SparkIT·Pay

Go Digital Go Cashless

# PAYMENT GATEWAY

**2017 APIs for integration**

Contact
**Email: mail@sparkitts.com**
**Website: www.sparkitpay.com**

Document version 1.7.3

# Contents

# 1. OVERVIEW

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

Through SparkIT Pay, 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 debit

SparkIT Pay also offers you a business UI (https://biz.sparkitpay.com) where you have access to all your prior transaction/payment details, settlement details, analytics, etc.

You can also use this UI to create invoices singly or in bulk, set reminders, recurring billing, and many more features.

Through this interface, you can also cancel past invoices (and in some cases, past transactions), manage your payables, vendor payments, set split ratios for vendor payments, process refunds, etc. This online interface can be accessed through https://biz.sparkitpay.com.

## 2. PAYMENT REQUEST API

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

## 2.1. Steps for Integration

- Initially your transaction limit would be set to a fixed amount (such as Rs. 2.34) and the said limit will be increased after a few successful test transactions.

- You need to submit a **POST REQUEST** to our server, at the below mentioned URL
  https://biz.sparkitpay.com/v2/paymentrequest
  **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 SparkIT Pay's payment page. After the customer makes the payment through SparkIT Pay (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 SparkIT Pay 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 SparkIT Pay, 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". We strongly recommend that you maintain uniqueness of your order IDs, to avoid confusion and conflicts while retrieving transaction details subsequently.

## 2.2. Parameters to be POSTed in Payment Request

*URL: https://biz.sparkitpay.com/v2/paymentrequest*

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| 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(15,2) | 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 – SparkIT Pay 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 – SparkIT Pay will make a POST request to this URL after a FAILED | varchar(255) | optional |

| | transaction, with a set of parameters, which you can process as you want to. | | |
|---|---|---|---|
| return_url_cancel | Return URL success – SparkIT Pay 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) | optional |
| payment_options | payment options to be displayed such credit card(cc), netbanking(nb), wallet(w) and atm card(atm).Tabs will be displayed by order in which values are sent. Values accepted are: cc,nb,w,atm (comma separated string) | varchar(50) | optional |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2. Note: the SALT will be provided by SparkIT Pay separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package | varchar(255) | mandatory |

## 2.3. Response Parameters returned by SparkIT Pay

| Parameter name | Description |
|---|---|
| transaction_id | A unique ID that can be used to trace the transaction uniquely within SparkIT Pay. 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 "YYYY-MM-DD HH:MM:SS" format |
| response_code | Status of the transaction (return code). 0 signifies successful transaction. Non-zero values signify error. |
| 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. |
| 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 |
| tate | 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 | Masked card number which was used to make the transaction. For example, 437748******0069<br>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 SparkIT Pay relationship manager if you would like this information to be returned to you. |
| hash | SparkIT Pay 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:** Consider, a failed response sent from SparkIT Pay server to your server via a user's browser and user chares the response code to Success, even though transaction is failed it will now show Success on your website. To make sure the transaction response is same as what SparkIT Pay server sent please check the hash before considering the transaction response success or failure.

**Note:** Format of transaction ID is as follows: "HDVISC1299876438". The 3rd to 6th digits (both inclusive) in the transaction ID signify the "bankcode". This information is sufficient 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. PAYMENT STATUS API

SparkIT Pay 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.

SparkIT Pay 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 SparkIT Pay for this API to work. If you receive errors such as "Unauthorized" while accessing this API, please contact your SparkIT Pay relationship manager to get this fixed.

*URL:* *https://biz.sparkitpay.com/v2/paymentstatus*

## 3.1 Parameters to be POSTed

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| order_id | This is your (merchant) reference number which you submitted while making the original transaction. | varchar(30) | Optional |
| transaction_id | This is the transaction ID generated by SparkIT Pay 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 SparkIT Pay 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 YYYY-MM-DD | varchar(10) | Optional |
| date_to | End date of date range to retrieve transactions, in YYYY-MM-DD | varchar(10) | Optional |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package | varchar(255) | Mandatory |

## 3.2. 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. A few sample responses for given requests are provided below:

In case of success,

```
{
 "data": [
  {
   "transaction_id": "HDVISC4180885759",
   "bank_code": "VISC",
   "payment_mode": "Credit Card",
   "payment_channel": "Visa",
   "payment_datetime": "2017-03-10 14:14:12",
   "response_code": 0,
   "response_message": "SUCCESS",
   "order_id": "8329",
   "amount": "7.54",
   "amount_orig": "4.00",
   "tdr_amount": "3.54",
   "tax_on_tdr_amount": "0.00",
   "description": "Web Payment for 9311",
   "error_desc": null,
   "customer_phone": "9535640653",
```

*"customer_name": "Sharath",*
*"customer_email": "Sharathkumar.harishi@gmail.com"*
*}*
*],*
*"hash":*
*"2A8532ACB303ED62713B9684D44CFFA11BE05709B15FD7ED8EA1E3863B1C26D23A06E36B0B3C2227D*
*93EE077AD67788E63557AD4684BB5DF76D14614AF6B4395"*
*}*

In case of error,

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

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"*
*}*
*}*

# 4. Refunds API

SparkIT Pay provides a refund API which merchants can use to programmatically issue refunds instead of clicking the "refund" button in the SparkIT Pay 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.

