



# Payment Gateway

APIs for integration

## Contents

|       |   |    |
|-------|---|----|
| 1     | OVERVIEW.....   | 3  |
| 2     | PAYMENT REQUEST API.....                                  | 4  |
| 2.1   | Steps for Integration.....                                | 4  |
| 2.2   | Parameters to be POSTed in Payment Request.....           | 4  |
| 2.3   | Response Parameters returned.....                         | 8  |
| 3     | ENCRYPTING / DECRYPTING PAYMENT REQUEST API.....          | 10 |
| 3.1   | Parameters to be POSTed in Encrypted Payment Request..... | 10 |
| 3.2   | Decryption of Response Parameters returned.....           | 12 |
| 4     | GET PAYMENT REQUEST URL (Two Step Integration).....       | 14 |
| 4.1   | Steps for Integration.....                                | 14 |
| 4.2   | Parameters to be posted in request.....                   | 15 |
| 4.3   | Successful Response Parameters returned.....              | 15 |
| 4.4   | Manually Expire a Payment Request URL.....                | 15 |
| 5     | FETCH INTENT URL PAYMENT REQUEST.....                     | 17 |
| 5.1   | Steps for Integration.....                                | 17 |
| 6     | PAYMENT STATUS API.....                                   | 18 |
| 6.1   | Parameters to be POSTed.....                              | 18 |
| 6.2   | Response Parameters.....                                  | 19 |
| 7     | REFUNDS APIs.....   | 24 |
| 7.1   | Refund request API.....                                   | 24 |
| 7.2   | Refund Status API.....                                    | 26 |
| 8     | SPLIT API.....  | 29 |
| 8.1   | Split Settlement API.....                                 | 29 |
| 8.1.1 | Split transaction before settlement API.....              | 29 |
| 9     | VENDOR APIs.....  | 31 |
| 9.1   | Add Vendor API.....                                       | 31 |
| 9.2   | Modify Vendor API.....                                    | 33 |
| 9.3   | Add Vendor Accounts API.....                              | 34 |
| 9.4   | Delete Vendor API.....                                    | 36 |
| 9.5   | Get Vendor API.....                                       | 37 |
| 10    | SETTLEMENT APIs.....                                      | 39 |

|        |  |    |
|--------|--|----|
| 10.1   | Get Settlements API .....                                | 39 |
| 10.1.1 | Parameters to be POSTed in Request .....                 | 39 |
| 10.2   | Get Settlement Details API.....                          | 40 |
| 10.2.1 | Parameters to be POSTed in Request .....                 | 41 |
| 11     | CHALLAN PAYMENT APIs .....                               | 43 |
| 11.1   | Request challan payment API .....                        | 43 |
| 11.2   | Request challan payment API url.....                     | 43 |
| 11.2.1 | Request parameters are as following: .....               | 44 |
| 11.2.2 | Response from this API will be in JSON format: .....     | 44 |
| 11.2.3 | List of error codes and corresponding messages: .....    | 45 |
| 12     | SERVER TO SERVER CALL BACK (WEB HOOKS).....              | 46 |
| 12.1   | Server to server response on Payment.....                | 46 |
| 12.2   | Server to server response on Settlement .....            | 47 |
| 13     | SEAMLESS PAYMENT REQUEST API .....                       | 48 |
| 13.1   | Steps for Integration .....                              | 48 |
| 13.2   | Parameters to be POSTed in Seamless Payment Request..... | 48 |
| 13.3   | Response Parameters .....                                | 50 |
| 14     | APPENDIX 1 - References .....                            | 52 |
| 15     | APPENDIX 2 - Hash calculation guide.....                 | 53 |
| 15.1   | How to Calculate Hash on API request .....               | 53 |
| 15.1.1 | Hashing generation algorithm .....                       | 53 |
| 15.2   | How to check the response Hash.....                      | 53 |
| 15.2.1 | Hash checking algorithm.....                             | 54 |
| 16     | APPENDIX 3 - List of bank codes .....                    | 56 |
| 17     | APPENDIX 4 - List of error codes.....                    | 65 |
| 18     | APPENDIX 5 – Currency Codes.....                         | 68 |

# 1 OVERVIEW

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

Through Payment-Gateway, your customers can make electronic payments through various payment modes such as:

- Credit cards
- Debit cards
- Net banking
- EMI
- Cash Cards/Wallets
- Mobile/web invoicing
- Integrated NEFT/RTGS
- Bank deposits
- Standing instruction on cards
- Customer account direct debit (e-NACH)
- UPI - Collect and Intent based
- BharatQR
- E-Collect

Payment-Gateway also offers you a web portal where you have access to all your prior transaction/payment details, settlement details, analytics, etc. You can also use this portal to create invoices singly or in bulk, set reminders, recurring billing, and many more features, manage your payables, vendor payments, set split ratios for vendor payments, process refunds, etc. Ask Payment-Gateway for management portal.

Following information will be required from Payment-Gateway for proceeding with integration, this information will be available in management portal.

|                |   |
|----------------|---|
| {pg_api_url}   | This is the domain url of Payment-Gateway APIs  |
| api_key        | This is the key required for api access   |
| salt           | This is like password to access APIs and used in HASH/CHECKSUM calculations in request and responses. (Keep it safely do not share it publicly) |
| encryption key | This is encryption key used in Payment request to encrypted the request data (Keep it safely do not share it publicly)                          |
| decryption_key | This is decryption key used for decrypting the payment response returned on return_url (Keep it safely do not share it publicly)                |

## 2 PAYMENT REQUEST API

When you integrate with Payment-Gateway, the customer will be re-directed from your merchant website to the Payment-Gateway payment page. After completion of the transaction, Payment-Gateway will direct the customer back to the merchant website

**URL:** *https://{pg\_api\_url}/v2/paymentrequest*

### 2.1 Steps for Integration

- You need to submit a **POST REQUEST** to our server, at the above-mentioned URL

**Note:** hash is a mandatory parameter. If your hash is not properly calculated or does not match for whatever reason, we will not be able to process the payment. The usage of hash is explained in subsequent sections.

- When you call this API, the customer is necessarily re-directed to Payment-Gateway's payment page. After the customer makes the payment through Payment-Gateway (entering his card details or netbanking details etc.), we direct the customer back to your merchant site.

**Note:** If you need the customer to enter credit card details on your (merchant) website and would NOT want us to redirect to the Payment-Gateway page, we can get that done, provided you are PCI-DSS certified. If you are not certified and would like to get certified, let us know. We will guide you appropriately on how to get it done.

- We recommend that you check the hash at your end again, after we send back the response to you. This is essential to prevent user data tampering fraud.
- Transaction ID and order ID:
  - When you submit your transaction request to Payment-Gateway, you need to submit an order ID as part of the request. This order ID can be used by you as a universal reference number for all transaction requests submitted by you.
  - When your customer clicks the "Pay" button on the payment page, a unique transaction ID is assigned to the transaction.
  - Order ID acts as a "merchant reference number". You must maintain uniqueness of your order IDs.

### 2.2 Parameters to be POSTed in Payment Request

**URL:** *https://{pg\_api\_url}/v2/paymentrequest*

| Parameter Name | Description | Data type | Optional / Mandatory |
|----------------|-------------|-----------|----------------------|
|----------------|-------------|-----------|----------------------|

|                     |   |               |           |
|---------------------|---|---------------|-----------|
| api_key             | Payment-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 |
| order_id            | This is your (merchant) reference number. It must be unique for every transaction. We do perform a validation at our end and do not allow duplicate order_ids for the same merchant.  | varchar(30)   | mandatory |
| mode                | This is the payment mode (TEST or LIVE are valid values)  | varchar(4)    | optional  |
| amount              | This is the payment amount.   | decimal(12,2) | mandatory |
| currency            | This is the 3-digit currency code (INR)   | varchar(3)    | mandatory |
| description         | Brief description of product or service that the customer is being charged for.   | varchar(255)  | mandatory |
| name                | Name of customer.   | varchar(255)  | mandatory |
| email               | Customer email address.   | varchar(255)  | mandatory |
| phone               | Customer phone number   | varchar(30)   | mandatory |
| address_line_1      | Customer address  | varchar(255)  | optional  |
| address_line_2      | Customer address 2  | varchar(255)  | optional  |
| city                | Customer city   | varchar(255)  | mandatory |
| state               | Customer State  | varchar(255)  | optional  |
| country             | Customer country  | varchar(100)  | mandatory |
| zip_code            | Customer zip code   | varchar(20)   | mandatory |
| timeout_duration    | Timeout duration (in seconds)   | varchar(10)   | optional  |
| udf1                | User defined field  | varchar(255)  | optional  |
| udf2                | User defined field 2  | varchar(255)  | optional  |
| udf3                | User defined field 3  | varchar(255)  | optional  |
| udf4                | User defined field 4  | varchar(255)  | optional  |
| udf5                | User defined field 5  | varchar(255)  | optional  |
| return_url          | Return URL success - Payment-Gateway will make a POST request to this URL after successful transaction, with a set of parameters, which you can process as you want to.   | varchar(255)  | mandatory |
| return_url_failure  | Return URL failure - Payment-Gateway will make a POST request to this URL after a FAILED transaction, with a set of parameters, which you can process as you want to.   | varchar(255)  | optional  |
| return_url_cancel   | Return URL success - Payment-Gateway will make a POST request to this URL in case of transaction cancellation, with a set of parameters, which you can process as you want to.  | varchar(255)  | optional  |
| percent_tdr_by_user | Percent of tdr amount paid by user (optional) (max value:100)   | decimal(5,2)  | optional  |

|                           |  |                |   |
|---------------------------|--|----------------|---|
| flatfee_tdr_by_user       | fixed fee paid by user (optional)  | decimal(10,2 ) | optional  |
| show_convenience_fee      | Controls whether the convenience fee amount (for surcharge merchants) is displayed to the customer (on the payment page) or not  | varchar(1)     | optional  |
| split_enforce_strict      | Controls whether payment is required to be split before settlement. By default, it is set to 'n', If this is set to 'y' then settlement will be on HOLD until splitsettlement api is called to provide split information.  | varchar(1)     | Mandatory if settlement amount needs to be split with vendors. Optional Otherwise |
| split_info                | Split info is for splitting the payment between vendor and themselves. In this field one must provide vendor code and what is the amount of the payment to be made to vendors. Following is an example how it will look<br><i><b>{"vendors":[{"vendor_code":"2VEN449","split_amount_fixed":"20"}, {"vendor_code":"2VEN450","split_amount_fixed":"20"}]}</b></i><br>All field in this JSON are mandatory. | varchar(500)   | Mandatory if settlement amount needs to be split with vendors. Optional Otherwise |
| payment_options           | payment options to be displayed such credit card (cc), net banking (nb), wallet (w), ATM card (atm), UPI (upi) and debit card with pin (dp). Tabs will be displayed by order in which values are sent.<br><br>Values accepted are: <i>cc,nb,w,atm,upi,dp</i> (comma separated string), sequence of values will also determine the tab sequence on payment page.  | varchar(50)    | optional  |
| payment_page_display_text | This text will be displayed below the logo on payment page.  | varchar(100)   | optional  |
| allowed_bank_codes        | Bank codes sent in this filed will be allowed in payment page, other bank codes will not be allowed to proceed with payment. Refer appendix 3 for the list of bank codes. To send multiple bank codes send a comma separated list. E.g. to allow only credit cards: <i>MACC,VICC,DINC,VISC,RUPC,MASC,AMXC</i>  | varchar(250)   | optional  |
| allowed_emi_tenure        | This will be a comma separated integer list depending upon the tenure (in months) of loan allowed to show in EMI payment method. Ex( 3 , 6, 9 etc.)  | varchar(50)    | optional  |
| allowed_bins              | BIN is Bank Identification Number, on a card it is first 6 digits. BINs passed here will only be allowed to transact, multiple BINs can be sent as comma   | varchar(250)   | optional  |

|                    |   |                |           |
|--------------------|---|----------------|-----------|
|                    | separated list. Refer appendix 3 for the list if the payment mode is card   |                |           |
| offer_code         | If there is any discount / offer provided by merchant on EMIs, then predefined codes must be mentioned in this field. (This is for specific use case; more information can be provided on demand)   | varchar(100)   | optional  |
| emi_info           | This is an optional param which is to be posted to issuer end in case emi detail are required.the format will be as mentioned below<br><pre>"emi_info": {   "subvention": "0.0",   "aggregator_name": "BENOW",   "bank_merchant_id": null,   "bank_term_id": null,   "bank_sku_code": null }</pre> } all the fields are optional and have varchar as data type.   | varchar(100)   | optional  |
| product_details    | Contains information regarding the goods/product for which the payment (emi) is being made. Values in this field should be sent in JSON format, for example:<br><pre>{<b>"manufacturer": "Samsung", "category": "Phone", "sub_category_1": "Smart Phone", "sub_category_2": "High-end", "model_name": "Samsung Galaxy S10 Pro"}</b></pre> Fields such as manufacturer, category, model_name are self-explanatory; sub_category_1 and sub_category_2 further describe the variants/types of that product.<br>All fields in this JSON are optional. | varchar(2048 ) | optional  |
| enable_auto_refund | Payment request is auto refunded in case of delay success depending upon the value present in the field is 'y' or 'n'. If this field is not sent default set for your account at the time of setup will take effect.  | varchar(1)     | optional  |
| hash               | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b>  | varchar(255)   | mandatory |

## 2.3 Response Parameters returned

| Parameter name    | Description  |
|-------------------|--|
| transaction_id    | A unique ID that can be used to trace the transaction uniquely within Payment-Gateway. Transaction IDs are alphanumeric. An example transaction ID is HDVISC1299876438 |
| payment_mode      | This tells the payment mode used by customer - example: "credit card", "debit card", "netbanking", etc.  |
| payment_channel   | This tells the payment channel used by customer - example: "Visa", "HDFC Bank", "Paytm", etc.  |
| payment_datetime  | Date and Time of this payment in "DD-MM-YYYY HH:MM:SS" format  |
| response_code     | Status of the transaction (return code). 0 signifies successful transaction. Non-zero values signify error. Response Code returned is documented in Appendix 4         |
| response_message  | The response message associated with the transaction.  |
| error_desc        | The detailed error description, if any   |
| order_id          | The same order_id that was originally posted by the merchant in the request.   |
| amount            | The same original amount that was sent by the merchant in the transaction request. In case of customer surcharge model this will be the amount paid by customer **.    |
| currency          | This is the 3digit currency code (INR), it will be same value that was originally sent by merchant.  |
| description       | The same description that was originally sent by the merchant in the transaction request.  |
| name              | The same value that was originally sent by merchant  |
| email             | The same value that was originally sent by merchant  |
| phone             | The same value that was originally sent by merchant  |
| address_line_1    | The same value that was originally sent by merchant  |
| address_line_2    | The same value that was originally sent by merchant  |
| city              | The same value that was originally sent by merchant  |
| state             | The same value that was originally sent by merchant  |
| country           | The same value that was originally sent by merchant  |
| zip_code          | The same value that was originally sent by merchant  |
| udf1              | The same value that was originally sent by merchant  |
| udf2              | The same value that was originally sent by merchant  |
| udf3              | The same value that was originally sent by merchant  |
| udf4              | The same value that was originally sent by merchant  |
| udf5              | The same value that was originally sent by merchant  |
| tdr_amount        | This is the TDR charged on the transaction **  |
| tax_on_tdr_amount | This is the Tax (GST) charged on the TDR Amount **   |
| amount_orig       | This is the amount requested by merchant **. Typically, this will be same as the amount field, but in case of customer surcharge model this will be a different value. |
| cardmasked        | Masked card number which was used to make the transaction **. For example, 437748*****0069   |
| emi_tenure        | This field is returned if "send_emi_details" merchant param is enabled and/or in payment request emi_info param is set to get this in response                         |

|                      |   |
|----------------------|---|
| emi_rate_of_interest | This field is returned if "send_emi_details" merchant param is enabled and/or in payment request emi_info param is set to get this in response                                      |
| hash                 | Payment-Gateway calculates the hash using the same algorithm which was outlined earlier. Hence, the merchant needs to check whether this returned hash matches the calculated hash. |

**\*\* Note:** This parameter will be returned as part of the response **only** if the merchant's account has been enabled for the same. Please speak to your Payment-Gateway relationship manager if you would like this information to be returned in response.

