PAYMENT GATEWAY

APIs for Disbursement

Contact

Website: https://zenpay.biz

Document version 2.0

Contents

1. Overview 2
2. Fund Transfer API 3
2.1. Fund Transfer Request API 3
2.2. Fund Transfer Status API 6
2.3. Fund Transfer Get Balance API 9
2.4. Account Status Check API 10
3. Hash calculation guide 12
3.1. How to Calculate Hash on API request 12
Hashing generation algorithm 12
Example PHP code to generate hash 13
3.2. How to check the response Hash 13
Hash checking algorithm 14
Example PHP code to check hash 14
Example PHP code to check hash if response is JSON 15

1. Overview

This document describes the steps for Technical integration process between merchant website and ZenPay for disbursement model.

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

This is like having a prepaid sim where to make calls one has to maintain a balance, similarly by keeping a reserve money with ZenPay PG through the API request merchants can pay their bills anytime independent of settlement money, settlement time frame etc.

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 have added the payee as vendor in ZenPay PG biz environment and should maintain a sufficient disbursement fund with ZenPay PG.

2.1. Fund Transfer Request API

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

https://pay.zenpay.biz/v3/fundtransfer

Parameters to be posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	ZenPay would assign a unique 40-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	varchar(40)	Mandatory

	that is assigned to you.		
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 to ZenPay Note: the SALT will be provided by ZenPay separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package.	varchar(255)	Mandatory
transfer_type	Whether the transfer has to be made via NEFT or IMPS. For amount above 2 lakh, NEFT will be used. IMPS is the default if no value passed	varchar(4)	Optional. Allowed values : NEFT,IMPS,RTGS
account_name	Account holder name of the bank account of payee. Optional if UPI details are given.	varchar(50)	Optional
account_number	Account number of the bank account of payee. Optional if	varchar(50)	optional

	UPI details are given.		
ifsc_code	IFSC code of the bank account of payee. Optional if UPI details are given.	varchar(50)	optional
bank_name	Bank name of the bank account of payee. Optional if UPI details are given.	varchar(50)	optional
bank_branch	Bank branch of the bank account of payee. Optional if UPI details are given.	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

ZenPay provides 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://pay.zenpay.biz/v3/fundtransferstatus

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	ZenPay would assign a unique 40-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 to ZenPay Note: the SALT will be provided by ZenPay 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", "AAAAAAAAAAAAAAAAAAA77AAABCSDKANSKAKSBFKKASKAKS");
```

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://pay.zenpay.biz/v3/fundtransfer/getbalance

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	ZenPay would assign a unique 40-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 to ZenPay Note: the SALT will be provided by ZenPay 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 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 jquerry format will be as shown below with error code 1023:

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

2.4. Account Status Check API

ZenPay provides an API which you can be used to check the status of any account. This helps to check if the account details provided are valid before making an actual fund transfer

URL: https://pay.zenpay.biz/v2/fundtransfer/validateaccount

Parameters to be Posted

Parameter Name	Description	Data type	Optional / Mandatory
api_key	ZenPay would assign a unique 40-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
account_name	The account holder's name	varchar(30)	Optional
account_number	The account number entered by user	varchar(30)	Mandatory
ifsc_code	The IFSC code of bank to be checked	varchar(30)	Mandatory
hash	You need to compute a hash of all your parameters and pass that hash to ZenPay Note: the SALT will be provided by ZenPay separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package.	varchar(255)	Mandatory

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

```
{
  "data": {
    "status": "SUCCESS",
    " beneficiary_name": " Mr  AMOL SHIVAJI BAD"
  }
}
```

If the details sent in Account Status Check request is incorrect then the response posted in jquery format will be as shown below with an error code 1028.

```
{
  "error": {
     "status": "FAILED"
  }
}
```

3. Hash calculation guide

3.1. How to Calculate Hash on API request

To calculate hash, you will need the salt provided by ZenPay.

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;
}
```

3.2. How to check the response Hash

It is important to make sure the response received from ZenPay is genuine, and to do so you will need to do a hash check on your server on receiving the response.

Every response received has a field called hash. Sometimes it is null, which means it in not important to check hash for the response, but if there is a hash present please perform hash check as described below and make sure integrity of the response received from ZenPay APIs.

To check hash, you will need the salt provided by ZenPay.

Hash checking algorithm

Example PHP code to check hash

```php
/**
 * @param string $salt
 * @param array $response_array
 * @return bool
 */
function responseHashCheck($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']);

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

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

/**
 * @param string $salt
 * @param array $input
 * @return string
 */

function hashCalculate($salt, $input)
{
    /* Columns used for hash calculation, Donot add or remove values from $hash_columns
array */
    $hash_columns = array_keys($input);
    /*Sort the array before hashing*/
    sort($hash_columns);

    /*Create a | (pipe) separated string of all the $input values which are available
in $hash_columns*/
    $hash_data = $salt;
    foreach ($hash_columns as $column) {
        if (isset($input[$column])) {
```

```
            if (strlen($input[$column]) > 0) {
                $hash_data .= '|' . trim($input[$column]);
            }
        }
    }
    $hash = strtoupper(hash("sha512", $hash_data));

    return $hash;
}
```

Example PHP code to check hash if response is JSON

```
/**
 * @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;
}
```

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

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