## 4.1. Refund request API

*URL: https://biz.sparkitpay.com/v2/refundrequest*

**Request Parameters:**

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| transaction_id | The unique alphanumeric transaction ID generated by SparkIT Pay for a prior transaction. | varchar(30) | Mandatory |
| 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 SparkIT Pay, for details about calculating hash refer Appendix 2. | varchar(255) | Mandatory |

| | Note: the SALT will be provided by SparkIT Pay separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package | | |
|---|---|---|---|

**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.

In case of success,

```
{
  "data": {
    "refund_id":5621,
  }
}
```

In case of error,

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

# 5. SPLIT API

## 5.1 SPLIT SETTLEMENT API

### 5.1.1 Split transaction before settlement API
*URL:* **https://biz.sparkitpay.com/v2/splitsettlementrequest**


**Request Parameters:**

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | The unique key provided to the merchant | varchar(40) | 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 SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package<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."
    }
}
```

# 6. VENDOR API

## 6.1. Add Vendor API

*URL: https://biz.sparkitpay.com/v1/addvendor*

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

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

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| vendor_code | This is the vendor code that you wish to add in the SparkIT Pay system. This has to be unique. Aplhanumeric 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. Optional. | varchar(300) | optional |
| account_name | Account holder name (of the vendor bank account) | varchar(300) | mandatory |
| account_number | Account number of the vendor. | varchar(50) | mandatory |
| ifsc_code | IFSC code of the vendor's bank. | varchar(50) | mandatory |
| bank_name | Bank name of the vendor's bank. | varchar(200) | optional |
| bank_branch | Bank branch of the vendor's bank. | varchar(300) | 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 |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2. | varchar(200) | mandatory |

| | Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package<br><br>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. | | |
| --- | --- | --- | --- |

**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": {
        "message": "Vendor added successfully"
    }
}
```

In case vendor already exists
```
{
    "error": {
        "code": 1024,
        "message": "Vendor code already exists"
    }
}
```

## 6.2. Modify Vendor API

*URL:  https://biz.sparkitpay.com/v1/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.

| Parameter Name | Description | Data type | Optional/ Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| vendor_code | This is the vendor code that you wish to modify in the SparkIT Pay system. This value must already exist in the system, failing which SparkIT Pay 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_name | Account holder name (of the vendor bank account) | varchar(300) | Optional |
| account_number | Account number of the vendor. | varchar(50) | Optional |
| ifsc_code | IFSC code of the vendor's bank. | varchar(50) | Optional |
| bank_name | Bank name of the vendor's bank. | varchar(200) | Optional |
| bank_branch | Bank branch of the vendor's bank. | varchar(300) | 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 |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package<br><br>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. | varchar(200) | mandatory |

The response will be in json format as show below:

In case of success,

```
{
    "data": {
        "message": "Vendor updated successfully"
    }
}
```

## 6.3. Delete Vendor API

*URL: https://biz.sparkitpay.com/v1/deletevendor*

This API can be used to delete a pre-existing vendor from the SparkIT Pay 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 | SparkIT Pay 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 |
| vendor_code | This is the vendor code that you wish to delete from the SparkIT Pay system. This value must already exist in the system, failing which SparkIT Pay will return an error. | varchar(30) | mandatory |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately. <span style="color:red">NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package</span><br>hash = toUpper ( sha512 (SALT\| api_key \|vendor_code)) | varchar(40) | mandatory |

This API can be used to delete a pre-existing vendor from the SparkIT Pay 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": {
        "message": "Vendor deleted successfully"
    }
```

```
}
```

## 6.4. Get Vendor API

*URL: https://biz.sparkitpay.com/v1/vendorstatus*

This API can be used to delete a pre-existing vendor from the SparkIT Pay 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 | SparkIT Pay 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 |
| vendor_code | This is the vendor code that you wish to retrieve from the SparkIT Pay system. This value must already exist in the system, failing which SparkIT Pay will return an error. | varchar(30) | mandatory |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package<br>hash = toUpper ( sha512 (SALT\| api_key \|vendor_code)) | varchar(40) | mandatory |

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

The response will be in json format as show below:

In case of success,
```
{
    "data": {
        "vendor_code": "SOA",
        "vendor_name": "Sharath Hegde",
        "vendor_contact_email": "sharathkumar.harishi@gmail.com",
        "vendor_contact_num": "9535640653",
        "vendor_contact_address": "Bangalore",
        "vendor_pan": "AJNPH1120Q",
        "vendor_approved": "n",
        "vendor_split_percentage": null,
        "vendor_split_amount": null,
```

```
        "account_name": "SHarath",
        "account_number": "1234567",
        "ifsc_code": "UTIB0000003",
        "bank_name": "HDFC",
        "bank_branch": "Gandhinagar",
        "bank_approved": "n"
    }
}
```

# 7. SETTLEMENT APIs

## 7.1. getsettlements API

*URL: https://biz.sparkitpay.com/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 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.

### 7.1.1. Parameters to be POSTed in Request

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| bank_reference | The bank reference number of the actual NEFT/IMPS/RTGS transaction performed by SparkIT Pay 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 YYYY-MM-DD format. | varchar(10) | Optional |
| date_to | The end date at which you need to retrieve settlement information. This needs to be passed in YYYY-MM-DD 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 SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately. NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package. | varchar(255) | Mandatory |

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"
 }
}
```

## 7.2. getsettlementdetails API

URL: https://biz.sparkitpay.com/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.