**Note:** It is important to validate the hash after you receive the response from Payment-Gateway. A failed response sent from Payment-Gateway server to your server via browser could be tampered by a malicious end-user and turned into "success". To make sure the transaction response is the same as what Payment-Gateway server sent please check the hash before considering the transaction response as final.

**Note:** Format of transaction ID is as follows: HDVISC1299876438". The 3<sup>rd</sup> to 6<sup>th</sup> digits (both inclusive) in the transaction ID signify the "bankcode". This information is enough to obtain the payment method and payment channel. A list of bankcodes and corresponding payment mode/channel is available in Appendix 3 of this document.

### 3 ENCRYPTING / DECRYPTING PAYMENT REQUEST API

- This section focuses on encrypting a payment request that we built in previous section
- For encrypting and decrypting payment request, encryption and decryption keys will be provided by Payment-Gateway.
- Payment request is encrypted using AES-265-CBC algorithm with encryption key and then posted to payment-gateway as explained in following sections.
- Payment-Gateway will post the final response of the transaction in encrypted format on the return\_url provided in the request, this response needs to be decrypted using the decryption key as described in following sections and then processed further.

#### 3.1 Parameters to be POSTed in Encrypted Payment Request

**URL:** *https://{pg\_api\_url}/v2/paymentrequest*

| Parameter Name | Description   | Data type     | Optional / Mandatory |
|----------------|---|---------------|----------------------|
| api_key        | Payment-Gateway 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            |
| encrypted_data | This field will contain encrypted data (Following section explains how to prepare data and then encrypt it.)  | varchar(5000) | mandatory            |
| iv             | This field will contain initial vector that is used for encryption, it must be 16 byte long cryptographically secure random string.   | varchar(16)   | mandatory            |

1. **Preparation of data:** - First you need to create JSON formatted data and then encrypt the JSON string. Following is the structure of JSON. Details of each field is same as defined in payment request in previous section.

```
{  
  "api_key": "f14e50fd-82f0-4ce0-bd4e-de924908d4ff",  
  "address_line_1": "Address Line 1",  
  "address_line_2": "Address Line 2",  
  "amount": "35.56",  
  "city": "Bangalore",
```

```

"country": "IND",
"currency": "INR",
"description": "Payment Short Description",
"email": "amit@example.com",
"mode": "LIVE",
"name": "Amit",
"order_id": "XYZ419",
"phone": "9900990099",
"return_url": "https://yoursite.com/return_page_success.php",
"return_url_failure": "https://yoursite.com/return_page_failure.php",
"return_url_cancel": "https://yoursite.com/return_page_success.php",
"state": "Karnataka",
"udf1": "Users Defined Value",
"udf2": "",
"udf3": "",
"udf4": "",
"udf5": "",
"zip_code": "560001",
"hash":
"23CD8F59DB509F55951CE74C922879128098F4998C72BFD7400011C6B305290DD9333101C1C
C70CEB04587C49E6949AA864F83D54088F691FD57C7222CDB2572"
}

```

2. **Encrypting the JSON data:** - After encrypting the json data above result will be passed in encrypted\_data and iv fields, to encrypt we use 256-bit AES encryption, encryption method is AES-256-CBC, encrypted data should be base64 encoded before posting. Encryption key will be provided by Payment-Gateway. Note that both encrypted\_data and iv are base64 encoded, if not already done (some encryption libraries do not base 64 encode the resulting encryption and the randomly generated iv bytes)

Sample encryption code in PHP is provided below for your reference. you can easily get libraries in other languages (.NET, JAVA, Nodejs) to do AES-256-CBC encryption and base64 encoding

```

/**
 * AES-256-CBC Encryption method, First it encrypts the data and then encodes it to
 base64
 * @param string $plain_data Plain text Data to be encrypted
 * @param string $encryption_key Provided by Payment Gateway
 * @return array
 */
function encryptData($plain_data, $encryption_key)
{
    $iv = openssl_random_pseudo_bytes(16); //random 16 byte/char long string or unix-
timestamp(10chars)+random-string(6chars) time().mt_rand(100000, 999999)
    $encrypted = openssl_encrypt($plain_data, "AES-256-CBC", $encryption_key,
OPENSSL_RAW_DATA, $iv);

```

```

    return [base64_encode($encrypted), base64_encode($iv)];
}

```

## 3.2 Decryption of Response Parameters returned

In the response, you will receive AES-256-CBC encrypted data which needs to be decrypted. Following sections explain how to decrypt the encrypted data.

| Parameter Name | Description   |
|----------------|---|
| api_key        | Payment-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. |
| encrypted_data | This field will contain encrypted data (Following section explains how to decrypt and get the response data.)   |
| iv             | This field will contain initial vector use this to decrypt the encrypted_data received in response  |

**Decrypting the encrypted data:** - To decrypt we use 256 bit AES encryption algorithm, decryption method is AES-256-CBC, decrypted data should be base64 decoded before using. Decrypted data will be in JSON format as shown below. Decryption key will be provided by Payment-Gateway.

Sample decryption code in PHP, you can easily get libraries in other languages (.NET, JAVA, Nodejs) to do AES-256-CBC decryption and base64 decoding.

```

/**
 * AES-256-CBC Decryption method, First it decodes base64 encoded encrypted data and
 * then decrypts it to get JSON data string
 *
 * @param String $encrypted_data Encrypted data received in response
 * @param String $decryption_key Provided by Payment Gateway
 * @param String $iv Initial Vector iv as received in response
 */
function decryptData($encrypted_data, $decryption_key, $iv)
{
    return openssl_decrypt(base64_decode($encrypted_data), 'AES-256-CBC',
    $decryption_key, OPENSSSL_RAW_DATA, base64_decode($iv));
}

```

JSON response after decryption will be as following, .

```

{
  "transaction_id": "FBEEBN12311231231",
  "payment_mode": "NetBanking",
  "payment_channel": "Federal Bank",
  "payment_datetime": "2018-09-19 16:19:46",
  "response_code": 1043,
  "response_message": "Transaction cancelled",
}

```

```
"error_desc": "Transaction cancelled",
"order_id": "607",
"amount": "100.81",
"currency": "INR",
"description": "Payment Short Description",
"name": "Amit",
"email": "amit@example.com",
"phone": "9900990099",
"address_line_1": "Address Line 1",
"address_line_2": "Address Line 2",
"city": "Bangalore",
"state": "Karnataka",
"country": "IND",
"zip_code": "560001",
"udf1": null,
"udf2": null,
"udf3": null,
"udf4": null,
"udf5": null,
"hash":
"64948C7CA2F8861285A246738FF48347FCAEB81D7EBB0F2D00B6BE661EAC1DADCFD59A0FA878CE878
9F0DB9E045FC0943661A9DE4B5984F5D6E32DDEE1BBD4B6"
}
```

## 4 GET PAYMENT REQUEST URL (Two Step Integration)

Payment-Gateway provides an API which returns a unique payment page URL on your merchant server in response which can be used in any browser to show the payment selection page and complete the transaction. This process gets complete in two steps.

**URL:** `https://{pg_api_url}/v2/getpaymentrequesturl`

### 4.1 Steps for Integration

- First step is, you need to submit a **POST REQUEST** to API URL  
**Note:** hash is a mandatory parameter. If your hash is not properly calculated or does not match for whatever reason, we will not be able to process the payment. The usage of hash is explained in subsequent sections.
- In response you will get a payment execution URL. Following is the sample response message in case of success.

```
{
  "data": {
    "url": "https://{pg_api_url}/v2/executepaymentrequesturl/3c1943aa-13be-4866-925e-
d56c32c62d47",
    "uuid": "3c1943aa-13be-4866-925e-d56c32c62d47",
    "expiry_datetime": "2019-06-14 16:38:36",
    "order_id": "T103"
  }
}
```

- The response message apart from payment URL will contain a UUID (unique identification number for this transaction request), expiry date/time (this url will not work after the given expiry date/time) and order id (the one sent in the request by merchant). For every get payment URL request a unique UUID is generated in response.
- Second step is after getting the URL, open it in any browser or app webview for showing the payment method selection page to customer to complete the payment.

It is best to use this API if payments are going to be done through mobile apps for preventing frauds or hash / data tampering. The request can be built on your server and the unique URL can be sent to Client app, from where the payment process would continue.

## 4.2 Parameters to be posted in request

**URL:** *https://{pg\_api\_url}/v2/getpaymentrequesturl*

Parameters to be posted for this API are exactly same as in v2/paymentrequest API (see Section 2.2), except for one additional optional parameter for defining the url expiry in minutes

| Parameter Name    | Description   | Data type   | Optional / Mandatory |
|-------------------|---|-------------|----------------------|
| expiry_in_minutes | This field is to define the response url expiry in minutes. This field accepts integer value minimum of 15 (15 minutes) and maximum of 10080 (7 days) | varchar(10) | optional             |

## 4.3 Successful Response Parameters returned

| Parameter Name  | Description   |
|-----------------|---|
| url             | Payment URL, which can open in any browser  |
| expiry_datetime | The Payment URL expiration time, by default set for 15 mins   |
| uuid            | Unique ID generated for every request   |
| order_id        | This is your (merchant) reference number which you submitted while making the original transaction. |

## 4.4 Manually Expire a Payment Request URL

This API is used if the links created by getpaymentrequesturl API needs to be expired before it automatically expires as per expiry\_in\_minutes. After this API is successfully run then user prevented from making a transaction on calling */v2/executepaymentrequesturl/{uuid}* URL

**URL:** *https://{pg\_api\_url}/v2/expirepaymentrequesturl*

| Parameter Name | Description   | Data type   | Optional / Mandatory |
|----------------|---|-------------|----------------------|
| api_key        | Payment-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            |

|      |  |              |           |
|------|--|--------------|-----------|
| uuid | This is the uuid returned by the <code>getpaymentrequesturl</code> api   | varchar(40)  | mandatory |
| hash | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b> | varchar(255) | mandatory |

The expire payment request url parameter will be in **jquery** format as shown below:

```
form.append("api_key", "f14e50fd-82f0-4ce0-bd4e-de234sffd4ff");
form.append("uuid", "3c1943aa-13be-4866-925e-d56c32c62d47");
form.append("hash", "AAAAAAAAAAAAAAAAAAAAA77AAABCSDKANSKAKSBFKKASKAKS");
```

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

```
{
  "data": {
    "code": "SUCCESS",
    "message": "The URL is expired",
  }
}
```

If the transaction is already in progress, the API will return an error message

```
{
  {
    "error": {
      "code": "TRANSACTION-IN-PROGRESS",
      "message": " Your request cannot be processed"
    }
  }
}
```

## 5 FETCH INTENT URL PAYMENT REQUEST

Payment Gateway will provide an API which returns a unique payment page URL on your merchant server in response which can be used in any browser to show the payment selection page and complete the transaction. This process gets complete in two steps.

**URL:** *https://{pg\_api\_url}/v2/getpaymentrequestintenturl*

### 5.1 Steps for Integration

- First step is, you need to submit a POST REQUEST to API URL
- In response you will get a payment intent URL. Following is the sample response message in case of success.

```
{
  "data": {
    "upi_intent_url":
    "upi:VV/pay?pa=MERCUAT@bank&pn=Karmendra+Biz&am=13.45&mam=13.45&tr=6478789&tn=P
    ayment+for+6478789&mc=5021&mode=04&purpose=00&utm_campaign=DEBIT&utm_medium=
    PAYFAIRUAT&utm_source=6478789",
    "payment_request_id": 6478789,
    "order_id" : 123456
  }
}
```

- The response message contains the intent url, transaction id and order id sent in the payment request.
- Second step is after getting the URL is to use trigger an app from the respective mobile device.

It is best to use this API if payments are going to be done through mobile payment apps for preventing frauds or hash / data tampering.

## 6 PAYMENT STATUS API

Payment-Gateway provides an API which you can use to check the status of any prior transaction. You can use this to reconcile transactions. We strongly recommend that you make it a practice to use this for every transaction that was made. This serves two purposes:

- The response might not reach you due to network issues or other problems such as user clicking refresh button on their browser, etc.
- This also protects against any tampering, since you have a second fallback check here.

Payment-Gateway offers a sophisticated API wherein you can apply "filters" on the resultset you want to retrieve. You can search our system by the transaction ID, or the order ID, or even by parameters such as date range, customer phone number, etc. You can also pass in various combinations of these parameters to get the resultset of your choice.

**Note:** Your designated server IP will need to be whitelisted by Payment-Gateway for this API to work. If you receive errors such as "Unauthorized" while accessing this API, please contact your Payment-Gateway relationship manager to get this fixed.

**URL:** [https://{pg\\_api\\_url}/v2/paymentstatus](https://{pg_api_url}/v2/paymentstatus)

### 6.1 Parameters to be POSTed

| Parameter Name | Description   | Data type   | Optional / Mandatory |
|----------------|---|-------------|----------------------|
| api_key        | Payment-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            |
| order_id       | This is your (merchant) reference number which you submitted while making the original transaction. You can send multiple order ids in this field as comma (,) separated list   | varchar(30) | Optional             |
| transaction_id | This is the transaction ID generated by Payment-Gateway for the given transaction   | varchar(30) | Optional             |
| bank_code      | This is the 4-letter bankcode which denotes the payment mode/channel of the payment.  | varchar(4)  | Optional             |
| response_code  | The numeric response code returned by Payment-Gateway during the original transaction   | number(4)   | Optional             |

|                |  |              |           |
|----------------|--|--------------|-----------|
| customer_phone | Phone number of the customer, as provided during the original paymentrequest API   | varchar(30)  | Optional  |
| customer_email | Email address of the customer, as provided during the original paymentrequest API  | varchar(255) | Optional  |
| customer_name  | Name of the customer, as provided during the original paymentrequest API   | varchar(255) | Optional  |
| date_from      | Start date of date range to retrieve transactions, in DD-MM-YYYY or YYYY-MM-DD HH:MM:SS format   | varchar(20)  | Optional  |
| date_to        | End date of date range to retrieve transactions, in DD-MM-YYYY or YYYY-MM-DD HH:MM:SS format   | varchar(20)  | Optional  |
| page_number    | Page number you need to retrieve, its value is limited by information received in the first response that is received  | integer      | Optional  |
| per_page       | Number to records need to see per page, this value should be between 1 and 50  | integer      | Optional  |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b> | varchar(255) | Mandatory |

## 6.2 Response Parameters

On successful call to this API you will receive JSON response. You can read the JSON response and process it at your end. If your result set is greater than 50 transactions, you would need to use pagination.

