCVTYwMTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwN
        DI2NTcwNTUwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X09
        2MGFmYTRMZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",
    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+07:00"
}
```


Response example for THB	
Using: PROMPTPAY	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29t
        L3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY29s
        b3I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJBTks
        uV1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYyNDA
        zMDNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzEzND
        AzMDNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM1NCV
        TYwMTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwNDI2NTc
        wNTUwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X092MGFm
        YTRMZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",
```

“qr_base64”:”iVBORw0KGgoAAAANSUhExxxxx”,

"pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873387xxxxx&lang=en",

"expired_at":"2023-06-22 15:06:04",

"expired_timezone":"GMT+07:00"

}

Using: b2b	**Status: 200**

{

"status": "success",

"message": "Submit Transaction Success!",

"transaction_no": "DP16873387xxxxx",

"transaction_code": "TEST-DP-16873xxxxx",

"amount": "510.00",

"bank_account_number":"123-4-xxxxx",

"bank_account_name":"Somchai xxxxx",

"qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29tL3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY29sb3I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJBTksuV1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYyNDAzMDNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzEzNDAzMDNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM1NCVTYwMTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwNDI2NTcwNTUwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X092MGFmYTRMZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",

"pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873387xxxxx&lang=en",

"expired_at":"2023-06-22 15:06:04",

"expired_timezone":"GMT+07:00"

}

Response example for TRY	**Status: 200**

{

```
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
   "iban": "TR33 0006 xxxxx",
   "name": "Ayşe xxxxx",
   "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+03:00"
}
```


Response example for MKK	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
   "bank_name": "AYA Bank",
   "account_number": "test",
   "name": "test",
   "reference": "OS-0065785",
   "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+06:30"
}
```

Response example for MXN	Status: 200


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
   "account_number": "01218000xxxxx",
   "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+06:00"
}
```


Response example for KHR	Status: 200


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
   "account_number": "01218000xxxxx",
   "qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29t
        L3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY29s
        b3I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJBTks
        uV1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYyNDA
        zMDNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzEzND
        AzMDNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM1NCV
        TYwMTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwNDI2NT
        cwNTUwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X092MGF
        mYTRMZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",
```

```
    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+07:00"
}
```


Response example for MYR	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "wallet_name": "TEST",
    "qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29t
        L3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY29s
        b3I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJBTks
        uV1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYyNDAz
        MDNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzEzNDA
        zMDNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM1NCVT
        YwMTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwNDI2NTc
        wNTUwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X092MGFm
        YTRMZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",
    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP168733
        87xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+08:00"
}
```


Response example for PHP	
Using: b2b	**Status: 200**

```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "bank_account_number":"123-4-xxxxx",
    "bank_account_name":"Somchai xxxxx",
    "bank_name":"xxxxx",
    "bank_branch":"xxxxx",
    "qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29tL3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY29sb3I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJBTksuV1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYyNDAzMDNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzEzNDAzMDNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM1NCVTYwMTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwNDI2NTcwNTUwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X092MGFmYTRMZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",
    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+05:30"
}
```


Using: others	**Status: 200**


```
{
    "status": "success",
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
    "bank_account_number":"123-4-xxxxx",
    "bank_account_name":"Somchai xxxxx",
```

```
    "qr":"https://s88pay.net/image?encoded=true&url=aHR0cHM6Ly9hcGkucXJzZXJ2ZXIuY29tL
        3YxL2NyZWF0ZS1xci1jb2RlLz9zaXplPTUxMng1MTImYmdjb2xvcj1GRkZGRkYmY29sb3
        I9MzkzQzNFJmRhdGE9MDAwMjAxMDEwMjEyMjY2NzAwMTZDT00uTk9CVUJBTksuV
        1dXMDExODkzNjAwNTAzMDAwMDA4ODU3MDAyMTQzNTI1MzY5MDczNzYyNDAzM
        DNVTUU1MTQ0MDAxNElELkNPLlFSSVMuV1dXMDIxNUlEMjAyMzI1MjI1NzEzNDAzM
        DNVTUU1MjA0NTQ5OTUzMDMzNjA1NDA2MTAwMDAzNTgwMklENTkwM1NCVTYw
        MTVKQUtBUlRBK1NFTEFUQU42MTA1MTI5NTA2Mjg0MDExNDAzMjAwNDI2NTcwNT
        UwMDUyNTBmYmQxNzQ4LTQxOWQtNDJjYi1hN2ZlLWMwNjE4cGF5X092MGFmYTRM
        ZE8yNDA2MDcwM0EwMTA4MDRQT1NQNjMwNEEyRTg=",

    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP168733
        87xxxxx&lang=en",

    "expired_at":"2023-06-22 15:06:04",

    "expired_timezone":"GMT+05:30"

}
```


Response example for PKR	**Status: 200**


```
{

    "status": "success",

    "message": "Submit Transaction Success!",

    "transaction_no": "DP16873387xxxxx",

    "transaction_code": "TEST-DP-16873xxxxx",

    "amount": "510.00",

    "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",

    "expired_at":"2023-06-22 15:06:04",

    "expired_timezone":"GMT+05:00"

}
```


Response example for KRW	**Status: 200**


```
{

    "status": "success",
```

```
    "message": "Submit Transaction Success!",
    "transaction_no": "DP16873387xxxxx",
    "transaction_code": "TEST-DP-16873xxxxx",
    "amount": "510.00",
   "name": "Juna",
   "bank_name": "케이뱅크 K-Bank",
   "bank_account_number": "2023030xxxxxxx",
   "bank_account_name": "TEST"
   "pay_url":"https://s88pay.net/SKU20220714094803/v4/deposit?transaction_no=DP16873
        387xxxxx&lang=en",
    "expired_at":"2023-06-22 15:06:04",
    "expired_timezone":"GMT+09:00"
}
```


10.2 Submit UTR API (for INR only)

To complete payment request, operator need to submit UTR to S88pay use this API:

{base_url}/api/{merchant_code}/v4/submit-utr

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	transaction_code	Y	Exact value that is sent from the payment request API before.
2	utr	Y	Unique 12-digits number.

Combine all of the parameters above into one string, separated by "&" character, and then encrypt it using encrypt_decrypt algorithms.

