UBANK CONNECT

UBANK CONNECT

API Integration Guide

Version: 1.3

Table of Content

REVISION HISTORY

VERSION	DESCRIPTION
1.0	Initial Version
1.1	Encryption and Decryption
1.2	Update payout diagram
1.3	Status API

INTRODUCTION

The purpose of this document is to facilitate the integration of the Merchant Back Office with the PSP Back Office by the merchant’s developers.

DEFINITION AND TERMS

TERMS	DESCRIPTION
Merchant Website	Merchant can register and login on the website
Merchant Dashboard	Merchants can access their account details, transaction details(referring to their Deposits & Payouts) settlements and all related reports.
Merchant ID	This will be the merchant ID provided by PSP
Secret Key	Token given to the merchant that is used for encryption
Payout	The process of sending money out.
Settlement	Settlement process referring to the merchant’s own money.
Encryption IV	This will be the Encryption IV provided by PSP
Merchant	PSP Merchant
HASH	This is the combination of some parameter

PAYOUT FLOW

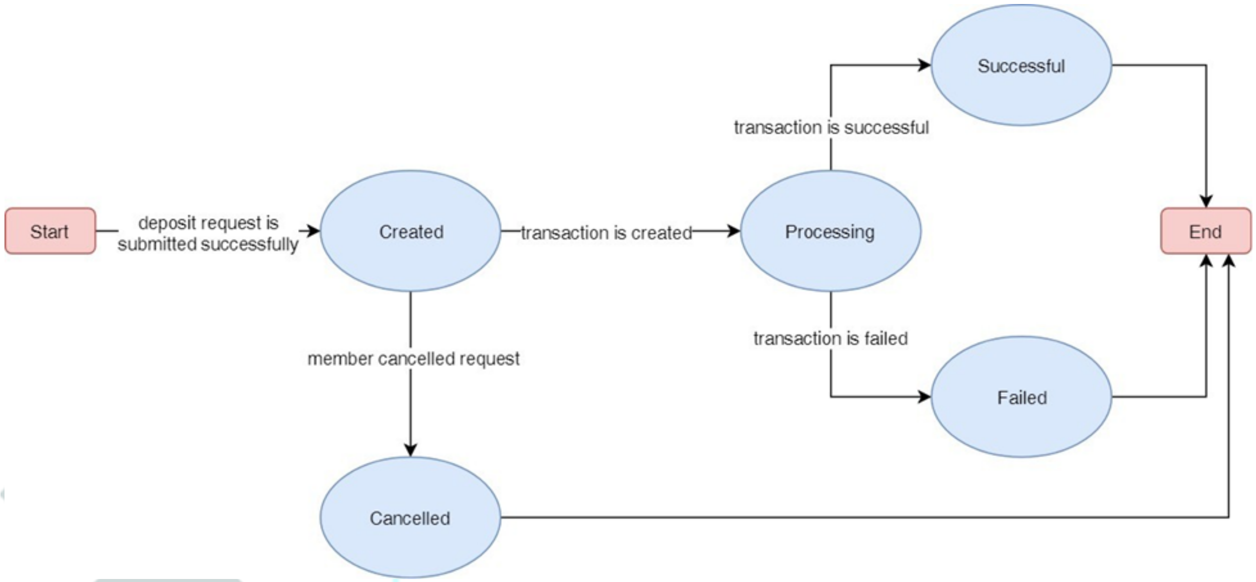
PSP supports 2 approaches in creating payout transactions: The first approach is to upload batch payouts in the PSP Merchant Site. The second one is to send payout requests from the Merchant Back Office. The latter approach is faster, since the process is automated.

Merchant Back Office to the PSP Back Office

Transactions are initiated from the Merchant Back Office. The first step is called Transaction Submission. In this step, the Merchant Back Office sends payout transaction details to the PSP Back Office.

Transaction Status

Transactions statuses will be reported to the Merchant Back Office via Deposit & Payout Callback API. Members will be redirected to RedirectURL (which is included in the API request parameters) after all processes are done.



Payout Request

Using below parameters in a CURL request the merchant will send the transaction requests to the PSP then the PSP will verify & validate the request and proceed with payout.

This API is an HTTP Form Submission using HTTP POST method. The URL will be sent to the merchant via email together with this document.

Payout API is the API used for Merchant Back Office to submit payout transactions to the PSP Back Office. The returned status is only used to know if the request has been submitted successfully.

PARAMETER	TYPE	MAX LENGTH	DESCRIPTION
X-API-KEY *Header*	String	-	BANKCONNECT_1890
Authorization *Header*	-	-	Merchant need to enter the basic auth credentials as follows: 1. User name- bankconnect_1.2_1890 2. Password- connectbank_1.2_1890
User-token *Header*	-	-	Merchant can create their user token using encryption in base64 using their credentials with concrete resolution operator as given below: **merchantid::secretkey**
end_point_url	URL	256	Merchant will send their end point URL to PSP.
Customer_order_id *Optional*	String	20	Merchant can send their customer order ID to differentiate transactions of their sub-merchants
enc_payout_json	-	-	This is the combination of below parameters as follows: [{"order_id":"93893832223","bank":"0","trx_type":"NEFT","payeename":"Test","bnf_nick_name":"test","amount":"1.00","account_no":"893338378373","ifsc":"SBINXXXXXXX","address1":"madanpur","city":"saritavihar","state":"delhi","pincode":"110076","email":"m@gmail.com","phone":"9821593611"}]
order_id	Integer	20	Order ID in Merchant Back Office
bank	Integer	0	It will be zero in all the cases
payeename	Text	20	Merchant will pass the name of Payee.
bnf_nick_name	Text	20	Merchant will pass the nickname of payee.