## 7.1.2. Parameters to be POSTed in Request

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| order_id | Order ID passed by the merchant during the original payment transaction request | varchar(30) | optional |
| transaction_id | Transaction ID assigned by SparkIT Pay 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 YYYY-MM-DD format. | varchar(10) | optional |
| date_to | The end date at which you need to retrieve settlement detail information. This needs to be passed in YYYY-MM-DD 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 SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package. | 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":
"D2EFF4776D973DA46563DA0F80139B84AFED77C58496A34DD0D653272A0EE1E5D09F4C94AD439451
2B16341A5A44906B4B10FF5B6AA1F03DE98A164B39881C4E"
}
```

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"
  }
}
```

# 8. CHALLAN PAYMENT API

## 8.1. Request challan payment API

*URL: https://biz.sparkitpay.com/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 | SparkIT Pay 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 |
| 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 SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package. | varchar(40) | mandatory |

## 8.2. Request challan payment API url

*URL:  https://biz.sparkitpay.com/v1/generatechallanurl*

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.

*Request parameters are as following:*

| Parameter Name | Description | Data type | Optional/ Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| 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 SparkIT Pay, using the following mechanism:<br>toUpper ( sha512 (SALT\|amount\| api_key \| email\|mobile\|name\|purpose))<br>Note: the SALT will be provided by SparkIT Pay separately.  Please ensure that the concatenated string (upon which the hash will be computed) is in alphabetical order.  Note: NEVER PASS SALT IN A FORM | varchar(40) | Mandatory |

*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.

*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. |

# 9. Server to Server Call Back (Web hooks)

## 9.1. Server to server response on Payment

To get server to server response, add callback URL in parameter named "Payment Callback URL" in settings tab of the http://biz.sparkitpay.in dashboard. If this is not found contact SparkIT Pay 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 SparkIT Pay. 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 "YYYY-MM-DD 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 | SparkIT Pay 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. |

## 9.2. Server to server response on Settlement

To get server to server response, add callback URL in parameter named "Settlement Callback URL" in settings tab of the http://biz.sparkitpay.in dashboard. If this is not found contact SparkIT Pay to set this up for you.

Whenever there is a successful settlement done by SparkIT Pay 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. |

# 10. Saved Card Token API.

## 10.1. Parameters to be POSTed in the Request

*URL:  https://biz.sparkitpay.com/v2/getsavedcardtokens*

This API is used to get saved card details of the customer for particular merchant.

*Request parameters are as following:*

| Parameter Name | Description | Data type | Optional/ Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| phone | Phone number of the person in whose name the card details are stored | varchar(10) | mandatory |
| email | Email ID of the person in whose name the card details are stored | varchar(100) | mandatory |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package<br>hash = toUpper ( sha512 (SALT| api_key | email | phone) ) | varchar(40) | mandatory |

*Response from this API will be in JSON format:*

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

```
{
  "data": [
    {
      "session_id": "58e620a7a1abc6.97930154",
      "token": "b8dd4590-6b7c-4013-9067-e5a25bc6d812",
      "card_number": "528945******3787"
    },
    {
      "session_id": "58e620a7a1abc6.97930154",
      "token": "7169e21f-beae-4449-b365-41be1dc5630e",
      "card_number": "607432******6800"
    }
  ],
```

```
    "hash":
"C2933D730B00D4F1AAC6FFF1BCDFB1648522AAEDD09E9634919B4501E4CAF873E4448
B063E6D2D5C238243272E37E494A455CD14A8461F04FFB0CC257E6E1C2A"
}
```
**data** – successful response will have "**data**" tag.

**session_id** - this is the session_id which has to be sent while using tokenized payment request api.

**token** – this is the unique identifier for this particular card details.

**card_number** – this is the saved card number in masked form

On failure json response is as following:
```
{
   "error": {
      "code": 404,
      "message": "No saved card details found"
   }
}
```
**error** – erred response will have "**error**" tag.

**code** - this is error category code

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

# 11. Tokenized Payment Request

## 11.1. Parameters to be POSTed in Transaction Request

*URL: https://biz.sparkitpay.com/v1/tokenizedpaymentrequest*

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| 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(255) | mandatory |
| mode | This is the payment mode (TEST or LIVE are valid values) | 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 has to be IND | varchar(50) | mandatory |
| country | Customer country has to be IND | varchar(50) | mandatory |
| zip_code | Customer zip code | varchar(20) | mandatory |
| udf1 | User defined field | varchar(300) | optional |
| udf2 | User defined field 2 | | |
| udf3 | User defined field 3 | | |
| udf4 | User defined field 4 | | |
| udf5 | User defined field 5 | | |
| session_id | Session ID | Varchar(255) | mandatory |
| token | Unique token provided by SparkIT Pay | Varchar(40) | mandatory |
| cvv | Card verification code | Varchar(4) | mandatory |

| | | | |
|---|---|---|---|
| return_url | Return URL success – SparkIT Pay will make a POST request to this URL after successful transaction, with a set of parameters, which you can process as you want to. | | |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package. | | |

