Contents

Document version 1.1

1. Overview

This document describes the steps for technical integration process between merchant website /application and Gateway.

Disbursement can be defined as “payment of money from a fund”.

Through this process a merchant can make payments to their Sub Merchants / Vendors for the goods and services used.

2. Fund Transfer API

Fund transfer API is used to make payments by merchant through disbursement model. The prerequisite requirement to successfully make payment is that merchant should maintain a sufficient disbursement fund with Gateway.

2.1. Fund Transfer Request API

To send the Fund Transfer Request use the below mentioned URL:

https://pgbiz.sabiopg.in/v3/fundtransfer

Parameters to be posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	Gateway would assign a unique 36-digit merchant key to you. This key is exclusive to your business/login account. If you have multiple login accounts, there will necessarily be one different api_key per login account that is assigned to you.	varchar(36)	Mandatory
merchant_reference_number	This is similar as order id it should be unique for every fund transfer	varchar(30)	Mandatory
Amount	Value of funds which are being transferred	decimal(10,2)	Mandatory
Hash	You need to compute a hash of all your parameters and pass that hash in API calls, for details about calculating hash refer Chapter 3.	varchar(255)	Mandatory

	Note: the SALT will be provided separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package		
transfer_type	Whether the transfer has to be made via NEFT or IMPS. For amount above 2 lakh, NEFT will be used	varchar(4)	Mandatory
account_name	Account holder name of the bank account of payee.	varchar(50)	Optional
account_number	Account number of the bank account of payee	varchar(50)	Mandatory
ifsc_code	IFSC code of the bank account of payee	varchar(50)	Mandatory
bank_name	Bank name of the bank account of payee.	varchar(50)	Optional
bank_branch	Bank branch of the bank account of payee	varchar(50)	Optional

The fund transfer request parameter will be in jquery format as shown below:

```
form.append("api_key", "f14e50fd-82f0-4ce0-bd4e-de924908d4ff");
form.append("merchant_reference_number", "124");
form.append("amount", "1000");
form.append("account_name", "1000");
form.append("account_number", "1000");
form.append("ifsc_code", "1000");
form.append("bank_name", "1000");
form.append("hash", "AAAAAAAAAAAAAAAAAAA77AAABCSDKANSKAKSBFKKASKAKS");
form.append("transfer_type", "NEFT");
```

On successful call to this API the response posted in jquery format will be as shown below:

```
{
  "data": {
    "status": "PROCESSING",
    "merchant_reference_number": "124",
    "transaction_id": "NEFT3057381244"
  }
}
```

If the fund transfer is terminated from bank end because of network issue or server issue etc. then the response posted in jquery format will be as shown below with an error code of 1029:

```
{
{
  "error": {
    "code": 1029,
    "message": "Transaction terminated"
  }
}
```

2.2. Fund Transfer Status API

We will provide an API which you can be used to check the status of any prior fund transfer transaction. You can use this to reconcile transactions. We recommend that you make it a practice to use this for every fund transfer transaction that was made. This serves two purposes:

- The response might not reach you due to network issues or server issue from bank end.

- This also works as a security check against any tampering, i.e. a second fallback check.

URL: https://pgbiz.sabiopg.in/v3/fundtransferstatus

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	Gateway would assign a unique 36-digit merchant key to you. This key is exclusive to your business/login account. If you have multiple login accounts, there will necessarily be one different api_key per login account that is assigned to you.	varchar(40)	Mandatory
merchant_reference_number	This is similar as order id it should be unique for every fund transfer	varchar(30)	Mandatory
hash	You need to compute a hash of all your parameters and pass that hash in API calls, for details about calculating hash refer Chapter 3. Note: the SALT will be provided separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package	varchar(255)	Mandatory

The Fund Transfer Status API parameter will be in jquery format as shown below:

```
form.append("api_key", "f14e50fd-82f0-4ce0-bd4e-de924908d4ff");
form.append("merchant_reference_number", "124");
form.append("hash", "AAAAAAAAAAAAAAAAAAAA77AAABCSDKANSKAKSBFKKASKAKS");
```

On successful call to this API the response posted in jquery format will be as shown below:

```
{
  "data": {
    "status": "SUCCESS",
    "merchant_reference_number": "124",
    "transaction_id": "NEFT3057381244"
  }
}
```

(Note: Transaction ID: Is a Unique Reference ID generated from bank for every fund transfer transaction to identify, reconcile, settle the same.)