For example :

$str = “transaction_code=TEST-DP-xxxxxxxxxx&utr=xxxxxxxxxxx”

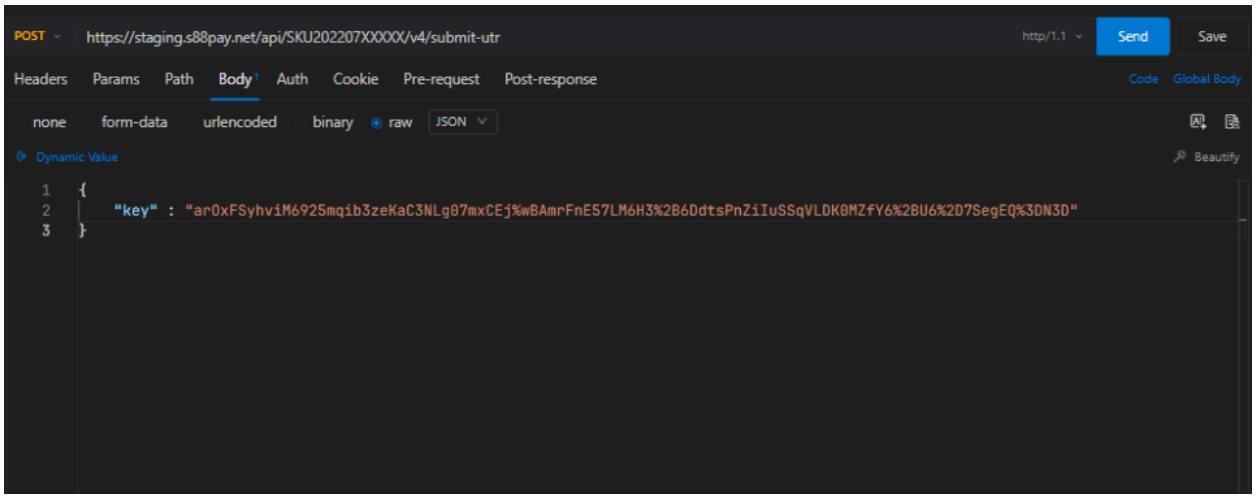
Encryption sample :

$key = encrypt_decrypt(‘encrypt’,$str, ‘{your_api_key}’, ‘{your_secret_key}’)

Encryption result example :

“arOxFSyhviM6925mqib3zeKaC3NLg07mxCEj%wBAmrFnE57LM6H3%2B6DdtsPnZiIuSSqVLDK0MZfY6%2BU6%2D7SegEQ%3DN3D”

Request Example :



Response example	**Status: 200**
{ *"status": "success",* *"message": "Success Submit Utr!"* *}*	

10.3 Submit RefNo API (for BDT and MMK)

To complete payment request, operator need to submit UTR to S88pay use this API:

{base_url}/api/{merchant_code}/v4/submit-refno

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	transaction_code	Y	Exact value that is sent from the payment request API before.
2	ref_no	Y	For MMK is Last 5 digits of the transaction.

Combine all of the parameters above into one string, separated by "&" character, and then encrypt it using encrypt_decrypt algorithms.

For example :

$str = "transaction_code=TEST-DP-xxxxxxxxxx&utr=xxxxxxxxxxxx"

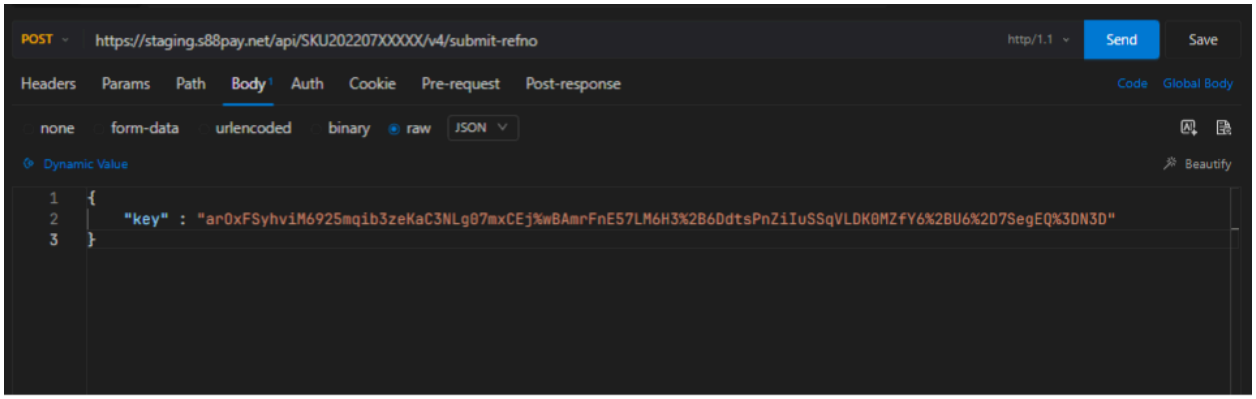
Encryption sample :

$key = encrypt_decrypt('encrypt',$str, '{your_api_key}', '{your_secret_key}')

Encryption result example :

"arOxFSyhviM6925mqib3zeKaC3NLg07mxCEj%wBAmrFnE57LM6H3%2B6DdtsPnZiIuSSqVLDK0MZfY6%2BU6%2D7SegEQ%3DN3D"

Request Example :



Response example	**Status: 200**
{ *"status": "success",* *"message": "Success Submit RefNo!"* *}*	

11. DEPOSIT TRANSACTIONS STATUS LIST

transaction_status	**name**	**description**
1	Pending	Transaction is pending
2	Completed	Transaction is completed
3	Failed	Transaction is Failed
4	Manual process	This transaction is necessary to be in contact with CS/Admin if the status is Manual process

12. PAYOUT TRANSACTIONS STATUS LIST

transaction_status	name	description
1	Pending	Transaction is pending
2	Completed	Transaction is completed
3	Failed	Transaction is Failed
4	Refund	Transaction is Refunded

13. PAYMENT (Deposit) METHOD

Each merchant has a unique payment code, which is a unique identifier that is used to process payments. To get your payment code, please contact us. We will provide a unique payment code for the merchant.

14. PAYOUT (Withdraw) METHOD

Each merchant has a unique payout code, which is a unique identifier that is used to process payouts. To get your payout code, please contact us. We will provide a unique payout code for the merchant.