Here is a sample HTML page that you can use to test the API before actually doing the integration:

```html
<html>

<head> </head>

<body>
    <form action="https://biz.sparkitpay.com/v1/paymentrequest"
name="payment" method="POST">
        <input type="hidden" value="f141a2b3c-12ab-41s0-bd4e-
de123456d4ff" name="api_key"/>
        <input type="hidden"
value="http://www.yoursite.com/payment_return_url" name="return_url"/>
        <input type="hidden" value="TEST" name="mode"/>
        <input type="hidden" value="00568" name="order_id"/>
        <input type="hidden" value="100" name="amount"/>
        <input type="hidden" value="INR" name="currency"/>
        <input type="hidden" value="Some details about the
transaction" name="description"/>
        <input type="hidden" value="karmendra" name="name"/>
        <input type="hidden" value="your_email_id@example.com"
name="email"/>
        <input type="hidden" value="9900990099" name="phone"/>
        <input type="hidden" value="245. 2nd Main"
name="address_line_1"/>
        <input type="hidden" value="RT Nagar" name="address_line_2"/>
        <input type="hidden" value="Bengaluru" name="city"/>
        <input type="hidden" value="Karnataka" name="state"/>
        <input type="hidden" value="560037" name="zip_code"/>
        <input type="hidden" value="India" name="country"/>
        <input type="hidden" value="" name="udf1"/>
        <input type="hidden" value="" name="udf2"/>
        <input type="hidden" value="" name="udf3"/>
        <input type="hidden" value="" name="udf4"/>
        <input type="hidden" value="" name="udf5"/>
        <input type="hidden" value="12345" name="session_id"/>
        <input type="hidden" value="abcdfef.987563" name="token"/>
        <input type="hidden" value="123" name="cvv"/>
```

```
        <input type="hidden" value="" name="hash"/>
        <button style="color: #fff; background-color: #5cb85c;border-
color: #4cae4c;display: inline-block; padding: 6px 12px; border: 1px
solid transparent;"> SUBMIT </button>
    </form>
</body>

</html>
```

## 11.2. Response Parameters returned by SparkIT Pay

Please refer section 2.3 for the response parameters posted by SparkIT Pay.

## 12. MULTIPLE PAYMENT REQUEST API (beta)

Talk to SparkIT Pay Customer Support before using these APIs.

In case you perform the normal payment integration process as outlined earlier in the document, the customer will need to pay the requested amount at one shot, with one payment instrument. This might be limiting for the customer in certain scenarios such as when the customer does not possess sufficient funds (or credit limit) in a single payment instrument.

In such a scenario, you might want to explore our "multiplepaymentrequest" option wherein the customer gets a chance to pay the given amount using multiple payment instruments, by splitting his transaction into multiple parts. There is no limit to the number of sub-transactions that the customer can generate, subject to the merchant transaction limits being honoured for individual sub-transactions.

### 12.1. Steps for Integration

- You need to submit a POST REQUEST to our server, at the below mentioned URL
  https://biz.sparkitpay.com/v1/multiplepaymentrequest
- 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 SparkIT Pay, we assign a transaction ID to you.
- The customer is re-directed to SparkIT Pay's "multi-payment" page. The customer can choose a series of amounts to pay, in a sequence. Each sub-transaction directs the customer to the default payment page, whereby the customer can perform a transaction using any payment instrument of his choice, among the available/enabled ones.
- 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.

### 12.2. Parameters to be POSTed in Multiple Payment Request

URL: *https://biz.sparkitpay.com/v1/multiplepaymentrequest*

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay would assign a unique 40-digit merchant key to you. This key is exclusive to | varchar(40) | mandatory |

| | 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. | | |
|---|---|---|---|
| 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(255) | mandatory |
| mode | This is the payment mode (TEST or LIVE are valid values) | 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 has to be IND | varchar(50) | mandatory |
| country | Customer country has to be IND | varchar(50) | mandatory |
| zip_code | Customer zip code | varchar(20) | mandatory |
| udf1 | User defined field | varchar(300) | optional |
| udf2 | User defined field 2 | | |
| udf3 | User defined field 3 | | |
| udf4 | User defined field 4 | | |
| udf5 | User defined field 5 | | |
| split_enforce_strict | This is an optional parameter that can be passed. If it is set to y, the settlement api will wait for a successful split api call for that transaction before creating the settlements. If this parameter is not being passed, don't include it in hash check. Default value is 'n' | varchar(1) | optional |
| return_url | Return URL success – SparkIT Pay will make a POST request to this URL after successful transaction, with a set of parameters, which you can process as you want to. | | |
| hash | You need to compute a hash of all your parameters and pass that hash to SparkIT Pay, for details about calculating hash refer Appendix 2. | | |

| | Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package. | | |
|---|---|---|---|