**Note:** few parameters in response will be visible only if it is enabled for your account, for example: refund\_details will be available if it is enabled for your merchant account.

A few sample responses for given requests are provided below:

In case of success,

```
{
  "data": [
    {
      "transaction_id": "SFSBIN2783912661",
```

```

"bank_code": "SBIN",
"payment_mode": "Netbanking",
"payment_channel": "State Bank of India",
"payment_datetime": "2018-06-13 16:44:03",
"response_code": 1000,
"response_message": "FAILED",
"authorization_staus": null,
"order_id": "427641",
"amount": "27.36",
"amount_orig": "2.00",
"tdr_amount": 21.49,
"tax_on_tdr_amount": 3.87,
"description": "Web Payment for 433487",
"error_desc": "FAILED",
"customer_phone": "9900990099",
"customer_name": "sharathkumar hegde",
"customer_email": "sharathkumar@example.com"
"currency": "INR",
"cardmasked": null,
"udf1": null,
"udf2": null,
"udf3": null,
"udf4": null,
"udf5": null,
"refund_details": {
  "refund_amount": 0
}
},
{
"transaction_id": "HDVISC4291974106",
"bank_code": "VISC",
"payment_mode": "Credit Card",
"payment_channel": "Visa",
"payment_datetime": "2018-06-13 16:45:39",
"response_code": 0,
"response_message": "SUCCESS",
"authorization_staus": "captured",
"order_id": "427643",
"amount": "1.93",
"amount_orig": "1.90",
"tdr_amount": 0.03,
"tax_on_tdr_amount": 0,

```

```

    "description": "Web Payment for 433489",
    "error_desc": null,
    "customer_phone": "9900990099",
    "customer_name": "sharathkumar hegde",
    "customer_email": "sharathkumar@example.com"
    "currency": "INR",
    "cardmasked": null,
    "udf1": null,
    "udf2": null,
    "udf3": null,
    "udf4": null,
    "udf5": null,
    "refund_details": {
      "refund_amount": 0
    }
  }
],
"page": {
  "total": 175,
  "per_page": 10,
  "current_page": 1,
  "last_page": 18,
  "from": 1,
  "to": 10
},
"hash":
"30FAAD865191B4064576F063177F0A4692C3DBBBF35D1A20463EAA449269C4715FD13528EA069B3A8
D5C25C62637ED825C297C2337CDC1CFB7FCD0D60DCFE9D"
}

```

In case of error,

```

{
  "error": {
    "code": 1001,
    "message": "The api key field is incorrect"
  }
}

```

In case there is no record present in our system for the combination of input, following error is returned

```
{
  "error": {
    "code": 1050,
    "message": "No data record found for the given input"
  }
}
```

In case there is no transaction id in our system for the order\_id, merchant\_order\_id or transaction\_id, following error is returned

```
{
  "error": {
    "code": 1028,
    "message": "No Transaction found"
  }
}
```

If there are more than 50 transactions for which the status is requested, you would see following error

```
{
  "error": {
    "code": 1086,
    "message": "More than 50 records, refine your search criteria or use pagination"
  }
}
```

In such cases where result set is expected to have status of more than 50 transactions, it is required to use pagination.

Pagination allows to access data in smaller chunks making it easier for server to return data quickly.

To allow for pagination one need to pass following additional parameter in request

| Parameter Name | Description   | Data type | Optional / Mandatory |
|----------------|---|-----------|----------------------|
| page_number    | Page number you need to retrieve, its value is limited by information received in the first response that is received | integer   | optional             |
| per_page       | Number to records need to see per page, this value should be between 1 and 50   | integer   | optional             |

If above parameters are passed, response will have additional information about pagination as following

```
"page": {
  "total": 175,
  "per_page": 10,
```

```
"current_page": 1,  
"last_page": 18,  
"from": 1,  
"to": 10  
},
```

This pagination information should be used in `page_number` field for subsequent api request.

## 7 REFUNDS APIs

Payment-Gateway provides a refund API which merchants can use to programmatically issue refunds instead of clicking the "refund" button in the Payment-Gateway UI. This API can be invoked on any prior successful transaction. The transaction which is being refunded should be in either "paid" or "settled" state, or in "refunded" state (in case of partial amount refunds). Refunds can be either for the full amount paid by the customer, or any part of it.

The API needs a valid transaction ID as input.

**Note:** processing of refunds is subject to availability of funds in subsequent settlement cycles. This API will return a failure response in case sufficient funds are not available to process the refund.

### 7.1 Refund request API

**URL:** *https://{pg\_api\_url}/v2/refundrequest*

#### Request Parameters:

| Parameter Name     | Description   | Data type   | Optional / Mandatory |
|--------------------|---|-------------|----------------------|
| api_key            | Payment-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            |
| transaction_id     | The unique alphanumeric transaction ID generated by Payment-Gateway for a prior transaction.  | varchar(30) | Mandatory            |
| merchant_refund_id | This is your (merchant) refund reference number. It must be unique for every refund request. If a refund request is sent with same merchant_refund_id we return the response of the previously successful refund request. <b>Warning: If you are NOT using this field then be careful, as each request will be treated as a new refund request. Thus it is recommended to use this field.</b> | varchar(30) | Optional             |

|                   |  |               |           |
|-------------------|--|---------------|-----------|
| merchant_order_id | This is your (merchant) reference number which you submitted while making the original transaction. Note that if this value does not match with related transaction_id field then you will get error. In typical cases do not send this field.   | varchar(30)   | Optional  |
| amount            | The amount which needs to be refunded. This needs to be less than or equal to the transaction amount.  | decimal(10,2) | Mandatory |
| description       | Description of the refund. Usually the reason for issuing refund, as specified by merchant.  | varchar(500)  | Mandatory |
| hash              | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b> | varchar(255)  | Mandatory |

### Response Parameters:

The output is a JSON which contains the error(s), if any, in validation, or a simple success message which confirms that the refund request has been accepted and will be processed during subsequent settlement cycle.

If the request is successfully processed response you will get a “data” block, and in case of failure you will see “error” block, you will not get “data” key in case of error.

In case of success, **NOTE:** that *refund\_reference\_no* is returned by the bank and it can be null in case refunds are not initiate by bank immediately, but is done at end of the day.

```
{
  "data": {
    "transaction_id": "HDVISC7472820193",
    "refund_id": 4351,
```

```

"refund_reference_no": null
"merchant_refund_id": 76783_R_1,
"merchant_order_id": 76783,
}
}

```

In case of error,

```

{
  "error": {
    "code": 1039,
    "message": "The refund amount is greater than transaction amount"
  }
}

```

## 7.2 Refund Status API

If a refund is initiated either from merchant or payment gateway end and merchant wants to check its status (details such as if it is refunded or not how much amount was paid and how much is refunded will be posted in response). To check the status of any refund which was initiated merchant should post the API request.

**URL:** *https://{pg\_api\_url}/v2/refundstatus*

### Request Parameters:

| Parameter Name | Description   | Data type   | Optional / Mandatory |
|----------------|---|-------------|----------------------|
| api_key        | Payment-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            |

|                   |  |              |           |
|-------------------|--|--------------|-----------|
| transaction_id    | The unique alphanumeric transaction ID generated by Payment-Gateway for a prior transaction.   | varchar(30)  | Mandatory |
| merchant_order_id | This is your (merchant) reference number which you submitted while making the original transaction. Note that if this value does not match with related transaction_id field then you will get error. In typical cases do not send this field.   | varchar(30)  | Optional  |
| hash              | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b> | varchar(255) | Mandatory |

### Response Parameters:

The output is a JSON it will give all details about a refund if any initiated for this transaction, if not it will give json with error. Partial multiple refunds are also shown in refund\_details

If the request is successfully processed response you will get a "data" block, and in case of failure you will see "error" block, you will not get "data" key in case of error.

**NOTE:** that *refund\_reference\_no* is returned by the bank and it can be null in case refunds are not initiate by bank immediately but is done at end of the day.

```
{
  "data": {
    "transaction_id": "FDAXIP9740656834",
    "merchant_order_id": "351177",
    "refund_amount": 2.04,
    "transaction_amount": "2.04",
    "refund_details": [
      {
        "refund_id": 3523,
        "refund_reference_no": "602201803257434370",
```

```
"merchant_refund_id": null,  
"refund_amount": "2.04",  
"refund_status": "Customer Refunded",  
"date": "2018-02-01 11:19:49"  
}  
]  
},  
"hash":  
"20D8CB42D14C35AAEF06BB200C82E560DCC1D0C19EEFFBFD07CBEEB3BD39AE746AFB30A5803D6375  
27CE1A45AE367565E8AF5933809E3F597D7CDDDCDB3C28FE"  
}
```

In case of error,

```
{  
  "error": {  
    "code": 1050,  
    "message": "No data record found for the given input"  
  }  
}
```

## 8 SPLIT API

### 8.1 Split Settlement API

#### 8.1.1 Split transaction before settlement API

**URL:** *https://{pg\_api\_url}/v2/splitsettlementrequest*

If the split information is missing during payment request, this API can be used to send split information to the system. The split information needs to be sent before settlements are done. Otherwise, it will be rejected.

#### Request Parameters:

| Parameter Name | Description   | Data type    | Optional / Mandatory |
|----------------|---|--------------|----------------------|
| api_key        | The unique key provided to the merchant   | varchar(36)  | Mandatory            |
| order_id       | The order id of the transaction   |              | Mandatory            |
| split_info     | The json format data can contain vendor_code and vendor_percent or vendor_code and vendor_amount, see the json structure below.   | json         | Mandatory            |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b><br><br>hash = strtoupper(hash('sha512', salt api_key order_id split_info)) | varchar(255) | Mandatory            |

The split\_info parameter will be in json format as shown below:

```
{
  "vendors": [
    {
      "vendor_code": "2VEN449",
      "split_amount_percentage": "80"
    }
  ]
}
```

```
    },
    {
      "vendor_code": "XYZ123",
      "split_amount_fixed": "11"
    }
  ]
}
```

### Response Parameters:

The response will be in json format as show below:

In case of success,

```
{
  "data": {
    "message": "The split settlement request is successful."
  }
}
```

In case of total split percentage or amount exceeds 100% or total settlement amount

```
{
  "error": {
    "code": 1024,
    "message": "Sum of split amount should be less than or equal
to settlement amount."
  }
}
```

In case of vendor code invalid or not approved

```
{
  "error": {
    "code": 1007,
    "message": "One or more Codes is either not added or not
approved."
  }
}
```

## 9 VENDOR APIs

### 9.1 Add Vendor API

**URL:** *https://{pg\_api\_url}/v2/addvendor*

This API allows the merchant to register new vendors with the Payment-Gateway system. These vendors can also be added manually from the Payment-Gateway dashboard.

When a vendor is added, it is "non-approved" by default. Payment-Gateway will approve the vendors separately. This is for security purposes.

| Parameter Name         | Description   | Data type    | Optional / Mandatory |
|------------------------|---|--------------|----------------------|
| api_key                | Payment-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            |
| vendor_code            | This is the vendor code that you wish to add in the Payment-Gateway system. This has to be unique. Alphanumeric values are permitted here.  | varchar(30)  | mandatory            |
| vendor_name            | A descriptive name to identify the vendor.  | varchar(100) | mandatory            |
| vendor_contact_email   | Email address where the vendor can be contacted. Has to be a valid email address.   | varchar(200) | mandatory            |
| vendor_contact_num     | Phone number where the vendor can be contacted.   | varchar(10)  | mandatory            |
| vendor_contact_address | Address where the vendor can be reached.  | varchar(300) | optional             |
| account_name           | Account holder name (of the vendor bank account). Optional if UPI details are given.  | varchar(300) | optional             |
| account_number         | Account number of the vendor. Optional if UPI details are given.  | varchar(50)  | optional             |
| ifsc_code              | IFSC code of the vendor's bank. Optional if UPI details are given.  | varchar(50)  | optional             |
| bank_name              | Bank name of the vendor's bank. Optional if UPI details are given.  | varchar(200) | optional             |
| bank_branch            | Bank branch of the vendor's bank. Optional if UPI details are given.  | varchar(300) | optional             |
| upi_id                 | UPI VPA of the vendor. Optional if bank account details are given.  | varchar(50)  | optional             |
| vendor_pan             | PAN number of the vendor  | varchar(10)  | optional             |
| description_1          | Vendor description 1  | varchar(200) | optional             |
| description_2          | Vendor description 2  | varchar(200) | optional             |

|                    |   |              |           |
|--------------------|---|--------------|-----------|
| activate_bharat_qr | Pass 'y' in case bharat qr needs to be generated for this vendor  | varchar(1)   | optional  |
| aadhar_number      | Aadhar number is required if bharat qr needs to be generated  | varchar(12)  | optional  |
| hash               | <p>You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.</p> <p>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b></p> <p>Please ensure that the concatenated string (upon which the hash will be computed) is computed based on columns in alphabetical order. Please include only those columns in your hash calculation which you are actually passing to us. For example, if you are not passing vendor_contact_address do not include that in hash calculation.</p> | varchar(200) | mandatory |

**Note:** This API will return error if the vendor already exists in the system AND is active. If an inactive/disapproved vendor exists, this API will update the details for that vendor code.