amount	Numeric	10	Merchant will pass the amount of the payout.
account_no	Numeric	18	Merchant will pass the account of the payee.
ifsc	String	11	Merchant will pass the IFSC code of the bank of the payee.
trx_type	Text	4	Merchant will pass the modes of payout i.e. IMPS, RTGS, & NEFT.
address1	String	10	Merchant will pass the address of the payee.
city	Text	10	Merchant will pass the name of the city.
state	Text	10	Merchant will pass the name of the state.
pincode	Numeric	6	Merchant will pass the pincode/zipcode.
phone	numeric	10	Merchant will pass the phone no of the payee.

Encrypted Payout Request Example

```
curl --location --request POST
'https://www.bankconnect.online/apisecure/payoutTransactionSecureIV' \
--header 'X-API-KEY: BANKCONNECT_1890' \
--header 'User-token: tZUEzdTVZeEhN==' \
--header 'Authorization: Basic YWtvbnRvX2FwaV92ZXJzaW9uXzE6Y2hlY2tvdXRfdmVyc2lvbl8x' \
--form 'end_point_url="enter_your_webhook_url"' \
--form
'enc_payout_json="RXg4SS9hck4vNUxwUk5MWVlTYnhMelQ2YnFoTjV0RWdBSWFlWXZldXJ3dW
N0UUZzK3ZRS3JHVjZSaUZVaVdlV2dwQWdmYkpDZC9XdU12cU1NQ1oyaVBPcXJReWdrMW9ka1Qz
REszSjZzZ3V5TVBrcm5BU3VXMjRNZFhocXdPQmFBRnFrSytEQlZpZExQWis4amZPZnlaSWt3SXZnUk
JQV0dDbHkrUnI1bFg2aG5PUFFNS2c1SXVaVUI0TU43QWVrRGdiTm1oOUg3QzVVRHJFUis3WnpCQ
2tiVHNrMjFkclg5Q05CblhxRmhFVW1LTlRPcXNPWXE2elRjK1RTYlI3VnZvYWprdmgxM3kzT0pXKzd3
ckJ4Nm96TlVvcjdWV0FnMlZZVS9ZcU1XSG82VkNWZ0JPNCtQcHpDUU9NOVAyK2o3eXgyNjUrazJV
am92bDhUQ00wT05EY2x1Q0x0VmEyOHhKbEQ4ajd4V0Y2a1hDeGRHMUhpSlhKMlRJRmZmMFdU
dDdYbDNaTWlRTlBlSmxYdUxTUTBhc0ZRNFgwcCs2bzVoU1h3UDg5Q0N0U3IwY0l6TkZWTG9QYVdj
KzI3WXl4aWFJeTJ2dFEyRzgya0lScysxUmF4L2NuQzBJY0FPWmQ3V0xjek5ONmRnRE9RNzI0OU5Vc
3ExZz09"'
```

Decrypted Payout Request Example

```
[{"order_id":"93893832223","bank":"0","trx_type":"NEFT","payeename":"Test","bnf_nick_name":"test","amount":"1.00","account_no":"893338378373","ifsc":"SBINXXXXXXX","address1":"madanpur","city":"saritavihar","state":"delhi","pincode":"110076","email":"m@gmail.com","phone":"9821593611"}]
```

Decrypted Payout Response Example

```
[{"status":"SUCCESS","order_id":"93893832223","payment_code":"SUCC200","message":"","utr":"39389383983"}]
```

The following table presents a complete list of possible statuses that are returned to the Merchant Back Office:

Status Codes

CODES	DESCRIPTION
SUCC200	Transaction has been submitted successfully.
SUCC201	Transaction is being Processed
SUCC202	Transaction is Failed
ERR411	Duplicate Order ID

Encryption

ENCODE your string to BASE64
CIPHERING type is AES-128-CTR
Bitwise disjunction of the flags is 0 always
Encryption IV will be provided by the Bankconnect
Encryption_key id MD5 format of your merchant id and secret like
MD5(merchant_no+mechant_secret)
Use openssl_encrypt for encryption
Final encryption is BASE64 of encryption value.

Decryption

Encryption IV will be provided by the Bankconnect
Encryption_key id MD5 format of your merchant id and secret like
MD5(merchant_no+mechant_secret)
Bitwise disjunction of the flags is 0 always
CIPHERING type is AES-128-CTR
DECODE encrypted data into BASE64
Use openssl_decrypt for decryption
Final decryption is BASE64 of decrypted value.

Payout Transaction Callback API

```
POST /apisecure/payoutTransactionStatus HTTP
Host: www.bankconnect.online
X-API-KEY: BANKCONNECT_1890
Authorization: Basic YWtvbnRvX2FwaV92ZXJzaW9uXzE6Y2hlY2tvdXRfdmVyc2lvbl8x
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="merchant_no"

0035
------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="secretkey"
```

```
35kjfkh45
------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="order_id"

93893832223
------WebKitFormBoundary7MA4YWxkTrZu0gW--
```