Here is a sample HTML page that you can use to test the API before actually doing the integration:

```html
<html>
<head> </head>
<body>
    <form action="https://biz.sparkitpay.com/v1/
multiplepaymentrequest" name="payment" method="POST">
        <input type="hidden" value="f141a2b3c-12ab-41s0-bd4e-
de123456d4ff" name="api_key"/>
        <input type="hidden"
value="http://www.yoursite.com/payment_return_url" name="return_url"/>
        <input type="hidden" value="TEST" name="mode"/>
        <input type="hidden" value="00568" name="order_id"/>
        <input type="hidden" value="100" name="amount"/>
        <input type="hidden" value="INR" name="currency"/>
        <input type="hidden" value="Some details about the
transaction" name="description"/>
        <input type="hidden" value="karmendra" name="name"/>
        <input type="hidden" value="your_email_id@example.com"
name="email"/>
        <input type="hidden" value="9900990099" name="phone"/>
        <input type="hidden" value="245. 2nd Main"
name="address_line_1"/>
        <input type="hidden" value="RT Nagar" name="address_line_2"/>
        <input type="hidden" value="Bengaluru" name="city"/>
        <input type="hidden" value="Karnataka" name="state"/>
        <input type="hidden" value="560037" name="zip_code"/>
        <input type="hidden" value="India" name="country"/>
        <input type="hidden" value="" name="udf1"/>
        <input type="hidden" value="" name="udf2"/>
        <input type="hidden" value="" name="udf3"/>
        <input type="hidden" value="" name="udf4"/>
        <input type="hidden" value="" name="udf5"/>
        <input type="hidden" value="" name="hash"/>
        <button style="color: #fff; background-color: #5cb85c;border-
color: #4cae4c;display: inline-block; padding: 6px 12px; border: 1px
solid transparent;"> SUBMIT </button>
    </form>
</body>

</html>
```

## 12.3. Response Parameters returned by SparkIT Pay

### 12.3.1. 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 SparkIT Pay. 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 "YYYY-MM-DD 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_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 |
| balance_amount | Balance amount not paid |
| hash | SparkIT Pay 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. |
| error_desc | Failure reason (if transaction is failed) |

### 12.3.2. Server to Server Multiple Payment Response

After every transaction success or failed, SparkIT Pay can post response as mentioned above. To get this activated contact SparkIT Pay support. This is useful if you need information about incomplete transactions.

# 13. MULTIPLE PAYMENT STATUS API (beta)

Talk to SparkIT Pay Customer Support before using these APIs.

## 13.1. Payment Status by merchant's order id

SparkIT Pay 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.

In multiple payment status we return ids of all the part transactions. You need to use Payment Status API in section 3 and pass these ids to get information about each part payment

*URL: https://biz.sparkitpay.com/v1/multiplepaymentstatus*

### 13.1.1. Parameters to be POSTed

| Parameter Name | Description |
|---|---|
| api_key | Unique merchant identifier key provided by SparkIT Pay. |
| hash | This is a mandatory parameter, and is required for cross-verification. You need to calculate the hash as follows:<br><br>sha512(SALT\|api_key\|order_id)<br><br>Note:  NEVER PASS SALT IN A FORM |
| order_id | This is the original order_id that you passed while making the transaction request. |

### 13.1.2 Parameters in the Response

| Parameter | Description |
|---|---|
| transaction_id | Transaction id is generated for each Multiple payment Request |
| pg_transaction_id | Transaction id is generated for each part payment of a single Multiple Payment request. |
| pg_order_id | Use this order_id to get status of this part payment using paymentstatus api in section 3. |
| order_id | The same order_id that was originally posted by the merchant in the request. |
| total_amount | The same original amount that was sent by the merchant in the transaction request. |

| | |
|---|---|
| split_amount | This is the amount that user tried to pay in part |
| hash | SparkIT Pay calculates the hash using the same algorithm which was outlined earlier. Hence, the merchant needs to check whether this returned hash matches the hash that was originally passed. |
| error_desc | Failure reason (if transaction is failed) |

```json
{
    "data": [
        {
            "order_id": "1001",
            "total_amount": "5.00",
            "transaction_id": "cdb799eab662031",
            "pg_transaction_id": "HTRQ8983",
            "pg_order_id": 5003,
            "split_amount": "2.50"
        },
        {
            "order_id": "1001",
            "total_amount": "5.00",
            "transaction_id": "cdb799eab662031",
            "pg_transaction_id": "ETRQ8984",
            "pg_order_id": 5004,
            "split_amount": "2.50"
        },
        {
            "order_id": "1001",
            "total_amount": "5.00",
            "transaction_id": "cdb799eab662031",
            "pg_transaction_id": "HTRQ8985",
            "pg_order_id": 5005,
            "split_amount": "2.50"
        }
    ]
}
```

## 14. 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 SparkIT Pay 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 SparkIT Pay payment page, this can be achieved using the seamless payment request API. SparkIT Pay 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 SparkIT Pay relationship manager.

### 14.1. Steps for Integration

- You need to submit a POST REQUEST to our server, at the below mentioned URL
  https://biz.sparkitpay.com/v2/paymentrequestseamless
- 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 SparkIT Pay, 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.

### 14.2. Parameters to be POSTed in Seamless Payment Request

URL: https://biz.sparkitpay.com/v2/paymentrequestseamless

| Parameter Name | Description | Data type | Optional / Mandatory |
|---|---|---|---|
| api_key | SparkIT Pay 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 |
| order_id | This is your (merchant) reference number. It must be unique for every transaction. | Varchar(40) | 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 has to 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 |
| Bankcode | Bankcode – this identifies the  payment mode and channel. | Varchar(20) | Mandatory |
| Cardnum | Card number (11 to 19 digits) | Varchar(19) | Optional |
| expdt | Expiry date in mm/yyyy format | Varchar(7) | Optional |
| Cardname | Card holder name | Varchar(30) | Optional |
| cvv | CVV/CVC | Varchar(4) | Optional |
| return_url | Return URL success – SparkIT Pay 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 – SparkIT Pay 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 – SparkIT Pay 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 SparkIT Pay, for details about calculating hash refer Appendix 2.<br><br>Note: the SALT will be provided by SparkIT Pay separately.  NEVER PASS SALT IN A FORM, DO NOT STORE SALT IN ANDROID APP APK or IPHONE APP package. | Varchar(200) | Mandatory |

## 14.3. Response Parameters returned by SparkIT Pay

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 SparkIT Pay. 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 "YYYY-MM-DD 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 | SparkIT Pay 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. |

# 15. APPENDIX 1 - References

- To download sample php integration kit go to
  https://bitbucket.org/OmniwareIntegrationTeam/tnp_integration_kit_php/downloads and click
  on Download Repository.