The response will be in json format as show below:

In case of success,

```
{
  "data": {
    "code": "SUCCESS",
    "account_id": "0026",
    "message": "Vendor is added Successfully"
    "static_qr_code": "0002010102110216407758000003774806
1661004600000385150821hdfc000000162713126410010A0000005240116yap94026@
equitas02031.027230010A00000052401059402628300010A00000052401121234123
412345204739953033565802IN5907abc
pay6032bangalore610656000162120708940260012314D41"
  }
}
```

In case vendor already exists

```
{
  "error": {
    "code": 1024,
```

```

    "message": "Vendor code already exists"
  }
}

```

## 9.2 Modify Vendor API

**URL:** [https://{pg\\_api\\_url}/v2/modifyvendor](https://{pg_api_url}/v2/modifyvendor)

Pre-existing vendors in the system can be modified using this API. This API works on approved as well as non-approved vendors. However, any modification to a pre-existing active vendor will immediately disapprove that vendor, automatically. If the vendor that is being modified does not exist, the API will return an error and will NOT automatically add the vendor. This will change the default account for the vendor

| Parameter Name         | Description   | Data type    | Optional/<br>Mandatory |
|------------------------|---|--------------|------------------------|
| api_key                | Payment-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              |
| vendor_code            | This is the vendor code that you wish to modify in the Payment-Gateway system. This value must already exist in the system, failing which Payment-Gateway will return an error.   | varchar(30)  | mandatory              |
| vendor_name            | A descriptive name to identify the vendor.  | varchar(100) | optional               |
| vendor_contact_email   | Email address where the vendor can be contacted. Has to be a valid email address.   | varchar(200) | optional               |
| vendor_contact_num     | Phone number where the vendor can be contacted.   | varchar(10)  | optional               |
| vendor_contact_address | Address where the vendor can be reached. Optional.  | varchar(300) | optional               |
| account_id             | Account id which needs to be modified. The account_id is returned in the addvendor API. If it is not passed, the last approved account of the vendor is updated   | varchar(10)  | optional               |
| account_name           | Account holder name (of the vendor bank account). Optional if UPI details are given.  | varchar(300) | optional               |
| account_number         | Account number of the vendor. Optional if UPI details are given.  | varchar(50)  | optional               |
| ifsc_code              | IFSC code of the vendor's bank. Optional if UPI details are given.  | varchar(50)  | optional               |

|                    |   |              |           |
|--------------------|---|--------------|-----------|
| bank_name          | Bank name of the vendor's bank. Optional if UPI details are given.  | varchar(200) | optional  |
| bank_branch        | Bank branch of the vendor's bank. Optional if UPI details are given.  | varchar(300) | optional  |
| upi_id             | UPI VPA of the vendor. Optional if bank account details are given   | varchar(50)  | optional  |
| vendor_pan         | PAN number of the vendor  | varchar(10)  | optional  |
| description_1      | Vendor description 1  | varchar(200) | optional  |
| description_2      | Vendor description 2  | varchar(200) | optional  |
| activate_bharat_qr | Pass 'y' in case bharat qr needs to be generated for this vendor  | varchar(1)   | optional  |
| aadhar_number      | Aadhar number is required if bharat qr needs to be generated  | varchar(12)  | optional  |
| hash               | <p>You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.</p> <p>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b></p> <p>Please ensure that the concatenated string (upon which the hash will be computed) is computed based on columns in alphabetical order. Please include only those columns in your hash calculation which you are actually passing to us. For example, if you are not passing vendor_contact_address do not include that in hash calculation.</p> | varchar(200) | mandatory |

The response will be in json format as show below:

In case of success,

```
{
  "data": {
    "code": "SUCCESS",
    "message": "Vendor Details Updated Successfully"
  }
}
```

### 9.3 Add Vendor Accounts API

**URL: [https://{pg\\_api\\_url}/v2/addvendoraccount](https://{pg_api_url}/v2/addvendoraccount)**

Multiple accounts can be added to pre-existing vendors in the system using this API. This API works on

approved as well as non-approved vendors. If the vendor that is being given does not exist, the API will return an error and will NOT automatically add details to the vendor.

| Parameter Name  | Description  | Data type    | Optional/<br>Mandatory |
|-----------------|--|--------------|------------------------|
| api_key         | Payment-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              |
| vendor_code     | This is the vendor code that you wish to modify in the Payment-Gateway system. This value must already exist in the system, failing which Payment-Gateway will return an error.  | varchar(30)  | mandatory              |
| account_name    | Account holder name (of the vendor bank account). Optional if UPI details are given.   | varchar(300) | optional               |
| account_number  | Account number of the vendor. Optional if UPI details are given.   | varchar(50)  | optional               |
| ifsc_code       | IFSC code of the vendor's bank. Optional if UPI details are given.   | varchar(50)  | optional               |
| bank_name       | Bank name of the vendor's bank. Optional if UPI details are given.   | varchar(200) | optional               |
| bank_branch     | Bank branch of the vendor's bank. Optional if UPI details are given.   | varchar(300) | optional               |
| upi_id          | UPI VPA of the vendor. Optional if bank account details are given  | varchar(50)  | optional               |
| default_account | Whether this will be the default account of the vendor or not. Possible values are "y" or "n".<br>IMPORTANT: System can have only one default account, if the value is passed as 'y' and default account exist, error will be displayed  | varchar(1)   | mandatory              |
| hash            | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b><br><br>Please ensure that the concatenated string (upon which the hash will be computed) is | varchar(200) | mandatory              |

|  |   |  |  |
|--|---|--|--|
|  | computed based on columns in alphabetical order. Please include only those columns in your hash calculation which you are actually passing to us. For example, if you are not passing vendor_contact_address do not include that in hash calculation. |  |  |
|--|---|--|--|

The response will be in json format as show below:

In case of success,

```
{
  "data": {
    "code": "SUCCESS",
    "account_id": "12313",
    "message": "Vendor account is added Successfully"
  }
}
```

## 9.4 Delete Vendor API

**URL:** [https://{pg\\_api\\_url}/v2/deletevendor](https://{pg_api_url}/v2/deletevendor)

This API can be used to delete a pre-existing vendor from the Payment-Gateway system. Subsequent to deletion, there can be no further split payments to this vendor. Importantly, deletion of a vendor will NOT impact pending payouts to the vendor. Any pending settlements will still occur

| Parameter Name | Description  | Data type    | Optional/Mandatory |
|----------------|--|--------------|--------------------|
| api_key        | Payment-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          |
| vendor_code    | This is the vendor code that you wish to delete from the Payment-Gateway system. This value must already exist in the system, failing which Payment-Gateway will return an error.  | varchar(30)  | mandatory          |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b> | varchar(255) | mandatory          |

|  |   |  |  |
|--|---|--|--|
|  | hash = toUpper ( sha512 (SALT  api_key  vendor_code)) |  |  |
|--|---|--|--|

This API can be used to delete a pre-existing vendor from the Payment-Gateway system. Subsequent to deletion, there can be no further split payments to this vendor. Importantly, deletion of a vendor will NOT impact pending payouts to the vendor. Any pending settlements will still occur

The response will be in json format as show below:

In case of success,

```
{
  "data": {
    "code": "SUCCESS",
    "message": "Vendor is deleted Successfully"
  }
}
```

## 9.5 Get Vendor API

**URL:** [https://{pg\\_api\\_url}/v2/vendorstatus](https://{pg_api_url}/v2/vendorstatus)

This API can be used to delete a pre-existing vendor from the Payment-Gateway system.

| Parameter Name | Description   | Data type    | Optional/ Mandatory |
|----------------|---|--------------|---------------------|
| api_key        | Payment-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           |
| vendor_code    | This is the vendor code that you wish to retrieve from the Payment-Gateway system. This value must already exist in the system, failing which Payment-Gateway will return an error.   | varchar(30)  | mandatory           |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</b><br><br>hash = toUpper ( sha512 (SALT  api_key  vendor_code)) | varchar(255) | mandatory           |

This API can be used to get details of a pre-existing vendor from the Payment-Gateway system.

The response will be in json format as show below:

In case of success,

```
{
  "data": {
    "vendor_code": "5dlee6743a3asd2d476a669b",
    "vendor_name": "Sindhu",
    "vendor_contact_email": "sharathkumar@example.com",
    "vendor_contact_num": "9900990099",
    "vendor_contact_address": "Domlur, Indira Nagar, Bangalore",
    "vendor_pan": "ARSPH1234Q",
    "vendor_approved": "y",
    "vendor_split_percentage": null,
    "vendor_split_amount": null,
    "earliest_settlement_time_frame": null,
    "latest_settlement_time_frame": null,
    "vendor_logo": null,
    "qr_code":
      "
      AB1BMVEX///8AAABVwtN+AAAAAXRSTlMAQObYZgAAAAIwSFlzAAAQAAADsQBlSsOGwAA...
      .....vRou3ed6h+OmH3Ch8fMD+wfnU1Eyj4zFVQAAAABJRU5ErkJggg==",
    "bank_accounts": [
      {
        "account_id": 3288,
        "account_name": "Dummy",
        "account_number": "1000000001",
        "ifsc_code": "UTIB0000003",
        "bank_name": "Dummy Bank",
        "bank_branch": "Demo Road",
        "upi_id": null,
        "bank_approved": "y",
        "default_account": "y"
      }
    ]
  }
}
```

## 10 SETTLEMENT APIS

### 10.1 Get Settlements API

URL: [https://{pg\\_api\\_url}/v2/getsettlements](https://{pg_api_url}/v2/getsettlements)

This API allows a merchant to programmatically access the status of any of his past settlements and other pertinent information pertaining to a prior settlement. If this API returns a blank `bank_reference_number`, it means the amount is not yet settled. If the API returns no data, it means that the system has not calculated settlements yet, you would need to re-check after 12:30 AM.

Please note that this API will not provide any information for failed transactions since, there can be no settlement for a failed transaction. To obtain information about failed transactions, use the payment status API described in an earlier section.

#### 10.1.1 Parameters to be POSTed in Request

| Parameter Name | Description  | Data type    | Optional / Mandatory |
|----------------|--|--------------|----------------------|
| api_key        | Payment-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            |
| bank_reference | The bank reference number of the actual NEFT/IMPS/RTGS transaction performed by Payment-Gateway to the merchant's current account  | varchar(100) | Optional             |
| date_from      | The start date from which you need to retrieve settlement information. This needs to be passed in DD-MM-YYYY format.   | varchar(10)  | Optional             |
| date_to        | The end date at which you need to retrieve settlement information. This needs to be passed in DD-MM-YYYY format.   | varchar(10)  | Optional             |
| completed      | Whether settlement is completed or not. Pass in 'y' or 'n' here.   | varchar(1)   | Optional             |
| settlement_id  | The unique numeric settlement ID assigned to each settlement   | number(20)   | Optional             |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT</b> | varchar(255) | Mandatory            |

|  |  |  |  |
|--|--|--|--|
|  | STORE SALT IN ANDROID APP APK or IPHONE APP package. |  |  |
|--|--|--|--|

This API returns a JSON in the following format:

```
{
  "data": [
    {
      "settlement_id": 10075,
      "bank_reference": "710061536126",
      "payout_amount": "2.06",
      "completed": "y",
      "account_name": "Tester Sharma",
      "account_number": "50100012341231",
      "ifsc_code": "HDFC0000002",
      "bank_name": "HDFC BANK",
      "bank_branch": "CMH RD, INDIRA NAGAR BRANCH",
      "settlement_datetime": "2017-02-20 16:31:28",
      "sale_amount": "3.00",
      "chargeback_amount": "0.00",
      "refund_amount": "0.00"
    }
  ],
  "hash":
  "684CDA22F7A429D68281444A8F6809A5FEFEA7A055258984E129554AC359C956E58E36B67A4EB9F948
  1E616888E722DDB95A81EFBED4416B24F19E3126077F5E"
}
```

In case there is no record found in the system for the combination of input parameter, following error is returned

```
{
  "error": {
    "code": 404,
    "message": "No record found"
  }
}
```

## 10.2 Get Settlement Details API

**URL:** [https://{pg\\_api\\_url}/v2/getsettlementdetails](https://{pg_api_url}/v2/getsettlementdetails)

This API allows a merchant to programmatically access the status of any of his past **settlement details** (transaction level settlements).

Please note that this API will not provide any information for failed transactions since by definition, there can be no settlement for a failed transaction. To obtain information about failed transactions, use the payment status API described in an earlier section.

### 10.2.1 Parameters to be POSTed in Request

| Parameter Name | Description   | Data type    | Optional / Mandatory |
|----------------|---|--------------|----------------------|
| api_key        | Payment-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            |
| order_id       | Order ID passed by the merchant during the original payment transaction request   | varchar(30)  | optional             |
| transaction_id | Transaction ID assigned by Payment-Gateway for this successful transaction  | varchar(30)  | optional             |
| bank_code      | Bank code signifying payment mode and channel   | varchar(4)   | optional             |
| customer_phone | Phone number of customer as provided during the original paymentrequest API call  | varchar(30)  | optional             |
| customer_email | Email ID of customer as provided during the original paymentrequest API call  | varchar(255) | optional             |
| customer_name  | Name of customer as provided during the original paymentrequest API call  | varchar(255) | optional             |
| date_from      | The start date from which you need to retrieve settlement detail information. This needs to be passed in DD-MM-YYYY format.   | varchar(10)  | optional             |
| date_to        | The end date at which you need to retrieve settlement detail information. This needs to be passed in DD-MM-YYYY format.   | varchar(10)  | optional             |
| completed      | Whether settlement is completed or not. Pass in 'y' or 'n' here.  | varchar(1)   | optional             |
| settlement_id  | The unique numeric settlement ID assigned to each settlement  | number(20)   | optional             |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package.</b> | varchar(255) | mandatory            |

This API returns a JSON in the following format:

```

{
  "data": [
    {
      "transaction_id": "HDMASC2746901262",
      "order_id": "225495",
      "settlement_id": 27837,
      "bank_reference": "710061536126",
      "settlement_datetime": null,
      "customer_name": "Tester",
      "customer_email": "tester@example.com",
      "customer_phone": "8050603774",
      "completed": "y",
      "description": "Settlement for Rs. 2.06 paid through transaction ID HDMASC2746901262 on 2017-09-28 13:36:19 for merchant hotel booking",
      "gross_transaction_amount": "2.06",
      "payment_mode": "Credit Card",
      "payment_channel": "Master",
      "applicable_tdr_percent": "3.00",
      "applicable_tdr_fixed_fee": "0.00",
      "percent_tdr_paid_by_merchant": "0",
      "tdr_amount": "0.06",
      "tax_on_tdr_amount": "0.00",
      "amount_reimbursed": "2.00"
    }
  ],
  "hash":
  "D2EFF4776D973DA46563DA0F80139B84AFED77C58496A34DD0D653272A0EE1E5D09F4C94AD4394512B16341A5A44906B4B10FF5B6AA1F03DE98A164B39881C4E"
}

```

In case there is no record found in the system for the combination of input parameter, following error is returned

```

{
  "error": {
    "code": 404,
    "message": "No record found"
  }
}

```

## 11 CHALLAN PAYMENT APIs

### 11.1 Request challan payment API

**URL:** [https://{pg\\_api\\_url}/v1/requestchallan](https://{pg_api_url}/v1/requestchallan)

This API allows the merchant to create a link which can be sent to customers by email and/or SMS. This link allows the customer to make easy payments without data entry hassles.

On clicking this link, the customer is taken directly to a confirmation page where he can verify his details (email ID, name and amount), and on confirmation, he is taken to the payment page.

| Parameter Name | Description   | Data type     | Optional/Mandatory |
|----------------|---|---------------|--------------------|
| api_key        | Payment-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          |
| name           | Name of the person whom the invoice is addressed to.  | varchar(100)  | Mandatory          |
| mobile         | Phone number of the person whom the invoice is addressed to.  | varchar(10)   | mandatory          |
| email          | Email ID of the person whom the invoice is addressed to.  | varchar(100)  | mandatory          |
| amount         | Amount which the user needs to pay.   | decimal(15,2) | mandatory          |
| purpose        | Purpose of payment - this should be a descriptive string which clearly tells the user what he is paying for.  | varchar(100)  | mandatory          |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package.</b> | varchar(255)  | mandatory          |

### 11.2 Request challan payment API url

**URL:** [https://{pg\\_api\\_url}/v1/generatchallanurl](https://{pg_api_url}/v1/generatchallanurl)

This API allows the merchant to create a url which can be sent to customers by email and/or SMS. This url allows the customer to make easy payments without data entry hassles.

