Merchant Integration API Specification document

Indian Bank

Table of Contents

Document Information

Document Name:	API Specification document Indian Bank Merchant
Document Identification:	API_Specification - IB_Merchant_API Integration_v1.6.docx

Document Revision History

Sr. No.	Version	Release Date	Author	Reviewer	Reason for change
1	1.0	30-Jan-2022	Vinit	Parthasarathi M	Initial version
2	1.1	15-Feb-2022	Naveen	Parthasarathi M	Collect API – added expdate, Payeraccount Refund API – added txnRefID Common checksum, encryption and decryption code snippet added.
3	1.2	15-Mar-2022	Naveen	Parthasarathi.M	Added Pay transaction API
4	1.3	23-Mar-2022	Naveen	Parthasarathi.M	Added validate address API
5	1.4	25-Oct-2022	Naveen	Parthasarathi.M	Added extra parameter in onboarding response and Query txn added TxnRefID. The changes are highlighted.
6	1.5	02-Nov-2022	Naveen	Parthasarathi.M	Added VPA deactivation API, refund API field changed from Mandatory to Conditional QR generation API added Checksum sample logic given for easy understanding
7	1.6	30-Mar-2023	Aaditya Kadam	Parthasarathi.M	Added checksum logic for all the APIs

1 Integration Scope

The purpose of this document is to specify API and details of all API's related to merchant integration which we will use at the time of transaction initiated for merchant

2 API Listing for Merchant Integration

Sr No	API NAME	Request Parameter	Response	Description
1	On-boarding API	entityId mobileNo paymentAddress merchantAccountNo accountType IFSC merchantLegalName channelId aggregatorCode merchantId MCC terminalId Checksum	Success / Failed	This API is use to create / onboard merchant at UPI switch end.
2	Onboard Check Status API	entityId mobileNo paymentAddress merchantAccountNo accountType IFSC merchantLegalName channelId aggregatorCode merchantId MCC terminalId Checksum	Success / Failed	This API is use to do check whether merchant is successfully onboarding in Indian bank system.
3	Callback API		NA	This is the final callback response that is sent to payment gateway [PG]. merchant to provide API details.
4	Query Transaction	entityId txnId checkSum	Success / Failed	This would give retrieve the transaction details of any particular transaction
5	Refund API	entityId txnAmount	Success/Failed	This would initiate the refund

		txnId checkSum		
6	Collect API	entityId txnAmount txnId payer detail expdate checkSum	Success/Failed	Collect request from merchant to customer
7	Pay API	entityId txnAmount txnId payer detail payee detail checkSum	Success/Failed	Pay API is used to send money from a payer's bank account to a payee's bank account as a preapproved transaction. UPI switch will be considered as preapproved transaction from merchants
8	Validate address	entityId txnAmount txnId payer detail payee detail checkSum	Success/Failed	This API will be used by the PSPs when their customer wants to add a beneficiary within PSP application (for sending & collecting money).
9	VPA deactivation	entityId paymentAddress mobileNo checkSum	Success/Failed	This API is used to deactivate the merchant created VPA
10	QR generation API	entityId paymentAddress mobileNo checkSum	Success/Failed	This API is used to generate QR String

3 API Description

This section describes the APIs that are used for UPI integration. We have 2 APIs for merchant Integration as Merchant Onboarding API and Check onboard status API.

3.1 Onboarding API

Merchant creation will be done in this Merchant Onboarding Module. Merchant will call merchant onboard API of Infrasoft and pass on required Merchant details to Infrasoft. Post successful validation at Infrasoft, VPA will be created in Indian Bank UPI system.