- Instructions to use the kit is in readme.md file **OR** you can also go to
  https://bitbucket.org/OmniwareIntegrationTeam/tnp_integration_kit_php/overview

# 16. Appendix 2 - Hash calculation guide

## 16.1. How to Calculate Hash on API request

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

### Hashing generation algorithm

Following are the steps to calculate hash.

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

### Example PHP code to generate hash

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

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

    return $secure_hash;
}
```

## 16.2. How to check the response Hash

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

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

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

## Hash checking algorithm

## Example PHP code to check hash

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

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

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

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

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

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

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

    return $hash;
}
```

# 17. Appendix 3 - List of bank codes

| Bankcode | Payment Mode | Payment Channel | Description |
|---|---|---|---|
| VISC | Credit Card | Visa | Visa Credit Card |
| VISD | Debit Card | Visa | Visa Debit Card |
| MASC | Credit Card | Master | Master Credit Card |
| MASD | Debit Card | Master | Master Debit Card |
| MAED | Debit Card | Maestro | Maestro Debit Card (non-SBI) |
| MSED | Debit Card | Maestro (SBI) | Maestro Debit Card (SBI) |
| DINC | Credit Card | Diners | Diners Credit Card |
| AMXC | Credit Card | Amex | American Express Credit Card |
| RUPD | Debit Card | Rupay | Rupay Debit Card |
| BOIA | ATM Card | Bank of India | Bank of India ATM Card |
| PNBA | ATM Card | Punjab National Bank | Punjab National Bank ATM Card |
| BOBA | ATM Card | Bank of Baroda | Bank of Baroda  ATM Card |
| BMBA | ATM Card | Bharatiya Mahila Bank | Bharatiya Mahila Bank  ATM Card |
| BOMA | ATM Card | Bank of Maharashtra | Bank of Maharashtra  ATM Card |
| CANA | ATM Card | Canara Bank | Canara Bank  ATM Card |
| INDA | ATM Card | IndusInd Bank | IndusInd Bank  ATM Card |
| IDFA | ATM Card | IDFC | IDFC   ATM Card |
| IOBA | ATM Card | Indian Overseas Bank | Indian Overseas Bank  ATM Card |
| KGBA | ATM Card | Kerala Gramin Bank | Kerala Gramin Bank  ATM Card |
| LVBA | ATM Card | Laxmi Vilas Bank | Laxmi Vilas Bank  ATM Card |
| PKGA | ATM Card | Pragathi Krishna Gramin Bank | Pragathi Krishna Gramin BankATM Card |
| RSCA | ATM Card | Rajasthan State Coop Bank | Rajasthan State Coop Bank  ATM Card |
| SMCA | ATM Card | Shivalik Mercantile Co-Op Bank Limited | Shivalik Mercantile Co-Op Bank Limited ATM Card |
| UBIA | ATM Card | Union Bank of India | Union Bank of India  ATM Card |
| PTMW | Wallet | Paytm | Paytm wallet |
| MBKW | Wallet | Mobikwik | Mobikwik Wallet |
| OXIW | Wallet | Oxigen | Oxigen Wallet |
| MRPW | Wallet | mRupee | mRupee Wallet |
| JIOW | Wallet | Reliance JioMoney | Reliance Jio Money wallet |
| TMWW | Wallet | The Mobile Wallet | The Mobile Wallet |
| PYCW | Wallet | Paycash | Paycash wallet |
| OLAW | Wallet | Ola Money | Ola Money Wallet |
| ITZH | Cash Card | ITZ Cash Card | ITZ Cash card |
| ICSH | Cash Card | Icash Card | Icash card |
| ALLN | Netbanking | Allahabad Bank | Allahabad Bank NetBanking |
| ADBN | Netbanking | Andhra Bank | Andhra Bank |
| ADBM | Netbanking | Andhra Bank - Corporate | Andhra Bank - Corporate |
| AXIN | Netbanking | AXIS Bank | AXIS Bank NetBanking |
| BDNN | Netbanking | Bandhan bank | Bandhan Bank |