On clicking above url, the customer is taken directly to a confirmation page where he can verify his details (email ID, name and amount), and on confirmation, he is taken to the payment page.

### 11.2.1 Request parameters are as following:

| Parameter Name | Description  | Data type     | Optional/Mandatory |
|----------------|--|---------------|--------------------|
| api_key        | Payment-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          |
| name           | Name of the person whom the invoice is addressed to.   | varchar(100)  | Mandatory          |
| mobile         | Phone number of the person whom the invoice is addressed to.   | varchar(10)   | mandatory          |
| email          | Email ID of the person whom the invoice is addressed to.   | varchar(100)  | mandatory          |
| amount         | Amount which the user needs to pay.  | decimal(15,2) | mandatory          |
| purpose        | Purpose of payment - this should be a descriptive string which clearly tells the user what he is paying for.   | varchar(100)  | mandatory          |
| hash           | You need to compute a hash of all your parameters and pass that hash to Payment-Gateway, using the following mechanism:<br>toUpper ( sha512 (SALT amount  api_key   email mobile name purpose))<br>Note: the SALT will be provided by Payment-Gateway separately.<br>Please ensure that the concatenated string (upon which the hash will be computed) is in alphabetical order.<br><b>Note:</b> NEVER PASS SALT IN A FORM | varchar(255)  | Mandatory          |

### 11.2.2 Response from this API will be in JSON format:

On successful call to this API you will receive JSON response in following format.

```
{
  "data": {
    "url": "http://biz.localhost.com/challan/b39b0596-73c4-4b7e-b63d-bbc13361e044",
    "uuid": "b39b0596-73c4-4b7e-b63d-bbc13361e044",
    "tnp_id": 81600
  }
}
```

**data** - successful response will have "data" tag.

**url** - this is the url what can be distributes as suitable.

**uuid** - this is the unique identifier for this request.

**tnp\_id** - this is another unique identifier that can be used for getting the transaction details using paymentStatusById API.

On failure json response is as following:

```
{
```

```
"error": {  
  "code": 221,  
  "message": "GEN-UNAUTHORIZED - The api key field is incorrect"  
}
```

**error** - erred response will have "error" tag.

**code** - this is error category code

**message** - this is more descriptive error tag and error message.

### 11.2.3 List of error codes and corresponding messages:

| Code | Message  |
|------|--|
| 221  | GEN-UNAUTHORIZED<br>The api key field is incorrect<br>The hash key field is invalid  |
| 998  | GEN-INVALID-PARAMS<br>The name field is required.<br>The email field is required.<br>The mobile field is required.<br>The amount field is required.<br>The purpose field is required.<br>The hash field is required. |

## 12 SERVER TO SERVER CALL BACK (WEB HOOKS)

### 12.1 Server to server response on Payment

To get server to server response, add callback URL in parameter named "Payment Callback URL" in your Payment-Gateway dashboard. If this is not found contact Payment-Gateway to set this up for you.

Whenever there is a successful payment done by your customer apart from receiving success or failure message on customers' browser, following response parameters are also posted to the mentioned callback URL.

These are very same response that we send as response to **paymentrequest** API.

| Parameter name   | Description   |
|------------------|---|
| transaction_id   | A unique ID that can be used to trace the transaction uniquely within Payment-Gateway. Transaction IDs are alphanumeric.  |
| payment_method   | This tells the payment method used by customer - example: "credit card", "debit card", "netbanking", etc.   |
| payment_datetime | Date and Time of this payment in "DD-MM-YYYY HH:MM:SS" format   |
| response_code    | Status of the transaction (return code). 0 signifies successful transaction. Non-zero values signify error.   |
| response_message | Can have a value of "success" or "failure". Order   |
| order_id         | The same order_id that was originally posted by the merchant in the request.  |
| amount           | The same original amount that was sent by the merchant in the transaction request.  |
| currency         | This is the 3digit currency code (INR), it will be same value that was originally sent by merchant.   |
| description      | The same description that was originally sent by the merchant in the transaction request.   |
| name             | The same value that was originally sent by merchant   |
| email            | The same value that was originally sent by merchant   |
| phone            | The same value that was originally sent by merchant   |
| address_line_1   | The same value that was originally sent by merchant   |
| address_line_2   | The same value that was originally sent by merchant   |
| city             | The same value that was originally sent by merchant   |
| state            | The same value that was originally sent by merchant   |
| country          | The same value that was originally sent by merchant   |
| zip_code         | The same value that was originally sent by merchant   |
| udf1             | The same value that was originally sent by merchant   |
| udf2             | The same value that was originally sent by merchant   |
| udf3             | The same value that was originally sent by merchant   |
| udf4             | The same value that was originally sent by merchant   |
| udf5             | The same value that was originally sent by merchant   |
| hash             | Payment-Gateway calculates the hash using the same algorithm which was outlined earlier. Hence, the merchant needs to check whether this returned hash matches the calculated hash. |

## 12.2 Server to server response on Settlement

To get server to server response, add callback URL in parameter named "Settlement Callback URL" in your Payment-Gateway dashboard. If this is not found contact Payment-Gateway to set this up for you.

Whenever there is a successful settlement done by Payment-Gateway to your bank account apart from receiving success or failure email message, following response parameters are also posted to the mentioned callback URL.

These are very same response that we send as response to **getsettlements** API.

| Parameter name      | Description   |
|---------------------|---|
| settlement_id       | Settlement Id for this aggregated settlement                                  |
| bank_reference      | Bank reference Number   |
| payout_amount       | Aggregated Amount paid to merchant  |
| completed           | Settlement is completed or not, 'y' or 'n'                                    |
| account_name        | Account Holders Name to which the Amount is settled                           |
| account_number      | Account Number to which the Amount is settled                                 |
| ifsc_code           | IFSC Code of the branch to which Account Number belongs                       |
| bank_name           | Bank name to which Account Number belongs                                     |
| settlement_datetime | Date of settlement  |
| sale_amount         | Total sale amount for the transactions included in this aggregated settlement |
| chargeback_amount   | Amount deducted from the sale amount for chargeback adjustment                |
| refund_amount       | Amount deducted from the sale amount for refunds adjustment.                  |

## 13 SEAMLESS PAYMENT REQUEST API

In case you perform the normal payment integration process as outlined earlier in the document, the customer will necessarily be redirected to the Payment-Gateway payment page wherein he will be required to enter his card details or select the appropriate bank/payment instrument (such as wallets etc.) with which he would like to make the payment.

If you would like your customer to perform the bank selection and/or enter his card information on your site, without using the Payment-Gateway payment page, this can be achieved using the seamless payment request API. Payment-Gateway would provide you a list of appropriate bankcodes which you would be required to pass along with the payment request. These bankcodes are available in Appendix 3 of this document.

Please note that in order to use our seamless payment request API you would need to be PCI-DSS compliant. For more information on PCI compliance, or if you would like us to assist you in your PCI compliance efforts, please contact your Payment-Gateway relationship manager.

### 13.1 Steps for Integration

- You need to submit a POST REQUEST to our server, at the below mentioned URL  
[https://{pg\\_api\\_url}/v2/paymentseamlessrequest](https://{pg_api_url}/v2/paymentseamlessrequest)
- Note: hash is a mandatory parameter. If your hash is not properly calculated or does not match for whatever reason, we will not be able to process the payment. The usage of hash is explained in subsequent sections.
- When you submit your transaction request to Payment-Gateway, we assign a transaction ID to you.
- The customer is automatically redirected to the appropriate bank or 3D-secure page, as the case may be.
- After the customer pays the entire amount using one or more payment instruments, he is redirected back to the merchant site.
- We recommend that you check the hash at your end again, after we send back the response to you. This is essential to prevent user data tampering fraud.

### 13.2 Parameters to be POSTed in Seamless Payment Request

**URL:** [https://{pg\\_api\\_url}/v2/paymentseamlessrequest](https://{pg_api_url}/v2/paymentseamlessrequest)

| Parameter Name | Description   | Data type   | Optional / Mandatory |
|----------------|---|-------------|----------------------|
| api_key        | Payment-Gateway 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 | varchar(36) | Mandatory            |

|                    |   |               |             |
|--------------------|---|---------------|-------------|
|                    | be one different api_key per login account that is assigned to you.   |               |             |
| order_id           | This is your (merchant) reference number. It must be unique for every transaction.  | varchar(30)   | Mandatory   |
| mode               | This is the payment mode (TEST or LIVE are valid values). Defaults to LIVE.   | varchar(4)    | Optional    |
| amount             | This is the payment amount.   | decimal(15,2) | Mandatory   |
| currency           | This is the 3digit currency code (INR)  | varchar(3)    | Mandatory   |
| description        | Brief description of product or service that the customer is being charged for.   | varchar(500)  | Mandatory   |
| name               | Name of customer.   | varchar(100)  | Mandatory   |
| email              | Customer email address.   | varchar(100)  | Mandatory   |
| phone              | Customer phone number   | varchar(50)   | Mandatory   |
| address_line_1     | Customer address  | varchar(100)  | Optional    |
| address_line_2     | Customer address 2  | varchar(100)  | Optional    |
| city               | Customer city   | varchar(50)   | Mandatory   |
| state              | Customer State  | varchar(50)   | Optional    |
| country            | Customer country must be IND  | varchar(50)   | Mandatory   |
| zip_code           | Customer zip code   | varchar(20)   | Mandatory   |
| udf1               | User defined field 1  | varchar(300)  | Optional    |
| udf2               | User defined field 2  | varchar(300)  | Optional    |
| udf3               | User defined field 3  | varchar(300)  | Optional    |
| udf4               | User defined field 4  | varchar(300)  | Optional    |
| udf5               | User defined field 5  | varchar(300)  | Optional    |
| bank_code          | Bank code identifies the payment mode and channel.  | varchar(20)   | Mandatory   |
| card_number        | Card number (11 to 19 digits)   | varchar(19)   | Conditional |
| expiry_date        | Expiry date in mm/yyyy format   | varchar(7)    | Conditional |
| card_holder_name   | Card holder name  | varchar(30)   | Conditional |
| cvv                | CVV/CVC   | varchar(4)    | Conditional |
| return_url         | Return URL success - Payment-Gateway will make a POST request to this URL after successful transaction, with a set of parameters, which you can process as you want to.                     | varchar(300)  | Mandatory   |
| return_url_failure | Return URL failure - Payment-Gateway will make a POST request to this URL after failed transaction, with a set of parameters, which you can process as you want to. Defaults to return_url. | varchar(300)  | Optional    |
| return_url_cancel  | Return URL cancel - Payment-Gateway will make a POST request to this URL after user-cancelled transaction, with a set of parameters, which you can process as you want to.                  | varchar(300)  | Optional    |
| hash               | You need to compute a hash of all your parameters and pass that hash to Payment-  | varchar(255)  | Mandatory   |

|  |  |  |  |
|--|--|--|--|
|  | <p>Gateway, for details about calculating hash refer Appendix 2.</p> <p>Note: the SALT will be provided by Payment-Gateway separately. <b>NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package.</b></p> |  |  |
|--|--|--|--|

### 13.3 Response Parameters

Response sent after complete amount is paid. This is a server to browser response

| Parameter name   | Description  |
|------------------|--|
| transaction_id   | A unique ID that can be used to trace the transaction uniquely within Payment-Gateway. Transaction IDs are alphanumeric. |
| payment_mode     | This tells the payment mode used by customer - example: "credit card", "debit card", "netbanking", etc.                  |
| Payment_channel  | The actual payment channel - for example Visa, Master, Diners, HDFC Bank, MobiKwik, etc.                                 |
| payment_datetime | Date and Time of this payment in "DD-MM-YYYY HH:MM:SS" format  |
| response_code    | Status of the transaction (return code). 0 signifies successful transaction. Non-zero values signify error.              |
| response_message | Can have a value of "Transaction Successful" or "Transaction Failed" or "Transaction Cancelled".                         |
| error_desc       | Failure reason (if transaction is failed)  |
| order_id         | The same order_id that was originally posted by the merchant in the request.   |
| amount           | The same original amount that was sent by the merchant in the transaction request.                                       |
| currency         | This is the 3digit currency code (INR), it will be same value that was originally sent by merchant.                      |
| description      | The same description that was originally sent by the merchant in the transaction request.                                |
| name             | The same value that was originally sent by merchant  |
| email            | The same value that was originally sent by merchant  |
| phone            | The same value that was originally sent by merchant  |
| address_line_1   | The same value that was originally sent by merchant  |
| address_line_2   | The same value that was originally sent by merchant  |
| city             | The same value that was originally sent by merchant  |
| state            | The same value that was originally sent by merchant  |
| country          | The same value that was originally sent by merchant  |
| zip_code         | The same value that was originally sent by merchant  |
| udf1             | The same value that was originally sent by merchant  |
| udf2             | The same value that was originally sent by merchant  |
| udf3             | The same value that was originally sent by merchant  |
| udf4             | The same value that was originally sent by merchant  |

|            |  |
|------------|--|
| udf5       | The same value that was originally sent by merchant  |
| cardmasked | The card number used by customer for making payment. This will NOT BE SENT by default, and is sent only in case the merchant has been explicitly approved to receive this information. Else this is always sent as null. |
| hash       | Payment-Gateway calculates the hash using the same algorithm which was outlined earlier. Hence, the merchant needs to check whether this returned hash matches the calculated hash.                                      |

## **14 APPENDIX 1 - References**

We have sample code in PHP available for payment gateway integration, we also have SDKs and various pre-built plugins of popular cart applications, to get the same get in contact with your contact person at Payment-Gateway

## 15 APPENDIX 2 - Hash calculation guide

### 15.1 How to Calculate Hash on API request

To calculate hash, you will need the salt provided by Payment-Gateway.

#### 15.1.1 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

##### 15.1.1.1 Example PHP code to generate hash

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

### 15.2 How to check the response Hash

It is important to make sure the response received from Payment-Gateway 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 is 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 Payment-Gateway APIs.

To check hash, you will need the salt provided by Payment-Gateway.