If the merchant_reference_number sent in fund transfer status request is incorrect then the response posted in jquery format will be as shown below with an error code 1028.

```
{
  "error": {
    "code": 1028,
    "message": "Transaction not found"
  }
}
```

2.3. Fund Transfer Get Balance API

To check the available /current balance of their disbursement fund merchant have to send this API request. Merchant only has to send the API key and hash in the message for balance enquiry.

URL : https://pgbiz.sabiopg.in/v3/fundtransfer/getbalance

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	Gateway would assign a unique 36-digit merchant key to you. This key is exclusive to your business/login account. If you have multiple login accounts, there will necessarily be one different api_key per login account that is assigned to you.	varchar(40)	Mandatory
hash	You need to compute a hash of all your parameters and pass that hash in API calls, for details about calculating hash refer Chapter 3. Note: the SALT will be provided separately. NEVER PASS SALT IN A FORM,	varchar(255)	Mandatory

The Fund Transfer Get Balance API parameter will be in jquery format as shown below:

```
form.append("api_key", "f14e50fd-82f0-4ce0-bd4e-de924908d4ff");
form.append("hash",
"A80A837179AC1424E2F6955B7D1E30801C63405515D8F922AB608BD41F44E78F14001
02328F046A83C82137E182F666B0D47069A155651C009633A8BA5F576EC");
```

On successful call to this API the response posted in jquery format will be as shown below:

```
{
  "data":  {
    "balance": "20.00"
  }
}
```

If there is a HASH mismatch error then the response in jquery format will be as shown below with error code 1023:

```
{
  "error": {
    "code": 1023,
    "message": "Hash Mismatch"
  }
}
```

2.4. Fund Loading account whitelisting for merchant

For whitelisting accounts which can load funds to the account, below api is used. Once whitelisted, only the whitelisted accounts can load funds to the ecollect account

URL : https://pgbiz.sabiopg.in/v3/fundtransferwhitelisting

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	Gateway would assign a unique 36-digit merchant key to you. This key is exclusive to your business/login account. If you have multiple login accounts, there will necessarily be one different api_key per login account that is assigned to you.	varchar(40)	Mandatory
hash	You need to compute a hash of all your parameters and pass that hash in API calls, for details about calculating hash refer Chapter 3. Note: the SALT will be provided separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package	varchar(255)	Mandatory
account_number	Account number of the merchant from which funds will be loaded to account	varchar(20)	Mandatory
ifsc_code	IFSC code of the source account	varchar(11)	Mandatory

2.5. Fund Loading account whitelisting for team

For whitelisting accounts which can load funds across all the merchants, below api is used. Once whitelisted, only the whitelisted accounts can load funds to the ecollect accounts

URL : https://pgbiz.sabiopg.in/v3/fundtransferwhitelistingteam

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	Gateway would assign a unique 36-digit merchant key to you. This key is exclusive to your business/login account. If you have multiple login accounts, there will necessarily be one different api_key per login account that is assigned to you.	varchar(40)	Mandatory
Hash	You need to compute a hash of all your parameters and pass that hash in API calls, for details about calculating hash refer Chapter 3. Note: the SALT will be provided separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package	varchar(255)	Mandatory

account_number	Account number of the merchant from which funds will be loaded to account	varchar(20)	Mandatory
ifsc_code	IFSC code of the source account	varchar(11)	Mandatory

2.6. Get Statement API

To check all the transfers done in and out of a particular merchant have to send this API request. Merchant only has to send the API key , date range and hash in the message for getting the statement.

URL : https://pgbiz.sabiopg.in/v3/getstatement

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	Gateway would assign a unique 36-digit merchant key to you. This key is exclusive to your business/login account. If you have multiple login accounts, there will necessarily be one different api_key per login account that is assigned to you.	varchar(40)	Mandatory
Hash	You need to compute a hash of all your parameters and pass that hash in API calls, for details about calculating hash refer Chapter 3. Note: the SALT will be provided separately.	varchar(255)	Mandatory

	NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package		
date_from	The start date from which you need to retrieve information. This needs to be passed in DD-MM-YYYY format.	varchar(10)	Mandatory
date_to	The end date at which you need to retrieve information. This needs to be passed in DD-MM-YYYY format.	varchar(10)	Mandatory

On successful call to this API the response posted in jquery format will be as shown below:

```
{
  "data":  {
    "balance": "20.00"
  }
}
```

3.Hash calculation guide

4.1 How to Calculate Hash on API request

To calculate hash, you will need the salt provided at the time of onbarding.

Hashing generation algorithm

Following are the steps to calculate hash.