15. CURRENCY LIST

Currency Code	Currency Name
INR	India Rupee
USDT	USD Token

CNY	Chinese Yuan
VND	Vietnam Dong
IDR	Indonesian Rupiah
BDT	Bangladesh Taka
JPY	Japanese Yen
BRL	Brazilian Real
TRY	Turkish New Lira
MMK	Myanmar Kyat
THB	Thai Baht
KRW	Korean Won
PKR	Pakistani Rupee

16. CNY Bank Code List

Bank Code	**Bank Name**
BOC	Bank of China
ICBC	ICBC
CCB	China Construction Bank
COMM	China Bank of Communications
HXB	Hua Xia Bank of China
CMB	China Merchants Bank
CEB	China Everbright Bank
ABC	Agricultural Bank of China

CMBC	China Minsheng Bank
CIB	Industrial Bank of China
CITIC	China CITIC Bank
BJB	Bank of China Beijing
SPDB	China Pudong Development Bank
BOSH	Bank of China Shanghai
HELP	Ping An Bank of China
PSBC	China Postal Savings Bank
GBS	China Hengfeng Bank
CZB	China Zheshang Bank
CBHB	China Bohai Bank
CSCB	Bank of China Changsha

17. VND Bank Code List

Bank Code	**Bank Name**
Deposit	
momo	Momo Ewallet
zalopay	ZaloPay Ewallet
viettelpay	ViettelPay Ewallet
vietcombank	VietcomBank
acbbank	ACBBank

msbbank	MSBBank
mbbank	MBBank
vietinbank	VietinBank
vpbank	VPBank
bidv	BIDV
tpbank	TPBank
ocb	Orient Commercial Bank
seabank	Seabank
bvbank	VIETCAPITALBANK
lpbank	Lien Viet Post Joint Stock Commercial Bank
vikkibank	VikkiBank
shinhanbank	ShinhanBank
hdbank	HDBank
mobifone	MobiFone
viettel	Viettel
pvcombank	PVcomBank
vinaphone	VinaPhone
techcombank	TechcomBank
Withdraw	
970407	NGAN HANG TMCP KY THUONG VIET NAM (TECHCOMBANK)
422589	NGAN HANG TNHH MTV CIMB (CIMB)
458761	NGAN HANG TNHH MTV HSBC (VIET NAM)
546034	NGAN HANG TMCP VIET NAM THINH VUONG - NH SO CAKE BY VPBANK (CAKE)

546035	NGAN HANG TMCP VIET NAM THINH VUONG - NH SO UBANK BY VPBANK (UBANK)
796500	DBS - CHI NHANH THANH PHO HO CHI MINH
801011	NGAN HANG NONGHYUP CHI NHANH HA NOI (NHB HN)
970400	NGAN HANG TMCP SAI GON CONG THUONG (SAIGONBANK)
970401	NGAN HANG TMCP PT NHA DONG BANG SONG CUU LONG
970403	NGAN HANG TMCP SAI GON THUONG TIN (SACOMBANK)
970406	NGAN HANG TMCP DONG A (DONGABANK)
963388	NGAN HANG SO TIMO BY BAN VIET BANK (TIMO)
970408	NGAN HANG TMCP DAU KHI TOAN CAU (GPB)
970409	NGAN HANG TMCP BAC A (NASB)
970410	NGAN HANG TNHH MTV STANDARD CHARTERED VIETNAM (SCVN)
970412	NGAN HANG TMCP DAI CHUNG VIET NAM (PVCOMBANK)
970414	NGAN HANG TMCP DAI DUONG (OCEANBANK)
970415	NGAN HANG TMCP CONG THUONG VIET NAM (VIETINBANK)
970416	NGAN HANG TMCP A CHAU (ACB)
970418	NGAN HANG TMCP DAU TU VA PHAT TRIEN VIET NAM (BIDV)
970419	NGAN HANG TMCP QUOC DAN (NCB)
970421	NGAN HANG LIEN DOANH VIET - NGA (VRB)
970422	NGAN HANG TMCP QUAN DOI (MB)
970423	NGAN HANG TMCP TIEN PHONG (TPBANK)
970424	NGAN HANG TNHH MTV SHINHAN VIET NAM (SHBVN)
970425	NGAN HANG TMCP AN BINH (ABBANK)

970426	NGAN HANG TMCP HANG HAI VIET NAM (MSB)
970427	NGAN HANG TMCP VIET A (VAB)
970428	NGAN HANG TMCP NAM A (NAMABANK)
970429	NGAN HANG TMCP SAI GON (SCB)
970430	NGAN HANG TMCP XANG DAU PETROLIMEX (PG BANK)
970431	NGAN HANG TMCP XUAT NHAP KHAU VIET NAM (EXIMBANK)
970432	NGAN HANG TMCP VIET NAM THINH VUONG (VPBANK)
970433	NGAN HANG TMCP VIET NAM THUONG TIN (VIETBANK)
970434	NGAN HANG TNHH INDOVINA
970436	NGAN HANG TMCP NGOAI THUONG VIET NAM (VIETCOMBANK)
970437	NGAN HANG TMCP PHAT TRIEN TP.HCM (HDB)
970438	NGAN HANG TMCP BAO VIET (BVB)
970439	NGAN HANG TNHH MTV PUBLIC VIET NAM (PBVN)
970440	NGAN HANG TMCP DONG NAM A (SEABANK)
970441	NGAN HANG TMCP QUOC TE VIB
970442	NGAN HANG TNHH MTV HONGLEONG VIET NAM
970443	NGAN HANG TMCP SAI GON - HA NOI (SHB)
970444	TM TNHH MTV Xay dung Viet Nam (CBBank)
970448	NGAN HANG TMCP PHUONG DONG (OCB)
970449	NGAN HANG TMCP BUU DIEN LIEN VIET (LPB)
970452	NGAN HANG TMCP KIEN LONG (KIENLONGBANK)
970454	NGAN HANG TMCP BAN VIET (VIETCAPITAL BANK)

970455	NGAN HANG CONG NGHIEP HAN QUOC (IBK)
970456	NGAN HANG CONG NGHIEP HAN QUOC CHI NHANH HCM (IBK HCM)
970457	NGAN HANG WOORIBANK
970458	NGAN HANG TNHH MTV UNITED OVERSEAS BANK (UOB)
970462	NGAN HANG KOOKMIN - CN HA NOI

18. THB Bank Code List

Bank Code	Bank Name
DEPOSIT BANK CODE	
PROMPTPAY	Promptpay
b2b	Bank Transfer
WITHDRAW BANK CODE	
BBL	Bangkok Bank
BAY	Bank Of Ayudhya
KTB	Krungthai Bank
KBANK	KasiKorn Bank
KKP	Kiatnakin Phatra Bank
TTB	TMBThanachart Bank
SCB	Siam Commercial Bank
LH	Land and Houses Bank
GSB	Government Savings Bank