## 15.2.1 Hash checking algorithm

### 15.2.1.1 Example PHP code to check hash

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

### 15.2.1.2 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, JSON_UNESCAPED_SLASHES);

    /* 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;
}
```

## 16 APPENDIX 3 - List of bank codes

| <b>Bankcode</b> | <b>Payment Mode</b>       | <b>Payment Channel</b>                            | <b>Description</b>                                |
|-----------------|---------------------------|---|---|
| <b>VISC</b>     | Credit Card               | Visa  | Visa Credit Card                                  |
| <b>MASC</b>     | Credit Card               | Master  | Master Credit Cards                               |
| <b>DINC</b>     | Credit Card               | Diners  | Diners Credit Card                                |
| <b>AMXC</b>     | Credit Card               | Amex  | American Express Credit Card                      |
| <b>RUPC</b>     | Credit Card               | Rupay   | Rupay Credit Card                                 |
| <b>VICC</b>     | Commercial Credit Card    | Visa  | Visa Commercial Credit Card                       |
| <b>MACC</b>     | Commercial Credit Card    | Master  | Master Commercial Credit Card                     |
| <b>VICI</b>     | International Credit Card | Visa  | Visa International Credit Card                    |
| <b>MACI</b>     | International Credit Card | Master  | Master International Credit Card                  |
| <b>AMXI</b>     | International Credit Card | Amex  | American Express International Credit Card        |
| <b>DINI</b>     | International Credit Card | Diners  | Diners International Credit Card                  |
| <b>VISD</b>     | Debit Card                | Visa  | Visa Debit Card                                   |
| <b>MASD</b>     | Debit Card                | Master  | Master Debit Card                                 |
| <b>MAED</b>     | Debit Card                | Maestro (non-SBI)                                 | Maestro Debit Card (non-SBI)                      |
| <b>MSED</b>     | Debit Card                | Maestro (SBI)                                     | Maestro Debit Card (SBI)                          |
| <b>RUPD</b>     | Debit Card                | Rupay   | Rupay Debit Card                                  |
| <b>VIDI</b>     | International Debit Card  | Visa  | Visa International Debit Card                     |
| <b>MADI</b>     | International Debit Card  | Master  | Master International Debit Card                   |
| <b>IOBP</b>     | Debit Pin                 | Indian Overseas Bank                              | Indian Overseas Bank                              |
| <b>SYDP</b>     | Debit Pin                 | Syndicate Bank                                    | Syndicate Bank                                    |
| <b>AXIP</b>     | Debit Pin                 | AXIS Bank   | AXIS Bank   |
| <b>ADBP</b>     | Debit Pin                 | Andhra Bank                                       | Andhra Bank                                       |
| <b>UCOP</b>     | Debit Pin                 | UCO Bank  | UCO Bank  |
| <b>SUBP</b>     | Debit Pin                 | Suryodaya Bank                                    | Suryodaya Bank                                    |
| <b>ICIP</b>     | Debit Pin                 | ICICI Bank  | ICICI Bank  |
| <b>IDFP</b>     | Debit Pin                 | IDFC Bank   | IDFC Bank   |
| <b>SBIP</b>     | Debit Pin                 | State Bank Of India                               | State Bank Of India                               |
| <b>CBIP</b>     | Debit Pin                 | Central Bank Of India                             | Central Bank Of India                             |
| <b>ALLN</b>     | Netbanking                | Allahabad Bank NetBanking (Erstwhile Indian Bank) | Allahabad Bank NetBanking (Erstwhile Indian Bank) |

|             |            |   |   |
|-------------|------------|---|---|
| <b>ADBN</b> | Netbanking | Union Bank of India (Erstwhile Andhra Bank)             | Union Bank of India (Erstwhile Andhra Bank)             |
| <b>ADBM</b> | Netbanking | Union Bank of India (Erstwhile Andhra Bank) - Corporate | Union Bank of India (Erstwhile Andhra Bank) - Corporate |
| <b>AXIN</b> | Netbanking | AXIS Bank   | AXIS Bank NetBanking                                    |
| <b>BDNN</b> | Netbanking | Bandhan bank  | Bandhan Bank  |
| <b>BBKN</b> | Netbanking | Bank of Bahrain and Kuwait                              | Bank of Bahrain and Kuwait                              |
| <b>BBRM</b> | Netbanking | Bank of Baroda - Corporate                              | Bank of Baroda Corporate Banking                        |
| <b>BBRN</b> | Netbanking | Bank of Baroda - Retail                                 | Bank of Baroda Retail Banking                           |
| <b>BOIN</b> | Netbanking | Bank of India   | Bank of India   |
| <b>BOMN</b> | Netbanking | Bank of Maharashtra                                     | Bank of Maharashtra                                     |
| <b>BCBN</b> | Netbanking | Bassein Catholic Bank                                   | Bassein Catholic Bank                                   |
| <b>BMBN</b> | Netbanking | Bharatiya Mahila Bank                                   | Bharatiya Mahila Bank                                   |
| <b>CANN</b> | Netbanking | Canara Bank   | Canara Bank   |
| <b>CSBN</b> | Netbanking | Catholic Syrian Bank                                    | Catholic Syrian Bank                                    |
| <b>CBIN</b> | Netbanking | Central Bank Of India                                   | Central Bank Of India                                   |
| <b>CITN</b> | Netbanking | Citi Bank NetBanking                                    | Citi Bank NetBanking                                    |
| <b>CUBN</b> | Netbanking | City Union Bank   | City Union Bank   |
| <b>CRPN</b> | Netbanking | Union Bank of India (Erstwhile Corporation Bank)        | Union Bank of India (Erstwhile Corporation Bank)        |
| <b>COSN</b> | Netbanking | Cosmos Bank   | Cosmos Bank   |
| <b>DBSN</b> | Netbanking | DBS Bank  | DBS Bank  |
| <b>DCBM</b> | Netbanking | DCB Bank - Corporate                                    | DCB Bank - Corporate Netbanking                         |
| <b>DENN</b> | Netbanking | Dena Bank   | Dena Bank   |
| <b>DSHN</b> | Netbanking | Deutsche Bank   | Deutsche Bank   |
| <b>DCBN</b> | Netbanking | Development Credit Bank                                 | Development Credit Bank                                 |
| <b>DHNM</b> | Netbanking | Dhanlakshmi Bank - Corporate Net Banking                | Dhanalakshmi Bank - corporate                           |
| <b>DHNN</b> | Netbanking | Dhanlakshmi Bank  | Dhanalakshmi Bank                                       |
| <b>FEDN</b> | Netbanking | Federal Bank  | Federal Bank  |
| <b>HDFN</b> | Netbanking | HDFC Bank   | HDFC Bank   |
| <b>ICIN</b> | Netbanking | ICICI Bank  | ICICI Netbanking  |
| <b>IDFN</b> | Netbanking | IDFC Bank   | IDFC Bank   |
| <b>ININ</b> | Netbanking | Indian Bank   | Indian Bank   |
| <b>IOBN</b> | Netbanking | Indian Overseas Bank                                    | Indian Overseas Bank                                    |
| <b>INDN</b> | Netbanking | IndusInd Bank   | IndusInd Bank   |
| <b>IDBN</b> | Netbanking | IDBI  | Industrial Development Bank of India                    |
| <b>INGN</b> | Netbanking | ING Vysya Bank  | ING Vysya Bank  |
| <b>JAKN</b> | Netbanking | Jammu and Kashmir Bank                                  | Jammu and Kashmir Bank                                  |
| <b>JSBN</b> | Netbanking | Janata Sahakari Bank                                    | Janata Sahakari Bank                                    |
| <b>KJBN</b> | Netbanking | Kalyan Janata Sahakari Bank                             | Kalyan Janata Sahakari Bank                             |
| <b>KRKN</b> | Netbanking | Karnataka Bank  | Karnataka Bank  |

|             |            |   |   |
|-------------|------------|---|---|
| <b>KRVN</b> | Netbanking | Karur Vysya - Retail                          | Karur Vysya                                   |
| <b>KRVM</b> | Netbanking | Karur Vysya - Corporate                       | Karur Vysya - Corporate Netbanking            |
| <b>KKBN</b> | Netbanking | Kotak Mahindra Bank                           | Kotak Mahindra Bank                           |
| <b>LVBN</b> | Netbanking | Laxmi Vilas Bank                              | Laxmi Vilas Bank - Retail                     |
| <b>LVBM</b> | Netbanking | Laxmi Vilas Bank - Corporate                  | Laxmi Vilas Bank - Corporate                  |
| <b>MBSN</b> | Netbanking | The Mehsana Urban Co Op Bank Ltd              | The Mehsana Urban Co Op Bank Ltd.             |
| <b>NKBN</b> | Netbanking | NKGSB Bank                                    | NKGSB Bank                                    |
| <b>OBCN</b> | Netbanking | PNB (Erstwhile-Oriental Bank of Commerce)     | PNB (Erstwhile-Oriental Bank of Commerce)     |
| <b>PMCN</b> | Netbanking | Punjab & Maharashtra Coop Bank                | Punjab & Maharashtra Coop Bank                |
| <b>PSBN</b> | Netbanking | Punjab & Sind Bank                            | Punjab & Sind Bank                            |
| <b>PNBN</b> | Netbanking | Punjab National Bank - Retail                 | Punjab National Bank - Retail Banking         |
| <b>PNBM</b> | Netbanking | Punjab National Bank - Corporate              | Punjab National Bank-Corporate                |
| <b>RTNN</b> | Netbanking | RBL Bank Limited                              | RBL Bank Limited                              |
| <b>RTNM</b> | Netbanking | RBL Bank Limited- Corporate Net Banking       | RBL Bank Limited- Corporate Net Banking       |
| <b>SRSN</b> | Netbanking | Saraswat Bank                                 | Saraswat Bank                                 |
| <b>SVCN</b> | Netbanking | Shamrao Vitthal Co-operative Bank             | Shamrao Vitthal Co-operative Bank             |
| <b>SVCM</b> | Netbanking | Shamrao Vitthal Co-operative Bank - Corporate | Shamrao Vitthal Co-operative Bank - Corporate |
| <b>SOIN</b> | Netbanking | South Indian Bank                             | South Indian Bank                             |
| <b>SCBN</b> | Netbanking | Standard Chartered Bank                       | Standard Chartered Bank                       |
| <b>SBJN</b> | Netbanking | State Bank of Bikaner and Jaipur              | State Bank of Bikaner and Jaipur              |
| <b>SBHN</b> | Netbanking | State Bank of Hyderabad                       | State Bank of Hyderabad                       |
| <b>SBIN</b> | Netbanking | State Bank of India                           | State Bank of India                           |
| <b>SBMN</b> | Netbanking | State Bank of Mysore                          | State Bank of Mysore                          |
| <b>SBPN</b> | Netbanking | State Bank of Patiala                         | State Bank of Patiala                         |
| <b>SBTN</b> | Netbanking | State Bank of Travancore                      | State Bank of Travancore                      |
| <b>SYDN</b> | Netbanking | Syndicate Bank (Erstwhile Canara Bank)        | Syndicate Bank (Erstwhile Canara Bank)        |
| <b>TMBN</b> | Netbanking | Tamilnad Mercantile Bank Ltd.                 | Tamilnad Mercantile Bank Ltd.                 |
| <b>TSCN</b> | Netbanking | Tamilnadu State Coop Bank                     | Tamilnadu State Coop Bank                     |
| <b>TJSN</b> | Netbanking | TJSB Bank                                     | TJSB Bank                                     |
| <b>UCON</b> | Netbanking | UCO Bank                                      | UCO Bank                                      |
| <b>UBIM</b> | Netbanking | Union Bank of India - Corporate               | Union Bank - Corporate Netbanking             |
| <b>UBIN</b> | Netbanking | Union Bank of India - Retail                  | Union Bank of India                           |
| <b>UNIN</b> | Netbanking | PNB (Erstwhile-United Bank of India)          | PNB (Erstwhile-United Bank of India)          |
| <b>VIJN</b> | Netbanking | Vijaya Bank                                   | Vijaya Bank                                   |
| <b>YESN</b> | Netbanking | Yes Bank                                      | Yes Bank                                      |

|             |            |  |  |
|-------------|------------|--|--|
| <b>ESFN</b> | Netbanking | Equitas Bank   | Equitas Small Finance Bank Limited                           |
| <b>PNYN</b> | Netbanking | PNB YUVA Bank  | PNB YUVA Bank  |
| <b>KCCN</b> | Netbanking | The Kalupur Commercial Cooperative Bank Limited              | The Kalupur Commercial Cooperative Bank Limited              |
| <b>AIRN</b> | Netbanking | Airtel Payment Bank  | Airtel Payment Bank  |
| <b>BHAN</b> | Netbanking | Bharat Bank  | Bharat Bank  |
| <b>NBLN</b> | Netbanking | Nainital Bank  | Nainital Bank  |
| <b>DEMN</b> | Netbanking | Demo Bank  | Demo Netbanking  |
| <b>GPPN</b> | Netbanking | GP Parsik Sahakari Bank                                      | GP Parsik Sahakari Bank                                      |
| <b>AUSN</b> | Netbanking | AU Small Finance Bank  | AU Small Finance Bank  |
| <b>FESN</b> | Netbanking | Federal Bank Scan And Pay                                    | Federal Bank Scan And Pay                                    |
| <b>SUBN</b> | Netbanking | Suryoday Small Finance Bank                                  | Suryoday Small Finance Bank                                  |
| <b>VRBN</b> | Netbanking | Varachha Co-Operative Bank                                   | Varachha Co-Operative Bank                                   |
| <b>NEBN</b> | Netbanking | North East Small Finance Bank                                | North East Small Finance Bank                                |
| <b>IDBM</b> | Netbanking | IDBI Bank - Corporate  | Industrial Development Bank of India - Corporate             |
| <b>YESM</b> | Netbanking | Yes Bank - Corporate   | Yes Bank - Corporate   |
| <b>CRPM</b> | Netbanking | Union Bank of India (Erstwhile Corporation Bank) - Corporate | Union Bank of India (Erstwhile Corporation Bank) - Corporate |
| <b>ADCN</b> | Netbanking | Ahmedabad District Co-operative Bank                         | Ahmedabad District Co-operative Bank                         |
| <b>ESAN</b> | Netbanking | ESAF Small Finance Bank                                      | ESAF Small Finance Bank                                      |
| <b>BRLM</b> | Netbanking | Barclays Corporate Banking                                   | Barclays Corporate Banking                                   |
| <b>ZOBN</b> | Netbanking | Zoroastrian Co-operative Bank                                | Zoroastrian Co-operative Bank                                |
| <b>ABPN</b> | Netbanking | Aditya Birla Payments Bank                                   | Aditya Birla Payments Bank                                   |
| <b>ALLM</b> | Netbanking | Allahabad Bank Corporate Netbanking (Erstwhile Indian Bank)  | Allahabad Bank Corporate Netbanking (Erstwhile Indian Bank)  |