| BBKN | Netbanking | Bank of Bahrain and Kuwait | Bank of Bahrain and Kuwait |
|------|-----------|---------------------------|---------------------------|
| BBRM | Netbanking | Bank of Baroda - Corporate | Bank of Baroda Corporate Banking |
| BBRN | Netbanking | Bank of Baroda - Retail | Bank of Baroda Retail Banking |
| BOIN | Netbanking | Bank of India | Bank of India |
| BOMN | Netbanking | Bank of Maharashtra | Bank of Maharashtra |
| BCBN | Netbanking | Bassein Catholic Bank | Bassein Catholic Bank |
| BMBN | Netbanking | Bharatiya Mahila Bank | Bharatiya Mahila Bank |
| CANN | Netbanking | Canara Bank | Canara Bank |
| CSBN | Netbanking | Catholic Syrian Bank | Catholic Syrian Bank |
| CBIN | Netbanking | Central Bank Of India | Central Bank Of India |
| CITN | Netbanking | Citi Bank NetBanking | Citi Bank NetBanking |
| CUBN | Netbanking | City Union Bank | City Union Bank |
| CRPN | Netbanking | Corporation Bank | Corporation Bank |
| COSN | Netbanking | Cosmos Bank | Cosmos Bank |
| DBSN | Netbanking | DBS Bank | DBS Bank |
| DCBM | Netbanking | DCB Bank - Corporate | DCB Bank - Corporate Netbanking |
| DENN | Netbanking | Dena Bank | Dena Bank |
| DSHN | Netbanking | Deutsche Bank | Deutsche Bank |
| DCBN | Netbanking | Development Credit Bank | Development Credit Bank |
| DHNM | Netbanking | Dhanlakshmi Bank - Corporate Net Banking | Dhanalakshmi Bank - corporate |
| DHNN | Netbanking | Dhanlakshmi Bank | Dhanalakshmi Bank |
| FEDN | Netbanking | Federal Bank | Federal Bank |
| HDFN | Netbanking | HDFC Bank | HDFC Bank |
| ICIN | Netbanking | ICICI Bank | ICICI Netbanking |
| IDFN | Netbanking | IDFC Bank | IDFC Bank |
| ININ | Netbanking | Indian Bank | Indian Bank |
| IOBN | Netbanking | Indian Overseas Bank | Indian Overseas Bank |
| INDN | Netbanking | IndusInd Bank | IndusInd Bank |
| IDBN | Netbanking | IDBI | Industrial Development Bank of India |
| INGN | Netbanking | ING Vysya Bank | ING Vysya Bank |
| JAKN | Netbanking | Jammu and Kashmir Bank | Jammu and Kashmir Bank |
| JSBN | Netbanking | Janata Sahakari Bank | Janata Sahakari Bank |
| KJBN | Netbanking | Kalyan Janata Sahakari Bank | Kalyan Janata Sahakari Bank |
| KRKN | Netbanking | Karnataka Bank | Karnataka Bank |
| KRVN | Netbanking | Karur Vysya - Retail | Karur Vysya |
| KRVM | Netbanking | Karur Vysya - Corporate | Karur Vysya - Corporate Netbanking |
| KKBN | Netbanking | Kotak Mahindra Bank | Kotak Mahindra Bank |
| LVBN | Netbanking | Laxmi Vilas Bank - Corporate | Laxmi Vilas Bank - Corporate |
| LVBM | Netbanking | Laxmi Vilas Bank - Retail | Laxmi Vilas Bank - Retail |
| MSBN | Netbanking | The Mehsana Urban Co Op Bank Ltd | The Mehsana Urban Co Op Bank Ltd. |
| NKBN | Netbanking | NKGSB Bank | NKGSB Bank |
| OBCN | Netbanking | Oriental Bank of Commerce | Oriental Bank of Commerce |

| PMCN | Netbanking | Punjab & Maharashtra Coop Bank | Punjab & Maharashtra Coop Bank |
|------|------------|-------------------------------|-------------------------------|
| PSBN | Netbanking | Punjab & Sind Bank | Punjab & Sind Bank |
| PNBN | Netbanking | Punjab National Bank - Retail | Punjab National Bank - Retail Banking |
| PNBM | Netbanking | Punjab National Bank - Corporate | Punjab National Bank-Corporate |
| RTNN | Netbanking | RBL Bank Limited | RBL Bank Limited |
| RTNM | Netbanking | RBL Bank Limited- Corporate Net Banking | RBL Bank Limited- Corporate Net Banking |
| SRSN | Netbanking | Saraswat Bank | Saraswat Bank |
| SVCN | Netbanking | Shamrao Vitthal Co-operative Bank | Shamrao Vitthal Co-operative Bank |
| SVCM | Netbanking | Shamrao Vitthal Co-operative Bank - Corporate | Shamrao Vitthal Co-operative Bank - Corporate |
| SOIN | Netbanking | South Indian Bank | South Indian Bank |
| SCBN | Netbanking | Standard Chartered Bank | Standard Chartered Bank |
| SBJN | Netbanking | State Bank of Bikaner and Jaipur | State Bank of Bikaner and Jaipur |
| SBHN | Netbanking | State Bank of Hyderabad | State Bank of Hyderabad |
| SBIN | Netbanking | State Bank of India | State Bank of India |
| SBMN | Netbanking | State Bank of Mysore | State Bank of Mysore |
| SBPN | Netbanking | State Bank of Patiala | State Bank of Patiala |
| SBTN | Netbanking | State Bank of Travancore | State Bank of Travancore |
| SYDN | Netbanking | Syndicate Bank | Syndicate Bank |
| TMBN | Netbanking | Tamilnad Mercantile Bank Ltd. | Tamilnad Mercantile Bank Ltd. |
| TSCN | Netbanking | Tamilnadu State Coop Bank | Tamilnadu State Coop Bank |
| TJSN | Netbanking | TJSB Bank | TJSB Bank |
| UCON | Netbanking | UCO Bank | UCO Bank |
| UBIM | Netbanking | Union Bank of India - Corporate | Union Bank - Corporate Netbanking |
| UBIN | Netbanking | Union Bank of India - Retail | Union Bank of India |
| UNIN | Netbanking | United Bank Of India | United Bank Of India |
| VIJN | Netbanking | Vijaya Bank | Vijaya Bank |
| YESN | Netbanking | Yes Bank | Yes Bank |
| JNCW | Wallet | Jana Cash | Jana Cash |
| ATLW | Wallet | Airtel Money | Airtel Money |
| PNBW | Wallet | PNB Wallet | PNB Wallet |
| FRCW | Wallet | FreeCharge | FreeCharge Wallet |
| EZCW | Wallet | ezeClick | ezeClick Wallet |
| ESFN | Netbanking | Equitas Bank | Equitas Small Finance Bank Limited |
| VDFW | Wallet | mPesa Wallet | mPesa - Vodafone Wallet |