BAAC	Bank for Agriculture
GHB	Government Housing Bank
UOB	United Overseas Bank
CIMB	CIMB Thai Bank
TCRB	Thai Credit Retail Bank
TMB	Thai Military Bank
TISCO	Tisco Bank
Depositor BANK CODE	
BBL	Bangkok Bank
KBANK	KasiKorn Bank
KTB	Krungthai Bank
TMB	Thai Military Bank
SCB	Siam Commercial Bank
CIMB	CIMB Thai Bank
UOB	United Overseas Bank
BAY	Bank Of Ayudhya
GSB	Government Savings Bank
GHB	Government Housing Bank
BAAC	Bank for Agriculture
TISCO	Tisco Bank

KKP	Kiatnakin Phatra Bank
TCRB	Thai Credit Retail Bank
LHBANK	Land and Houses Bank
TTB	TMBThanachart Bank

19. IDR BANK CODE LIST

Bank Code	Bank Name
Deposit Bank Code	
BCA	Bank Central Asia
DANAMON	Danamon Bank
BNI	Bank Negara Indonesia
BRI	Bank Rakyat Indonesia
CIMB	CIMB Bank
MBBI	Maybank Bank
MDR	Mandiri Bank
PMTB	Permata Bank
PANIN	Panin Bank
GOPAY	Gopay Wallet
DOKUWALLET	Doku Wallet
DANA	Dana Wallet
LINKAJA	LinkAja
SHOPEEPAY	Shopeepay Wallet

OVO	Ovo Wallet
SAKUKU	Sakuku Wallet
QRIS	QRIS Payment
Withdraw Bank Code	
BNI	Bank Negara Indonesia
BCA	Bank Central Asia
BRI	Bank Rakyat Indonesia
MANDIRI	Mandiri Bank
MAYBANK	Maybank Bank
PMTB	Permata Bank
OVO	Ovo Wallet
LINKAJA	Linkaja Wallet
GOPAY	Gopay Wallet
DANA	Dana Wallet
SHOPEEPAY	Shopeepay Wallet
PANIN	Bank Panin
MAYBANK	Bank BII Maybank
BRI_SYR	Bank BRI SYARIAH
BNI_SYR	Bank BNI Syariah
BANK_OCBC	Bank OCBC – Indonesia
BJB	Bank BANTEN(Bank BJB)
BJB SYARIAH	Bank BANTEN(Bank BJB SYARIAH)
CIMB	Bank CIMB Niaga
CIMB SYARIAH	Bank CIMB SYARIAH

CAPITAL	Bank Capital Indonesia
COMMONWEALTH	Bank Commonwealth
DANAMON	Bank Danamon
DANAMON SYARIAH	Bank Danamon SYARIAH
MANDIRI_SYR	Bank Syariah Mandiri
PERMATA	Permata Bank
MEGA_SYR	Bank Syariah Mega
BALI	BPD Bali
CHINATRUST	Bank CTBC (China Trust) Indonesia
BANK_MAYBANK	Bank Maybank Indocorp
BANK_MERIN	Bank Merincorp
BANK_AGRIS	Bank Agris
TELKOMSEL_TCASH	Telkomsel Tcash
LINK_AJA	Link Aja
DOMPETKU	Indosat Dompetku
BPR_KS	BPR KS
HARDA_INTERNASIONAL	Bank Harda
BANK_VICTORIA	Bank Victoria International
VICTORIA INTL	Bank Victoria International
MANDIRI_TASPEN	Bank Mandiri Taspen Pos
FAMA	Bank Fama Internasional
CENTRATAMA	Centratama Nasional Bank
INDEX_SELINDO	Bank Index Selindo
MAYORA	Bank Mayora Indonesia
MULTI_ARTA_SENTOSA	Bank Multi Arta Sentosa

BTPN_SYARIAH	Bank Purba Danarta
ARTOS	Bank Artos IND
BANK_KESEJAHTERAAN_EKONOMI	Bank Kesejahteraan Ekonomi(seabank)
BANK_ANGLOMAS	Anglomas Internasional Bank
BANK_LIMAN	Liman International Bank
BANK_AKITA	Bank Akita
OCBC	Bank Dipo International (Bank Sahabat Sampoerna)
BANK_PERSY	Bank Persyarikatan Indonesia
PRIMA_MASTER	Prima Master Bank
BANK_HARFA	Bank Harfa
BANK_INA	Bank Ina Perdana
NATIONALNOBU	Bank Alfindo (Bank National Nobu)
ROYAL	Bank Royal Indonesia
BANK_INDOMONEX	Bank Indomonex (Bank SBI Indonesia)
AGRONIAGA	Bank BRI Agro
BANK_YUDHA	Bank Yudha Bhakti
BANK_BUMIPUTERA	Bank MNC / Bank Bumiputera
BANK_BINTANG	Bank Bintang Manunggal
JASA_JAKARTA	Bank Jasa Jakarta
BANK_SRI_PARTHA	Bank Sri Partha
BISNIS_INTERNASIONAL	Bank Bisnis Internasional
BUKOPIN	Bank Bukopin
MEGA	Bank Mega
BJB_SYR	Bank BJB Syariah
BANK_SWAGUNA	Bank Swaguna

TABUNGAN_PENSIUNAN_NASIONAL	Bank Tabungan Pensiunan Nasional (BTPN)
JENIUS	JENIUS
BANK_HIM	Bank Himpunan Saudara 1906
BTN	Bank Tabungan Negara (BTN)
QNB_INDONESIA	Bank QNB Kesawan (Bank QNB Indonesia)
BANK_HARM	Bank Harmoni International
ICBC	Halim Indonesia Bank (Bank ICBC Indonesia)
CCB	Bank Windu Kentjana
GANESHA	Bank Ganesha
BANK_HAGAKITA	Bank Hagakita
MASPION	Bank Maspion Indonesia
SHINHAN	Bank Metro Express (Bank Shinhan Indonesia)
MESTIKA_DHARMA	Bank Mestika Dharma
MUAMALAT	Bank Muamalat
BANK_OF_INDIA	Bank of India Indonesia
NUSANTARA_PARAHYANGAN	Bank Nusantara Parahyangan
BANK_SULTRA	Bank Sultra
SULAWESI	Bank Sulawesi Tengah
BENGKULU	Bank Bengkulu
PAPUA	Bank Papua
MALUKU	Bank Maluku Malut
BANK_NTT	Bank NTT
BPD_NTB	Bank NTB, NTB Syariah
SUMSEL_DAN_BABEL_SULUT	Bank Sulut Gorontalo
SULSELBAR	Bank Sulsel dan Barat