| <b>AXIM</b> | Netbanking | AXIS Bank - Corporate  | AXIS Bank - Corporate NetBanking                             |
| <b>ICIM</b> | Netbanking | ICICI Bank - Corporate                                       | ICICI Bank - Corporate NetBanking                            |
| <b>RBSN</b> | Netbanking | Royal Bank Of Scotland                                       | Royal Bank Of Scotland NetBanking                            |
| <b>JSFN</b> | Netbanking | Jana Small Finance   | Jana Small Finance NetBanking                                |
| <b>HSBN</b> | Netbanking | HSBC NetBanking  | HSBC NetBanking  |
| <b>USFN</b> | Netbanking | Ujjivan Small Finance Bank                                   | Ujjivan Small Finance NetBanking                             |
| <b>FSFN</b> | NetBanking | Fincare Small Finance Bank                                   | Fincare Small Finance Bank                                   |
| <b>BDNM</b> | Netbanking | Bandhan bank - Corporate                                     | Bandhan bank - Corporate                                     |
| <b>APGN</b> | Netbanking | Andhra Pragathi Grameena Bank                                | Andhra Pragathi Grameena Bank                                |
| <b>BCCN</b> | Netbanking | Bassien Catholic Coop Bank                                   | Bassien Catholic Coop Bank                                   |
| <b>CPFN</b> | Netbanking | Capital Small Finance Bank                                   | Capital Small Finance Bank                                   |
| <b>GSCN</b> | Netbanking | Gujarat State Co-operative Bank Limited                      | Gujarat State Co-operative Bank Limited                      |
| <b>JPBN</b> | Netbanking | Jio Payments Bank  | Jio Payments Bank  |
| <b>KVGN</b> | NetBanking | Karnataka Vikas Grameena Bank                                | Karnataka Vikas Grameena Bank                                |

|             |            |   |   |
|-------------|------------|---|---|
| <b>MGBN</b> | Netbanking | Maharashtra Gramin Bank                         | Maharashtra Gramin Bank                         |
| <b>SOMN</b> | Netbanking | SBM Bank India                                  | SBM Bank India                                  |
| <b>SPCN</b> | Netbanking | The Surat People's Co-operative Bank Limited    | The Surat People's Co-operative Bank Limited    |
| <b>SUCN</b> | Netbanking | The Sutex Co-op Bank Ltd                        | The Sutex Co-op Bank Ltd                        |
| <b>TBSN</b> | Netbanking | Thane Bharat Sahakari Bank Ltd                  | Thane Bharat Sahakari Bank Ltd                  |
| <b>UTSN</b> | Netbanking | Utkarsh Small Finance Bank                      | Utkarsh Small Finance Bank                      |
| <b>VCBN</b> | Netbanking | The Varachha Co-operative Bank                  | The Varachha Co-operative Bank                  |
| <b>PYTN</b> | Netbanking | Paytm Bank                                      | Paytm Net Banking                               |
| <b>KGBN</b> | NetBanking | Kaveri Grameena Bank                            | Kaveri Grameena Bank                            |
| <b>NNBN</b> | Netbanking | Nutan Nagarik Sahakari Bank Limited             | Nutan Nagarik Sahakari Bank Limited             |
| <b>SMCN</b> | Netbanking | Shivalik Mercantile Cooperative Bank Ltd        | Shivalik Mercantile Cooperative Bank Ltd        |
| <b>ABAN</b> | Netbanking | ABN Amro Bank                                   | ABN Amro Bank                                   |
| <b>MCBN</b> | NetBanking | The Mugberia Central Co-operative Bank Ltd      | The Mugberia Central Co-operative Bank Ltd      |
| <b>ACCN</b> | NetBanking | The Akola District Central Cooperative Bank Ltd | The Akola District Central Cooperative Bank Ltd |
| <b>PGBN</b> | NetBanking | Pragathi Gramin Bank                            | Pragathi Gramin Bank                            |
| <b>SGBN</b> | NetBanking | Shreyas Gramin Bank                             | Shreyas Gramin Bank                             |
| <b>KABN</b> | Netbanking | Karnataka Gramin Bank                           | Karnataka Gramin Bank                           |
| <b>PDBN</b> | Netbanking | Pandyan Grama Bank                              | Pandyan Grama Bank                              |
| <b>SSSN</b> | Netbanking | SUCO SOUHARDA SAHAKARI BANK                     | SUCO SOUHARDA SAHAKARI BANK                     |
| <b>MDCN</b> | Netbanking | MALDA DISTRICT CENTRAL COOPERATIVE BANK LTD     | MALDA DISTRICT CENTRAL COOPERATIVE BANK LTD     |
| <b>OGBN</b> | Netbanking | Odisha Gramya Bank                              | Odisha Gramya Bank                              |
| <b>BGVN</b> | Netbanking | Bangiya Gramin Vikash Bank                      | Bangiya Gramin Vikash Bank                      |
| <b>MRBN</b> | Netbanking | MANIPUR RURAL BANK                              | MANIPUR RURAL BANK                              |
| <b>AGVN</b> | Netbanking | Assam Gramin Vikash Bank                        | Assam Gramin Vikash Bank                        |
| <b>TGBN</b> | Netbanking | Tripura Gramin Bank                             | Tripura Gramin Bank                             |
| <b>TGCN</b> | Netbanking | Tamluk-Ghatal Central Co-operative Bank Ltd     | Tamluk-Ghatal Central Co-operative Bank Ltd     |
| <b>SDCN</b> | Netbanking | The Surat District Co-Op Bank                   | The Surat District Co-Op Bank                   |
| <b>WUCN</b> | Netbanking | The Washim Urban Co-operative Bank Ltd          | The Washim Urban Co-operative Bank Ltd          |
| <b>AMCN</b> | Netbanking | THE AHMEDABAD MERCANTILE CO-OP BANK LTD         | THE AHMEDABAD MERCANTILE CO-OP BANK LTD         |
| <b>PCBN</b> | Netbanking | Prime Co-Operative Bank Ltd                     | Prime Co-Operative Bank Ltd                     |
| <b>HGBN</b> | Netbanking | Haryana Gramin Bank                             | Haryana Gramin Bank                             |
| <b>PLGN</b> | Netbanking | Pallavan Grama Bank                             | Pallavan Grama Bank                             |
| <b>UUCN</b> | Netbanking | The Udaipur Urban Co-op Bank Ltd                | The Udaipur Urban Co-op Bank Ltd                |

|             |            |   |   |
|-------------|------------|---|---|
| <b>KDCN</b> | Netbanking | The Kasaragod District Co-operative bank Ltd      | The Kasaragod District Co-operative bank Ltd      |
| <b>RCCN</b> | Netbanking | Raiganj Central Co op Bank Ltd                    | Raiganj Central Co op Bank Ltd                    |
| <b>PSCN</b> | Netbanking | THE PUNJAB STATE COOPERATIVE BANK LTD             | THE PUNJAB STATE COOPERATIVE BANK LTD             |
| <b>JNSN</b> | Netbanking | Jodhpur Nagrik Sahakari Bank Limited              | Jodhpur Nagrik Sahakari Bank Limited              |
| <b>FCCN</b> | Netbanking | THE FARIDKOT CENTRAL CO-OPERATIVE BANK LTD        | THE FARIDKOT CENTRAL CO-OPERATIVE BANK LTD        |
| <b>ISBN</b> | Netbanking | Indraprastha Sehkari Bank Ltd                     | Indraprastha Sehkari Bank Ltd                     |
| <b>MPCN</b> | Netbanking | The Malappuram District Co-operative Bank Limited | The Malappuram District Co-operative Bank Limited |
| <b>SNSN</b> | Netbanking | Smriti Nagrik Sahakari Bank                       | Smriti Nagrik Sahakari Bank                       |
| <b>JCCN</b> | Netbanking | THE JALANDHAR CENTRAL COOPERATIVE BANK LIMITED    | THE JALANDHAR CENTRAL COOPERATIVE BANK LIMITED    |
| <b>LCCN</b> | Netbanking | THE LUDHIANA CENTRAL COOPERATIVE BANK LTD         | THE LUDHIANA CENTRAL COOPERATIVE BANK LTD         |
| <b>NJGN</b> | Netbanking | Narmada Jhabua Gramin Bank                        | Narmada Jhabua Gramin Bank                        |
| <b>MUCN</b> | Netbanking | Shree Mahalaxmi Urban Co-op Credit Bank Ltd       | Shree Mahalaxmi Urban Co-op Credit Bank Ltd       |
| <b>JUCN</b> | Netbanking | The Junagadh Commercial Co-op Bank Ltd            | The Junagadh Commercial Co-op Bank Ltd            |
| <b>SCCN</b> | Netbanking | THE SATARA DISTRICT CENTRAL CO-OPERATIVE BANK LTD | THE SATARA DISTRICT CENTRAL CO-OPERATIVE BANK LTD |
| <b>TDCN</b> | Netbanking | Thrissur District Cooperative Bank Ltd            | Thrissur District Cooperative Bank Ltd            |
| <b>FZCN</b> | Netbanking | THE FAZILKA CENTRAL COOP. BANK LTD                | THE FAZILKA CENTRAL COOP. BANK LTD                |
| <b>MCCN</b> | Netbanking | Mumbai District Central Co-op Bank Ltd            | Mumbai District Central Co-op Bank Ltd            |
| <b>BOIM</b> | Netbanking | Bank Of India - Corporate                         | Bank Of India - Corporate                         |
| <b>CUBM</b> | Netbanking | City Union Bank - Corporate                       | City Union Bank - Corporate                       |
| <b>HDFM</b> | Netbanking | HDFC Bank - Corporate                             | HDFC Bank - Corporate                             |
| <b>IOBM</b> | Netbanking | Indian Overseas Bank - Corporate                  | Indian Overseas Bank - Corporate                  |
| <b>SBIM</b> | Netbanking | State Bank of India - Corporate                   | State Bank of India - Corporate                   |
| <b>BBBN</b> | Netbanking | Bank of Baroda - Retail(B2B)                      | Bank of Baroda Retail Banking(B2B)                |
| <b>BBBM</b> | Netbanking | Bank of Baroda - Corporate(B2B)                   | Bank of Baroda Corporate Banking(B2B)             |
| <b>HSBE</b> | EMI        | Hong Kong And Shanghai Banking Credit EMI         | Hong Kong And Shanghai Banking Credit EMI         |
| <b>YESE</b> | EMI        | Yes Bank Credit EMI                               | Yes Bank Credit EMI                               |
| <b>SCBE</b> | EMI        | Standard Chartered Bank Credit EMI                | Standard Chartered Bank Credit EMI                |
| <b>AXIE</b> | EMI        | AXIS Bank Credit EMI                              | AXIS Bank Credit EMI                              |
| <b>RBLE</b> | EMI        | RBL Bank Credit EMI                               | RBL Bank Credit EMI                               |
| <b>HDFE</b> | EMI        | HDFC Bank Credit EMI                              | HDFC Bank Credit EMI                              |

|             |            |   |   |
|-------------|------------|---|---|
| <b>CITE</b> | EMI        | CITIBANK Credit EMI                     | CITIBANK Credit EMI                     |
| <b>KKBE</b> | EMI        | Kotak Mahindra Bank Credit EMI          | Kotak Mahindra Bank Credit EMI          |
| <b>INDE</b> | EMI        | IndusInd Bank Credit EMI                | IndusInd Bank Credit EMI                |
| <b>AMXE</b> | EMI        | American Express Credit EMI             | American Express Credit EMI             |
| <b>ICIE</b> | EMI        | ICICI Bank Credit EMI                   | ICICI Bank Credit EMI                   |
| <b>SBIE</b> | EMI        | State Bank of India Credit EMI          | State Bank of India Credit EMI          |
| <b>BOBE</b> | EMI        | Bank of Baroda Credit EMI               | Bank of Baroda Credit EMI               |
| <b>AXDE</b> | EMI        | AXIS BANK DEBIT EMI                     | AXIS Bank Debit EMI                     |
| <b>HDDE</b> | EMI        | HDFC Bank Debit EMI                     | HDFC Bank Debit EMI                     |
| <b>BODE</b> | EMI        | Bank of Baroda Debit EMI                | Bank of Baroda Debit EMI                |
| <b>ICDE</b> | EMI        | ICICI Bank Debit EMI                    | ICICI Bank Debit EMI                    |
| <b>KKDE</b> | EMI        | Kotak Mahindra Bank Debit EMI           | Kotak Mahindra Bank Debit EMI           |
| <b>SBDE</b> | EMI        | State Bank of India Debit EMI           | State Bank of India Debit EMI           |
| <b>KKBL</b> | Direct EMI | Kotak Mahindra Bank Credit EMI          | Kotak Mahindra Bank Credit EMI          |
| <b>BOBL</b> | Direct EMI | Bank of Baroda Credit EMI               | Bank of Baroda Credit EMI               |
| <b>ICIL</b> | Direct EMI | ICICI Bank Credit EMI                   | ICICI Bank Credit EMI                   |
| <b>SBIL</b> | Direct EMI | State Bank of India Credit EMI          | State Bank of India Credit EMI          |
| <b>AXDL</b> | Direct EMI | AXIS BANK DEBIT EMI                     | AXIS Bank Debit EMI                     |
| <b>AXIL</b> | Direct EMI | AXIS Bank Credit EMI                    | AXIS Bank Credit EMI                    |
| <b>HDFL</b> | Direct EMI | HDFC BANK Credit EMI                    | HDFC Bank Credit EMI                    |
| <b>ICDL</b> | Direct EMI | ICICI Bank Debit EMI                    | ICICI Bank Debit EMI                    |
| <b>HDDL</b> | Direct EMI | HDFC Bank Debit EMI                     | HDFC Bank Debit EMI                     |
| <b>AUIL</b> | Direct EMI | AU Small Finance Bank Credit EMI        | AU Small Finance Bank Credit EMI        |
| <b>RBLL</b> | Direct EMI | RBL Bank Credit Card EMI                | RBL Bank Credit Card EMI                |
| <b>SCBL</b> | Direct EMI | Standard Chartered Bank Credit Card EMI | Standard Chartered Bank Credit Card EMI |
| <b>YESL</b> | Direct EMI | Yes Bank Credit Card EMI                | Yes Bank Credit Card EMI                |
| <b>BOIA</b> | ATM Card   | Bank of India                           | Bank of India ATM Card                  |
| <b>PNBA</b> | ATM Card   | Punjab National Bank                    | Punjab National Bank ATM Card           |
| <b>BOBA</b> | ATM Card   | Bank of Baroda                          | Bank of Baroda ATM Card                 |
| <b>BMBA</b> | ATM Card   | Bharatiya Mahila Bank                   | Bharatiya Mahila Bank ATM Card          |
| <b>BOMA</b> | ATM Card   | Bank of Maharashtra                     | Bank of Maharashtra ATM Card            |
| <b>CANA</b> | ATM Card   | Canara Bank                             | Canara Bank ATM Card                    |
| <b>INDA</b> | ATM Card   | IndusInd Bank                           | IndusInd Bank ATM Card                  |
| <b>IDFA</b> | ATM Card   | IDFC                                    | IDFC ATM Card                           |
| <b>IOBA</b> | ATM Card   | Indian Overseas Bank                    | Indian Overseas Bank ATM Card           |
| <b>KGBA</b> | ATM Card   | Kerala Gramin Bank                      | Kerala Gramin Bank ATM Card             |
| <b>LVBA</b> | ATM Card   | Laxmi Vilas Bank                        | Laxmi Vilas Bank ATM Card               |
| <b>PKGA</b> | ATM Card   | Pragathi Krishna Gramin Bank            | Pragathi Krishna Gramin Bank ATM Card   |
| <b>RSCA</b> | ATM Card   | Rajasthan State Coop Bank               | Rajasthan State Coop Bank ATM Card      |

|             |              |  |   |
|-------------|--------------|--|---|
| <b>SMCA</b> | ATM Card     | Shivalik Mercantile Co-Op Bank Limited | Shivalik Mercantile Co-Op Bank Limited ATM Card |
| <b>UBIA</b> | ATM Card     | Union Bank of India                    | Union Bank of India ATM Card                    |
| <b>ITZH</b> | Cash Card    | ITZ Cash Card                          | ITZ Cash card                                   |