1. Create a | (pipe) delimited string called hash_data with first value as the salt.
2. Now sort the post fields based on their keys and create a | delimited string, for the fields with values.
3. Hash the hash_data string using SHA512 hashing algorithm and save the hash in secure_hash string
4. Convert the secure_hash string to upper case

Example PHP code to generate hash

```php
/**
 * @param array $parameters
 * @param string $salt
 * @param string $hashing_method
 * @return null|string
 */
function generateHashKey($parameters, $salt, $hashing_method = 'sha512')
{
    $secure_hash = null;
    ksort($parameters);
    $hash_data = $salt;
    foreach ($parameters as $key => $value) {
        if (strlen($value) > 0) {
            $hash_data .= '|' . trim($value);
        }
    }

    if (strlen($hash_data) > 0) {
        $secure_hash = strtoupper(hash($hashing_method, $hash_data));
    }

    return $secure_hash;
}
```

Example PHP code to check hash if response is JSON

```php
/**
 * @param $salt
 * @param $response_json
 * @return bool
 */
function responseHashCheck($salt, $response_array)
{

    /* If hash field is null no need to check hash for such response */
    if (is_null($response_array['hash'])) {
        return true;
    }

    $response_hash = $response_array['hash'];
    unset($response_array['hash']);
    $response_json = json_encode($response_array);

    /* Now we have response json without the hash */
    $calculated_hash = hashCalculate($salt, $response_json);

    return ($response_hash == $calculated_hash) ? true : false;
}

/**
 * @param $salt
 * @param $input_json
 * @return string
 */
function hashCalculate($salt, $input_json)
{

    /* Prepend salt with input json and calculate the hash using SHA512 */
    $hash_data = $salt . $input_json;
    $hash = strtoupper(hash('sha512', $hash_data));

    return $hash;
}
```

4. List of error codes

error numeric code	error code	error description
0	SUCCESS	Transaction successful
1023	Hash Mismatch	Hash Mismatch
1028	Transaction Not Found	If the Merchant reference number (order id) sent in request is invalid.
1029	Transaction Terminated	Transaction Failed to process due to network, server etc. issues from bank end

5. List of Disbursement Status

This section describes the various status given by the status api

Eg:

```
{"data":{"status":"PROCESSING","merchant_reference_number":"398107415993782272","transaction_id":null,"beneficiary_name":null,"error_message":null}}
```

Status Code	**Status description**
PROCESSING INCOMPLETE PENDING	Disbursement In progress. Status query to be run till it becomes SUCCESS or FAILURE
SENT_TO_BENEFICIARY	Disbursement is successfully initiated from source bank, but not received response from the customer's bank. Status query to be run till it becomes SUCCESS or FAILURE
FAILED FAILURE	Disbursement is failed due to the reason given in error code field
SUCCESS	Disbursement is successful
RETURNED_FROM_BENEFICIARY	Disbursement is initiated from source bank, but rejected by the customer's bank

6. Server to server callback

Once the transfer is completed, s2s can be send to the url defined against the merchant. Below is a sample response send:

```
{
  "transaction_reference_number":"3a7bd60e92344ecb45a0a0047330000",
  "status":"SUCCESS",
  "error_message":null,
  "bank_reference_number": 207517810965,
  "merchant_reference_number":"C8V8o5knTyW2439y6nuiDDBWby",
  "transaction_date":"2022-03-08 15:11:12",
  "transaction_amount":"1.00",
  "api_key":"f14e50fd-82f0-4ce0-bd4e-de924908d4ff",

"hash":"27FFA306D43ABC7553E4D55897012469E67C67CF0EC32B5F5A1480C297AD249E3EA00A4E65F4
C7D4692857496CD0BDBBB92B2DB95AC1E8B894F9D9541218A7B9"
}
```

7. Credit Notification

The ecollect_account_number can be shared to customers who can make transfers to the account. Once payment is received in these accounts, s2s post can be made to the url defined in the dashboard against the merchant. Below is a sample post for the e-collection :

```
 {
  "notify": {
          "status": "CREDITED",
          "ecollect_accout_no": "333112261000",
          "bank_reference_number": "205644248481",
           "transfer_type": "UPI",
          "transaction_date": "2022-02-25 18:13:19",
          "transaction_amount": 1,
          "remitter_name": "AKSHAY KUMAR",
          "remitter_account_number": "00001220345016666",
          "remitter_ifsc_code": "SBIN0014000",
          "remitter_note": "NA",
          "api_key": "766846de-90aa-4e79-a45c-9495a84ff6c2",
          "hash":"9B28C03820ED2AB17990AE64ADD4080A66DEAFB80ED1AC097538D1C29096F1696B9A
07662259D2F6 0DACDEDB1FF7EEF4DB8E5388806883F662DA8AE7533CF18D"
          }
}
```