BPD_KALTENG	Bank Kalteng
BPD_KAITIM	Bank Kalimantan Timur dan Utara
KALIMANTAN_BARAT	Bank Kalimantan Barat
BPD_KALSEL	Bank Kalsel
LAMPUNG	Bank Lampung
SUMSEL_DAN_BABEL	Bank Sumsel Babel
RIAU_DAN_KEPRI	Bank Riau
BANK_NAGARI	Bank Nagari
SUMUT	Bank Sumut
ACEH	BPD Aceh, BPD Aceh Syariah
JAMBI	BPD Jambi
BANK_JATIM	Bank Jatim
BANK_JATENG	Bank Jateng
DAERAH_ISTIMEWA	BPD DIY
DKI	Bank DKI
BANK_JABAR	Bank Jabar dan Banten (BJB)
MAYAPADA	Bank Mayapada
JTRUST	Bank JTRUST
BANK_IFI	Bank IFI
BANK_HAGA	Bank Haga
BANK_ANTAR	Bank Antardaerah
BANK_EKONOMI	Bank Ekonomi
BUMI_ARTA	Bank Bumi Arta
BOC	Bank OF China
BANK_WOOR	Bank Woori Indonesia
DEUTSCHE	Deutsche Bank AG.

BANK_ANZ	Bank ANZ Indonesia
BANK_DANAMON	Korea Exchange Bank Danamon
BNP_PARIBAS	Bank BNP Paribas Indonesia
BANK_KEPPEL	Bank Keppel Tatlee Buana
BANK_ABN	Bank ABN Amro
STANDARD_CHARTERED	Standard Chartered Bank
MIZUHO	Bank Mizuho Indonesia
RESONA	Bank Resona Perdania
DBS	Bank DBS Indonesia
MITSUI	Bank Sumitomo Mitsui Indonesia
BANK_TOKYO	The Bank of Tokyo Mitsubishi UFJ LTD
HSBC	The Hongkong & Shanghai B.C. (Bank HSBC)
BANK_COMP	The Bangkok Bank Comp. LTD
BANK_C_AGR	Bank Credit Agricole Indosuez
ARTHA	Bank Artha Graha Internasional
BANK_ING	ING Indonesia Bank
BAML	Bank of America, N.A
JPMORGAN	JP. Morgan Chase Bank, N.A
CITIBANK	Citibank
BANK_LTD	American Express Bank LTD
NISP	Bank OCBC NISP
BANK_LIPPO	Bank Lippo
BANK_BUANA	Bank UOB Indonesia
ARTA_NIAGA_KENCANA	Bank Arta Niaga Kencana
EXIMBANK	Bank Ekspor Indonesia

SINARMAS SYARIAH	Bank Sinarmas SYARIAH
BTN SYARIAH	Bank BTN SYARIAH
BISNIS	Bisnis International
JPMORGAN	JPMorgan Chase Bank
SHINHAN	Bank SHINHAN
IBK	Bank Agris Prima(IBK)
MIZUHO	MIZUHO
BANK_SWADESI	Bank Swadesi
BSI	BANK SYARIAH INDONESIA (BSI)

20. BDT BANK CODE LIST

Bank Code	Bank Name
Deposit Bank Code	
1001	BKASH
1002	ROCKET
1003	UPAY
1004	NAGAD
Withdraw Bank Code	
2001	BKASH
2002	ROCKET
2003	UPAY
2004	NAGAD

21. MYR BANK CODE LIST

Bank Code	Bank Name
Deposit Bank Code	
CIMB	CIMB Bank Berhad
MBB	Maybank Berhad
HLB	Hong Leong Bank Berhad
RHB	RHB Banking Group
PBB	Public Bank Berhad
AMB	AmBank Group
BIMB	Bank Islam Malaysia Berhad
Withdraw Bank Code	
CIMB	CIMB Bank Berhad
MBB	Maybank Berhad
HLB	Hong Leong Bank Berhad
RHB	RHB Banking Group
PBB	Public Bank Berhad
AMB	AmBank Group
BIMB	Bank Islam Malaysia Berhad

22. PHP BANK CODE LIST

Bank Code	Bank Name
Deposit Bank Code	

gcash	Gcash
Withdraw Bank Code	
gcash	Gcash
bpi	BPI Bank
Unibank	BDO Unibank
mbt	Metropolitan Bank and Trust Co
LBOB	LANDBANK / OFBank
SBC	Security Bank Corporation
UBP	Union Bank of the Philippines
PNB	Philippine National Bank
CBC	China Banking Corporation
EWBC	East West Banking Corporation
RCBC	RCBC/DiskarTech
UCPB	United Coconut Planters Bank (UCPB)
PSB	Philippine Savings Bank
AUB	Asia United Bank Corporation
PBC	Philippine Bank of Communications
DBP	Development Bank of the Philippines
AB	ALLBANK(A Thirft Bank)
Asenso	Rural Bank of Guinobatan / Asenso
BM	Bangko Mabuhay
BC	Bank of Commerce
BK	BanKo,A Subsidiary of BPI
Bayad	CIS Bayad Center / Bayad
BNB	BDO NeTwork Bank

CB	Camalig Bank
CARD Bank	CARD Bank
CLB	Cebuana Lhuillier Bank / Cebuana Xpress
CBS	China Bank Savings
Coins	DCPay / COINS.PH
CTBC	CTBC Bank (Philippines) Corporation
DCDB	Dumaguete City Development Bank
DB	Dungganon Bank
ESB	Equicom Savings Bank, Inc.
GP	GrabPay
ISLA	ISLA Bank
JC	Zybi Tech Inc. / JuanCash
Komo	East West Rural Bank / Komo
LSB	Legazpi Saving Bank
MBS	Malayan Bank Savings and Mortgage Bank, Inc.
MBP	Maybank Philippnies
MCCB	Mindanao Consolidated CoopBank
NB	Netbank
OP	OmniPay, Inc.
PRB	Partner Rural Bank (Cotabato), Inc.
PMP	PayMaya Philippines
PBB	Philippine Business Bank
PTC	Philippine Trust Company
PDB	Producers Bank
QB	Queenbank
QCRB	Quezon Capital Rural Bank