# 18. Appendix 4 - List of error codes

| Error Code | Numeric code | Description |
|---|---|---|
| SUCCESS | 0 | Transaction successful |
| FAILED | 1000 | Transaction failed |
| INVALID-API-KEY | 1001 | The api key field is incorrect |
| INVALID-LIVE-MODE-ACCESS | 1002 | The live mode access is not allowed |
| INVALID-ORDER-ID-FIELD | 1003 | The order id field should be unique |
| ORDER-ID-FIELD-NOT-FOUND | 1004 | The order id field is not found |
| INVALID-AUTHENTICATION | 1005 | Invalid authentication at bank |
| WAITING-BANK-RESPONSE | 1006 | Waiting for the response from bank |
| INVALID-INPUT-REQUEST | 1007 | Invalid input in the request message |
| TRANSACTION-TAMPERED | 1008 | Your transaction is successful, but we suspect malicious activity in your account. Please contact the support team |
| DECLINED-BY-BANK | 1009 | Bank Declined Transaction |
| INVALID-AMOUNT | 1010 | Amount cannot be less than 1 |
| AUTHORIZATION-REFUSED | 1011 | Authorization refused |
| INVALID-CARD | 1012 | Invalid Card/Member Name data |
| INVALID-EXPIRY-DATE | 1013 | Invalid expiry date |
| DENIED-BY-RISK | 1014 | Transaction denied by risk |
| INSUFFICIENT-FUND | 1015 | Insufficient Fund |
| INVALID-AMOUNT-LIMIT | 1016 | Total Amount limit set for the terminal for transactions has been crossed |
| INVALID-TRANSACTION-LIMIT | 1017 | Total transaction limit set for the terminal has been crossed |
| INVALID-DEBIT-AMOUNT-LIMIT | 1018 | Maximum debit amount limit set for the terminal for a day has been crossed |
| INVALID-CREDIT-AMOUNT-LIMIT | 1019 | Maximum credit amount limit set for the terminal for a day has been crossed |
| MAXIMUM-DEBIT-AMOUNT-CROSS | 1020 | Maximum debit amount set for per card for rolling 24 hrs has been crossed |
| MAXIMUM-CREDIT-AMOUNT-CROSS | 1021 | Maximum credit amount set for per card for rolling 24 hrs has been crossed |
| MAXIMUM-TRANSACTION-CROSS | 1022 | Maximum transaction set for per card for rolling 24 hrs has been crossed |
| HASH-MISMATCH | 1023 | Hash Mismatch |
| INVALID-PARAMS | 1024 | Invalid parameters |
| UNKNOWN-ERROR | 9999 | Unknown error occurred |
| INVALID-BANK-CODE | 1025 | Invalid bank code |
| INVALID-MERCHANT | 1026 | Merchant is not active |
| INVALID-TRANSACTION | 1027 | Transaction is invalid |
| TRANSACTION-NOT-FOUND | 1028 | Transaction not found |

| TRANSACTION-TERMINATED | 1029 | Transaction terminated |
|---|---|---|
| TRANSACTION-INCOMPLETE | 1030 | Transaction incomplete |
| AUTO-REFUNDED | 1031 | Transaction auto refunded |
| REFUNDED | 1032 | Transaction refunded |
| SINGLE-TRANSACTION-LOWER-LIMIT-CROSS | 1033 | The amount provided is less than transaction lower limit |
| SINGLE-TRANSACTION-UPPER-LIMIT-CROSS | 1034 | The amount provided is more than transaction upper limit |
| TRANSACTION-DAILY-LIMIT-CROSS | 1035 | The daily transaction limit is exceeded for the merchant |
| TRANSACTION-MONTHLY-LIMIT-CROSS | 1036 | The monthly transaction limit is exceeded for the merchant |
| DAILY-TRANSACTION-NUMBER-CROSS | 1037 | The daily transaction number is exceeded for the merchant |
| MONTHLY-TRANSACTION-NUMBER-CROSS | 1038 | The monthly transaction number is exceeded for the merchant |
| INVALID-REFUND-AMOUNT | 1039 | The refund amount is greater than transaction amount |
| INVALID-CVV | 1040 | Invalid Card Verification Code |
| AUTO-REFUNDED-TNP | 1041 | Transaction is auto refunded by TnP |
| FAILED-NO-BANK-RESPONSE | 1042 | Transaction failed as there was no response from bank |
| TRANSACTION-CANCELLED | 1043 | Transaction cancelled |