| <b>ICSH</b> | Cash Card    | Icash Card                             | Icash card                                      |
| <b>DONH</b> | Cash Card    | DONE Cash Card                         | DONE Cash Card                                  |
| <b>PWMH</b> | Cash Card    | PayWorld Cash Card                     | PayWorld Cash Card                              |
| <b>PTMW</b> | Wallet       | Paytm                                  | Paytm wallet                                    |
| <b>MBKW</b> | Wallet       | Mobikwik                               | Mobikwik Wallet                                 |
| <b>OXIW</b> | Wallet       | Oxigen                                 | Oxigen Wallet                                   |
| <b>MRPW</b> | Wallet       | mRupee                                 | mRupee Wallet                                   |
| <b>JIOW</b> | Wallet       | Reliance JioMoney                      | Reliance Jio Money wallet                       |
| <b>TMWW</b> | Wallet       | The Mobile Wallet                      | The Mobile Wallet                               |
| <b>PYCW</b> | Wallet       | Paycash                                | Paycash wallet                                  |
| <b>OLAW</b> | Wallet       | Ola Money                              | Ola Money Wallet                                |
| <b>JNCW</b> | Wallet       | Jana Cash                              | Jana Cash                                       |
| <b>ATLW</b> | Wallet       | Airtel Money                           | Airtel Money                                    |
| <b>PNBW</b> | Wallet       | PNB Wallet                             | PNB Wallet                                      |
| <b>FRCW</b> | Wallet       | FreeCharge                             | FreeCharge Wallet                               |
| <b>EZCW</b> | Wallet       | ezeClick                               | ezeClick Wallet                                 |
| <b>VDFW</b> | Wallet       | mPesa Wallet                           | mPesa - Vodafone Wallet                         |
| <b>SBIW</b> | Wallet       | SBI Buddy                              | SBI Buddy                                       |
| <b>AMPW</b> | Wallet       | Amazon Pay                             | Amazon Pay                                      |
| <b>PZAW</b> | Wallet       | Payzapp                                | Payzapp wallet                                  |
| <b>IDMW</b> | Wallet       | Idea Money                             | Idea Money                                      |
| <b>CITW</b> | Wallet       | Citibank Reward Points                 | Citibank Reward Points                          |
| <b>YPAW</b> | Wallet       | YpayCash                               | YpayCash  |
| <b>ZIPW</b> | Wallet       | ZipCash                                | ZipCash   |
| <b>YESW</b> | Wallet       | Yes Pay                                | Yes Pay   |
| <b>BKSW</b> | Wallet       | bKash Wallet                           | bKash Wallet                                    |
| <b>DCBW</b> | Wallet       | DCB Cippy                              | DCB Cippy                                       |
| <b>PHPW</b> | Wallet       | PhonePe                                | PhonePe wallet                                  |
| <b>AMRW</b> | Wallet       | Airtel Money Rwanda                    | Airtel Money Rwanda                             |
| <b>MTRW</b> | Wallet       | MTN Rwanda                             | MTN Rwanda                                      |
| <b>VISP</b> | Prepaid Card | Visa                                   | Visa Prepaid Card                               |
| <b>MASP</b> | Prepaid Card | Master                                 | Master Prepaid Card                             |
| <b>RUPP</b> | Prepaid Card | Rupay                                  | Rupay Prepaid Card                              |
| <b>ZEMF</b> | Cardless EMI | ZestMoney                              | ZestMoney                                       |
| <b>INCF</b> | Cardless EMI | Instacred                              | Instacred                                       |
| <b>EASF</b> | Cardless EMI | EarlySalary                            | EarlySalary                                     |
| <b>BHQR</b> | Bharat QR    | Bharat QR                              | Bharat QR Payments(Dynamic)                     |

|             |                        |                            |                             |
|-------------|------------------------|----------------------------|-----------------------------|
| <b>BSQR</b> | Bharat QR(Static)      | Bharat QR(Static)          | Bharat QR Payments(Static)  |
| <b>ECLV</b> | E-Collect              | E-Collect                  | E-Collect(Virtual Accounts) |
| <b>UPIU</b> | UPI                    | Unified Payments Interface | Unified Payments Interface  |
| <b>LYWA</b> | WhatsApp               | Lyra WhatsApp              | Lyra WhatsApp               |
| <b>EZP1</b> | EazyPay                | EazyPay                    | EazyPay                     |
| <b>PYM1</b> | Pharmarack Credit Card | Pharmarack                 | Pharmarack Credit Card      |

## 17 APPENDIX 4 - List of error codes

| Error Numeric Code | Error Code                  | Error Description   |
|--------------------|-----------------------------|---|
| <b>0</b>           | SUCCESS                     | Transaction successful  |
| <b>1000</b>        | FAILED                      | Transaction failed  |
| <b>1001</b>        | INVALID-API-KEY             | The api key field is incorrect  |
| <b>1002</b>        | INVALID-LIVE-MODE-ACCESS    | The live mode access is not allowed   |
| <b>1003</b>        | INVALID-ORDER-ID-FIELD      | The order id field should to be unique                                      |
| <b>1004</b>        | ORDER-ID-FIELD-NOT-FOUND    | The order id field is not found   |
| <b>1005</b>        | INVALID-AUTHENTICATION      | Invalid authentication at bank  |
| <b>1006</b>        | WAITING-BANK-RESPONSE       | Waiting for the response from bank  |
| <b>1007</b>        | INVALID-INPUT-REQUEST       | Invalid input in the request message  |
| <b>1008</b>        | TRANSACTION-TAMPERED        | Transaction tampered  |
| <b>1009</b>        | DECLINED-BY-BANK            | Bank Declined Transaction   |
| <b>1010</b>        | INVALID-AMOUNT              | Amount cannot be less than 1  |
| <b>1011</b>        | AUTHORIZATION-REFUSED       | Authorization refused   |
| <b>1012</b>        | INVALID-CARD                | Invalid Card/Member Name data   |
| <b>1013</b>        | INVALID-EXPIRY-DATE         | Invalid expiry date   |
| <b>1014</b>        | DENIED-BY-RISK              | Transaction denied by risk  |
| <b>1015</b>        | INSUFFICIENT-FUND           | Insufficient Fund   |
| <b>1016</b>        | INVALID-AMOUNT-LIMIT        | Total Amount limit set for the terminal for transactions has been crossed   |
| <b>1017</b>        | INVALID-TRANSACTION-LIMIT   | Total transaction limit set for the terminal has been crossed               |
| <b>1018</b>        | INVALID-DEBIT-AMOUNT-LIMIT  | Maximum debit amount limit set for the terminal for a day has been crossed  |
| <b>1019</b>        | INVALID-CREDIT-AMOUNT-LIMIT | Maximum credit amount limit set for the terminal for a day has been crossed |
| <b>1020</b>        | MAXIMUM-DEBIT-AMOUNT-CROSS  | Maximum debit amount set for per card for rolling 24 hrs has been crossed   |
| <b>1021</b>        | MAXIMUM-CREDIT-AMOUNT-CROSS | Maximum credit amount set for per card for rolling 24 hrs has been crossed  |
| <b>1022</b>        | MAXIMUM-TRANSACTION-CROSS   | Maximum transaction set for per card for rolling 24 hrs has been crossed    |
| <b>1023</b>        | HASH-MISMATCH               | Hash Mismatch   |
| <b>1024</b>        | INVALID-PARAMS              | Invalid parameters  |
| <b>1025</b>        | INVALID-BANK-CODE           | Invalid bank code   |
| <b>1026</b>        | INVALID-MERCHANT            | Merchant is not active  |
| <b>1027</b>        | INVALID-TRANSACTION         | Invalid transaction   |
| <b>1028</b>        | TRANSACTION-NOT-FOUND       | Transaction not found   |
| <b>1029</b>        | TRANSACTION-TERMINATED      | Transaction terminated  |
| <b>1030</b>        | TRANSACTION-INCOMPLETE      | Transaction incomplete  |

|             |                                      |  |
|-------------|--------------------------------------|--|
| <b>1031</b> | AUTO-REFUNDED                        | Transaction auto refunded  |
| <b>1032</b> | REFUNDED                             | Transaction refunded   |
| <b>1033</b> | SINGLE-TRANSACTION-LOWER-LIMIT-CROSS | The amount provided is less than transaction lower limit           |
| <b>1034</b> | SINGLE-TRANSACTION-UPPER-LIMIT-CROSS | The amount provided is more than transaction upper limit           |
| <b>1035</b> | TRANSACTION-DAILY-LIMIT-CROSS        | The daily transaction limit is exceeded for the merchant           |
| <b>1036</b> | TRANSACTION-MONTHLY-LIMIT-CROSS      | The monthly transaction limit is exceeded for the merchant         |
| <b>1037</b> | DAILY-TRANSACTION-NUMBER-CROSS       | The daily transaction number is exceeded for the merchant          |
| <b>1038</b> | MONTHLY-TRANSACTION-NUMBER-CROSS     | The monthly transaction number is exceeded for the merchant        |
| <b>1039</b> | INVALID-REFUND-AMOUNT                | The refund amount is greater than transaction amount               |
| <b>1040</b> | INVALID-CVV                          | Invalid Card Verification Code                                     |
| <b>1041</b> | AUTO-REFUNDED-TNP                    | Transaction is auto refunded by TnP                                |
| <b>1042</b> | FAILED-NO-RESPONSE                   | Transaction failed as there was no response from bank              |
| <b>1043</b> | TRANSACTION-CANCELLED                | Transaction cancelled  |
| <b>1044</b> | UNAUTHORIZED                         | Unauthorized   |
| <b>1045</b> | FORBIDDEN                            | Forbidden Access   |
| <b>1046</b> | TRANSACTION-ALREADY-CAPTURED         | Transaction already captured                                       |
| <b>1047</b> | AUTHORIZED                           | Transaction authorized   |
| <b>1048</b> | CAPTURED                             | Transaction captured   |
| <b>1049</b> | VOIDED                               | Transaction voided   |
| <b>1050</b> | NO-RECORD-FOUND                      | No data record found for the given input                           |
| <b>1051</b> | ACQUIRER-ERROR                       | Error occurred at the bank end                                     |
| <b>1052</b> | INVALID-EMAIL                        | Invalid Email ID   |
| <b>1053</b> | INVALID-PHONE                        | Invalid phone number   |
| <b>1054</b> | SESSION-TIMEOUT                      | Session expired. Please go back and try again.                     |
| <b>1055</b> | SEAMLESS-NOT-ALLOWED                 | Seamless payment not allowed                                       |
| <b>1056</b> | INVALID-VPA                          | Virtual Payee Address is invalid                                   |
| <b>1057</b> | 3DS-FAILED                           | 3D Secure authentication failed                                    |
| <b>1058</b> | ACCOUNT-BLOCKED                      | Bank Account is blocked  |
| <b>1059</b> | ACQUIRER-MAX-TRANSACTION-LIMIT       | Transaction amount limit has exceeded                              |
| <b>1060</b> | ACQUIRER-MIN-TRANSACTION-LIMIT       | The amount provide is less than minimum transaction amount allowed |
| <b>1061</b> | ACQUIRER-VELOCITY-LIMIT              | Transaction frequency limit has exceeded                           |
| <b>1062</b> | CARD-EXPIRED                         | Card has expired   |
| <b>1063</b> | HOTLISTED                            | Stolen or Lost Card  |
| <b>1064</b> | INACTIVE-CARD                        | The card is inactive   |

|             |                            |   |
|-------------|----------------------------|---|
| <b>1065</b> | INCORRECT-PIN              | Incorrect PIN   |
| <b>1066</b> | INSUFFICIENT-BALANCE       | Insufficient Balance  |
| <b>1068</b> | INVALID-BIN                | Bin not found   |
| <b>1069</b> | INVALID-CARDHOLDER-NAME    | Invalid card holder name  |
| <b>1070</b> | INVALID-CARD-INFO          | Invalid brand or bin range not enabled  |
| <b>1071</b> | INVALID-INPUT-DATA         | Invalid input data  |
| <b>1072</b> | ISSUER-DECLINE             | Payment declined by bank  |
| <b>1073</b> | ISSUER-DISALLOWED          | Transaction not allowed by the issuer   |
| <b>1074</b> | ISSUER-ERROR               | Payment processing failed due to error at bank                                |
| <b>1075</b> | ISSUER-LIMIT-DECLINE       | Exceeds withdrawal frequency or count limit                                   |
| <b>1076</b> | MAX-RETRIES                | Maximum number of PIN retries exceeded  |
| <b>1077</b> | OTP-LIMIT-EXCEEDED         | OTP validation attempts limit exceeded  |
| <b>1078</b> | PIN-ATTEMPTS-EXCEEDED      | PIN attempts limit exceeded   |
| <b>1079</b> | RESTRICTED-CARD            | Restricted card   |
| <b>1080</b> | SUSPECTED-FRAUD            | Suspected fraud   |
| <b>1081</b> | TIMEOUT                    | Payment was not completed on time   |
| <b>1082</b> | UPI-PIN-NOT-SET            | UPI PIN is not set  |
| <b>1083</b> | UPI-REQUEST-EXPIRED        | UPI request expired   |
| <b>1084</b> | USER-BACK-REFRESH          | User pressed refresh button   |
| <b>1085</b> | ACQUIRER-DOWN              | Payment acquirer is down  |
| <b>1086</b> | TOO-MANY-RECORDS           | Response contains too many records, use pagination                            |
| <b>1087</b> | NOT-ENOUGH-PARAMETERS      | Not enough parameters to get response, add one or more optional parameters    |
| <b>1088</b> | TRANSACTION-IN-PROCESS     | We are processing your transaction  |
| <b>1089</b> | INVALID-ACCESS-TOKEN       | Invalid access token  |
| <b>1090</b> | TEST-MODE-DISABLED         | The test mode access is not allowed   |
| <b>1091</b> | INVALID-TRANSFER-TYPE      | The disbursement transfer type is not allowed                                 |
| <b>1092</b> | INACTIVE-ACCOUNT           | The customers account is inactive   |
| <b>1093</b> | INVALID-ACCOUNT            | The customers account is invalid  |
| <b>1094</b> | TRANSACTION-COUNT-EXCEEDED | Transaction count limit has exceeded for the customer                         |
| <b>1095</b> | AMOUNT-MISMATCH            | Transaction amount mismatched may be due to a attempted tampering in response |
| <b>9999</b> | UNKNOWN-ERROR              | Unknown error occurred  |

## 18 APPENDIX 5 – Currency Codes

| Currency   | Currency Code   |
|------------|-----------------|
| <b>INR</b> | Indian Rupee    |
| <b>USD</b> | US Dollar       |
| <b>EUR</b> | Euro            |
| <b>AED</b> | UAE Dirham      |
| <b>GBP</b> | Pound Sterling  |
| <b>MUR</b> | Mauritius Rupee |
| <b>RWF</b> | Rwanda Franc    |
| <b>LKR</b> | Sri Lanka Rupee |