RBB	Robinsons Bank Corporation
SB	Seabank
SP	ShopeePay
SCB	Standard Chartered Bank
STP	Starpay
SLB	Sterling Bank of Asia
SSB	Sun Savings Bank
TC	TayoCash
USB	UCPB Savings Bank
USSC	USSC Monet Services
VB	Veterans Bank
RSB	RCBC Saving Bank
IB	ING Bank N.V.
WDB	Wealth Development Bank
usdt-trc	usdt-trc
(TRC)	USDT(TRC)
ERC	USDT-ERC
ALIPAY	Alipay / Lazada Wallet
BCH	Bank of China
BRB	Binangonan Rural Bank / BRBDigital
SME	CARD SME Bank
CPI	CIMB Philippines, Inc.
ERB	Entrepreneur Rural Bank, Inc./ENTRP
GOT	GoTyme Bank
IRI	I-Remit / iCASH
IEM	Infoserve / Nationlink

LDB	Luzon Development Bank
MYA	Maya Bank, Inc.
PAS	Pacific Ace Savings Bank
PPS	PalawanPay
TDB	Tonik Bank
TPI	TraxionPay/DigiCOOP/COOPNET
UDB	UnionDigital Bank

23. KRW BANK CODE LIST

은행 이름	**Bank Name**
Bank Code for Withdrawal	
bok_kr	한국은행 Bank of Korea
KDB_KR	한국산업은행 Korea Development Bank
tossbank_kr	토스뱅크 TossBank
kbank_kr	케이뱅크 K-Bank
kakaobank_kr	카카오뱅크 Kakao Bank
jejubank_kr	제주은행 Jeju Bank
jeonbuk_kr	전북은행 Jeonbuk Bank
wooribank_kr	우리은행 Woori Bank
shinhyup_kr	신협은행 Shinhyup Bank
shinhan_kr	신한은행 Shinhan Bank
suhyup_kr	수협은행 Suhyup Bank
kfccco_kr	새마을금고 Saemaul Geumgo

fsb_kr	상호저축은행 Mutual Savings Bank
busanbank_kr	부산은행 Busan Bank
daegubank_kr	대구은행 Daegu Bank
nonghyup_kr	농협은행 Nonghyup Bank
ibok_kr	기업은행 Industrial Bank of Korea
kookmin_kr	국민은행 Kookmin Bank
kwangju_kr	광주은행 Kwangju Bank
kyongnam_kr	경남은행 Kyongnam Bank
scb_kr	SC제일은행 Standard Chartered Bank
071	POST BANK 우체국은행
kebhana_kr	KEB하나은행 KEB Hana Bank

24. JPY BANK CODE LIST

Code	Bank Name
MHBK	株式会社みずほ銀行
UFJ	株式会社三菱UFJ銀行
SMBC	株式会社三井住友銀行
DIWA	株式会社りそな銀行
SAIB	株式会社埼玉りそな銀行
HKDB	株式会社北海道銀行
AOMB	株式会社青森銀行
MCHI	株式会社みちのく銀行
AKIT	株式会社秋田銀行
HOKB	株式会社北都銀行
SNAI	株式会社荘内銀行

YAMB	株式会社山形銀行
BAIW	株式会社岩手銀行
TOHK	株式会社東北銀行
BOSS	株式会社七十七銀行
TOHO	株式会社東邦銀行
GB	株式会社群馬銀行
ASIK	株式会社足利銀行
JOYO	株式会社常陽銀行
KGBK	株式会社筑波銀行
MUBK	株式会社武蔵野銀行
CHBA	株式会社千葉銀行
CHIK	株式会社千葉興業銀行
TOMI	株式会社東京都民銀行
HAMA	株式会社横浜銀行
DAIS	株式会社第四銀行
HETS	株式会社北越銀行
YCBS	株式会社山梨中央銀行
HABK	株式会社八十二銀行
RIKB	株式会社北陸銀行
HKOK	株式会社北國銀行
FKUI	株式会社福井銀行
SHIZ	株式会社静岡銀行
SRFX	株式会社スルガ銀行
SMZG	株式会社清水銀行
OGAK	株式会社大垣共立銀行
JURO	株式会社十六銀行
MIEB	株式会社三重銀行
HYKGTSU	株式会社百五銀行
SIGA	株式会社滋賀銀行
BOKF	株式会社京都銀行
OSAB	株式会社近畿大阪銀行
BIKE	株式会社池田泉州銀行
NANT	株式会社南都銀行
KIYO	株式会社紀陽銀行

TJMA	株式会社但馬銀行
BIRD	株式会社鳥取銀行
SGBK	株式会社山陰合同銀行
CHGK	株式会社中国銀行
HIRO	株式会社広島銀行
YMBK	株式会社山口銀行
AWAB	株式会社阿波銀行
HYAK	株式会社百十四銀行
IYOB	株式会社伊予銀行
SIKO	株式会社四国銀行
FKBK	株式会社福岡銀行
CHIH	株式会社筑邦銀行
BKSG	株式会社佐賀銀行
EITN	株式会社十八銀行
SHWA	株式会社親和銀行
HIGO	株式会社肥後銀行
OITA	株式会社大分銀行
MIYA	株式会社宮崎銀行
KAGO	株式会社鹿児島銀行
RYUB	株式会社琉球銀行
BOKI	株式会社沖縄銀行
NISI	株式会社西日本シティ銀行
KITQ	株式会社北九州銀行
SONY	ソニー銀行株式会社
RAKT	楽天銀行株式会社
NTSS	住信SBIネット銀行株式会社
JICR	auじぶん銀行
DNEX	株式会社大和ネクスト銀行
LTCB	株式会社新生銀行
NCBT	株式会社あおぞら銀行
JPPS	株式会社ゆうちょ銀行
SHBK	株式会社SBJ銀行
MTBC	三菱UFJ信託銀行株式会社
YTBC	みずほ信託銀行株式会社

STBC	三井住友信託銀行株式会社
CHAS	ニューヨークメロン信託銀行
MTBJ	日本マスタートラスト信託銀行
SSTB	ステート・ストリート信託銀行
NMTB	野村信託銀行株式会社
OTBC	オリックス銀行株式会社
SKTB	株式会社しんきん信託銀行
NCTB	農中信託銀行株式会社
SHTC	新生信託銀行株式会社
JSF	日証金信託銀行株式会社
JSTC	株式会社新銀行東京
JTSB	日本トラスティ・サービス信託銀行株式会社
TCSB	資産管理サービス信託銀行株式会社
CITI	シティバンク、エヌ・エイ东京支店
CHAS	JPモルガン・チェース银行
BOFA	バンク・オブ・アメリカ・エヌ・エイ
HSBC	香港上海銀行
SCBL	スタンダードチャータード銀行
BARC	バークレイズ銀行
DUTCH	オランダ銀行
KOEX	ハナ銀行
BKID	インド银行
IRVT	ニューヨーク銀行
BNPA	ビー・エヌ・ピー・パリバ銀行
SOGE	ソシエテ ジェネラル銀行
BOFC	ユニオン・バンク
HVBK	ウリィ银行
KODB	韓国産業銀行
CCBC	彰化商業銀行
FCBK	第一商業銀行
BKTW	台湾銀行
COMM	交通銀行
MBTC	メトロポリタン銀行

ICBK	中国工商銀行
CZNB	国民銀行
BBVA	ビルバオ・ビスカヤ・アルヘンタリア銀行
PPB	PayPay銀行
SEVEN	セブン銀行

25. BRL BANK CODE LIST

Bank Code	**Bank Name**
Withdraw Bank Code	
CPF	[the account value]
CNPJ	[the account value]
EMAIL	[the account value]
PHONE	[the account value]
RANDOMKEY	[the account value]
EVP	[the account value]

26. TRY BANK CODE LIST

Bank Code	**Bank Name**
Deposit Bank Code	
IBAN	IBAN
Withdraw Bank Code	
6006	IBAN

27. MMK BANK CODE LIST

Bank Code	Bank Name
Deposit Bank Code	
AYA	AYA Bank
KBZ	KBZ Bank
WAVEPAY	Wave Pay
KBZPAY	KBZ Pay
YOMA	YOMA Bank
Withdraw Bank Code	
AYA	AYA Bank
KBZ	KBZ Bank
WAVEPAY	Wave Pay
KBZPAY	KBZ Pay
YOMA	YOMA Bank

28. PKR BANK CODE LIST

Bank Code	Bank Name
Deposit Bank Code	
EASYPAISA	EASYPAISA
JAZZCASH	JAZZCASH

Withdraw Bank Code	
ALBARAKA	Al Baraka Islamic Bank
ABL	Allied Bank Limited
EASYPAISA	EASYPAISA
JAZZCASH	JAZZCASH
APNA	Apna Microfinance Bank
AKBL	Askari Bank Limited
BAHL	Bank Al Habib Limited
BAFL	Bank Alfalah Limited
BIPL	Bank Islami Pakistan Limited
BOK	Bank of Khyber
CITI	Citi Bank N.A.
DIBPL	Dubai Islamic Bank Pakistan Limited
FBL	Faysal Bank Limited
FINCA	FINCA Microfinance Bank
FWBL	First Women Bank Limited
HBL	Habib Bank Limited
HMB	Habib Metropolitan Bank Limited
JSBL	JS Bank Limited
MCB	MCB Bank Limited
MCB_ISLAMIC	MCB Islamic Bank
MEBL	Meezan Bank
NBP	National Bank of Pakistan
NRSP_MF_BANK	NRSP Microfinance Bank
NIB	NIB Bank Limited
SBL	Samba Bank Limited

SILK	Silk Bank Limited
SINDH	Sindh Bank Limited
SNBL	Soneri Bank Limited
SCB	Standard Chartered Bank Ltd
SMBL	Summit Bank Limited
TMB	Telenor Microfinance Bank
BOP	The Bank of Punjab
U_BANK	U Microfinance Bank
UBL	United Bank Limited

29. ERROR LIST

These are the error responses at "**dopayment**" end point :

Error Code	**Description**
109	Incomplete parameter
110	Invalid Bank ID
111	Invalid Transaction Code (not unique)
112	Invalid Timestamp
113	Invalid Transaction. The transaction is already processed
114	Active bank account is not found
115	Invalid Key / Merchant code
116	Bank is under maintenance
117	Server Error for this payout method !

118	Format Key Paramater Invalid
119	Insufficient Balance

These are the error responses at "**payout**" end point :

Error Code	**Description**
115	Invalid Key / Merchant code!, error code 115
109	Incomplete parameter!, error code 109
111	Invalid Transaction Code (not unique)!, error code 111
110	Payout Method Not Available, error code 110,
116	Payout method is under maintenance!, error code 116
117	Server error for this payout method!, error code 117
118	Format Key parameter invalid!, error code 118
114	Currency of payment method is not supported for your merchant!, error code 114
112	Invalid Timestamp!, error code 112
113	Currency rate is not supported for your merchant currency at this moment!, error code 113
119	Insufficient Balance!, error code 119
120	Invalid IFSC Code! Please try again, error code 120
121	Withdraw Bank not found!, error code 121
122	Amount cannot has decimal number, please try again! error code 122
123	Request Withdrawal Failed! Please Try Again, error code 123
124	Your IP address has been blocked, error code 124
131	There are problem to connect payout account, please try again!, error code 131

30. REGISTER KYC API

{base_url}/api/v1/kyc/register ***(for JPY)***

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	Parameters Name	Required	Description
1	user_id	Y	Enter a user name (English or Japanese).
2	username_katakana	Y	Enter a user name with Japanese format

combine all of the parameters above into one string, separate by "&" character and then encrypt it using encrypt_decrypt algorithms.

For example :

$str = "user_id=xxx&username_katakana=xxx"

Encryption sample :

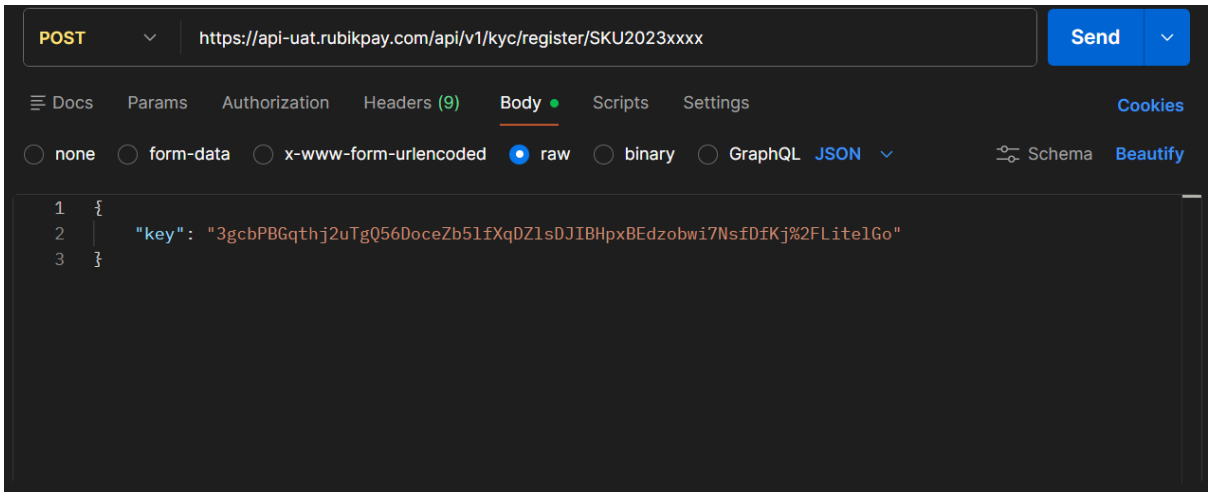
$key = encrypt_decrypt('encrypt',$str, '{your_api_key}', '{your_secret_key}')

Encryption result example :

"YAyMkOrskqFu%2FtF6suEgjE0IK4vfkqPFm8yGL4YT2b9WoaOBBzCC%2BqnoYsZJ3%2F3MssB7mUhkrEVRoqmX6pwwkeWWa%2BdMqEo0dxShDIYrJnDW3HkmgpD7HADDnQwxd2xAAzycB0%2BGoTzG9m08UeYD0cQqlpli0auTXFir%2FowlBcPJ48NKC9396Qfhi2EgiDYtuPwc9Ffc1HpVlRBo1nHNl5j30vArkvg0zmk53Al1XPPlw%2BikZuUxoNQ8REIDZTaUBq171nu3Ynly%2BU%2F0VYVc1Aiq3y8CIDylguHX2WcKZCXo0k7tiVaT%2BYjUUpyyvlEwwhmxOeLtik3xxK1SHW7bri22NVJgC1wlzforohWLA8LEyTcb7%2Ff%2B3zlMrpysT7ZSosmuZRNK3mPneylASOvYm9Rhs9CM6szCifN6QmLtXpjB648fM75%2FGrdw0NXp9QYd"

Refer to point 6 for the encryption/decryption function explanation.

Request example :



Response example	**Status: 200**

{

"status": "success",

"message": "Processing KYC",

"data": {
"user_id" : " example",
"status" :"pending"

}

}

31. CHECK STATUS KYC API

{base_url}/api/v1/kyc/status ***(for JPY)***

- base_url : Provided by provider
- Method : POST
- Merchant_code : Provided by provider
- Key : Generated by encryption from API key and API secret. The key contains these parameters:

No	**Parameters Name**	**Required**	**Description**

1	user_id	Y	Enter a user name (English or Japanese).

combine all of the parameters above into one string, separate by “&” character and then encrypt it using encrypt_decrypt algorithms.

For example :

$str = “user_id=xxx”

Encryption sample :

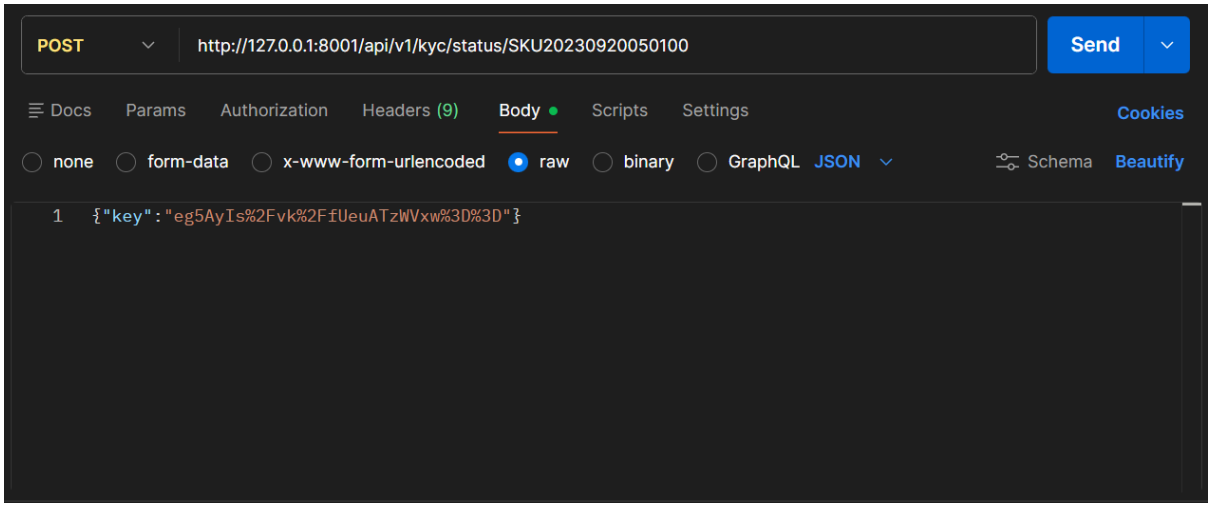
$key = encrypt_decrypt(‘encrypt’,$str, ‘{your_api_key}’, ‘{your_secret_key}’)

Encryption result example :

“YAyMkOrskqFu%2FtF6suEgjE0IK4vfkqPFm8yGL4YT2b9WoaOBBzCC%2BqnoYsZJ3%2F3MssB7mUhkrEVRoqmX6pwwkeWWa%2BdMqEo0dxShDIYrJnDW3HkmgpD7HADDnQwxd2xAAzycB0%2BGoTzG9m08UeYD0cQqlpli0auTXFir%2FowlBcPJ48NKC9396Qfhi2EgiDYtuPwc9Ffc1HpVlRBo1nHNl5j30vArkvg0zmk53Al1XPPlw%2BikZuUxoNQ8REIDZTaUBq171nu3Ynly%2BU%2F0VYVc1Aiq3y8CIDylguHX2WcKZCXo0k7tiVaT%2BYjUUpyyvlEwwhmxOeLtik3xxK1SHW7bri22NVJgC1wlzforohWLA8LEyTcb7%2Ff%2B3zlMrpysT7ZSosmuZRNK3mPneylASOvYm9Rhs9CM6szCifN6QmLtXpjB648fM75%2FGrdw0NXp9QYd”

Refer to point 6 for the encryption/decryption function explanation.

Request example :



Response example	**Status: 200**
{	

```
    "status": "success",
    "message": "KYC status polled successfully",
    "data": {
        "user_id" : " example",
        "status" :"approved"
    }
}
```