Sample URL : **URL PROVIDED BY INFRASOFT at the time of integration**
Method : **Post**
Content-Type : **application/JSON**

3.1.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Min: 10 Max: 12	Entity Id of the bank. It should be always 'INB'
2	mobileNo	String	M	Min: 10 Max: 12	Mobile number of the merchant
3	paymentAddress	String	M	Max: 99	Payment address of the merchant which will link to QR
4	merchantAccountNo	String	M	Min: 1 Max: 20	Account number of the merchant
5	accountType	String	M	Min: 1 Max: 20	Account type of the merchant like "SAVINGS", "CURRENT" etc.
6	IFSC	String	M	Min: 1 Max: 11	Ifsc code associated with account that will be used for onboarding.
7	merchantLegalName	String	M	Min: 1 Max: 100	Legal name of the merchant.
8	channelId	String	M	Min: 1 Max: 100	Channel of a transaction like UPI, USSD etc. For MERCHANT it will be UPI.
9	aggregatorCode	String	M	Min: 1 Max: 100	Code of Aggregator. Incase of MERCHANT, they have to pass "MERCHANT".
10	merchantId	String	M	Min: 1 Max: 15	Merchant ID of merchant given by aggregator to individual merchant.

11	MCC	String	M	Max: 4	Merchant category code allocated to merchant.
12	terminalId	String	M	Min: 1 Max: 100	Terminal ID generated by aggregator while on boarding merchant.
13	Checksum	String	M		Calculated as given below.

The response code will be sent as mentioned below:

Status	Response code
SUCCESS	00
FAILURE	01
SUSPECTED	91

3.1.2 Sample Request

Encrypted Request

```
{
        "data":"< encrypted json data…….>"
}
```

After Decryption Sample Request Posted

```
{
    "entityId":" INB",
    "mobileNo":" 918888899999",
    "paymentAddress":"abc@indianbnk ",
    "merchantAccountNo":"123456789",
    "accountType":"Saving",
    "IFSC":"IDIB0000001",
    "merchantLegalName ":"ABC vendor Ltd.",
    "channelId":"UPI",
    "aggregatorCode":"MERCHANT",
    "merchantId":"9999",
    "MCC":"9890",
    "terminalId ":" INB719519273055705102",
```

"Checksum":
"6E00E309B6EBF2BE49A64DE331C4F3030263F8F86AF97CEF8873C4996C9553D55FD51A9A917E1EBB7092E8C063233AC61ECE3D3A775A0B64516A7281B9F1F6F99CAACE3DC14D54ACC127AFF16DF2639D378F4862C26AD45167113EB64E0C8AE3AC04F310C560FFBA448D67079C9CDBAD0B502AD69156E26F75AEDDDAFDFF7A7E80CEF759F889F16DE33A1F08E54DA2B21A2C543790B1BDC320120A1BB777AD76DF7B89631B6B3D9AC90C072B9E5A9D59D465B9409A8F39BC08E78BE5916ED42DDD3C60676EE7A196ACBA02C3537829A74B1B0B769D0E67B1CD222F626100BAC749CD08C70C72890382C98073C3B41D813D054BBDD3F8F1C2F4F6FB75AE8603D6"
}

Note: Request message (in JSON format) will be encrypted with MERCHANT key as per the logic mentioned in section "***Encryption and Request Checksum Logic"***

3.1.3 Response Parameter:

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	responseCode	String	M	Min:1 Max: 10	Reponse code for the request processed at Infrasoft end
2	ResponseStatus	String	M	Min:1 Max: 10	Status of response. SUCCESS / FAILURE / DEEMED /SUSPECTED
4	txnID	String	M	Min:1 Max: 35	Transaction ID of a request generated by Infrasoft.
5	merchantId	String	M	Min:1 Max: 15	Merchant ID of merchant given by aggregator to individual merchant.
6	**merchantCode**	String	M	Min:1 Max: 15	MerchantCode is used to identify the Unique submerchants .
7	terminalId	String	M	Min:1 Max: 35	Unique terminalId generated by agrgegator while initiating request for merchant on-boarding.

3.1.4 Sample Response:

Success sample response:

{
"responseCode":"00",
"ResponseStatus":"Success",
"data":{
"txnID":" INBde92b19862e44226b2ff318c8462c6a9",

"merchantId":"9999",
“merchantCode”:12345”,
“terminalId”:” 719519273055705102”
}
}

Failure sample response:
{
“responseCode”:”01”,
“ResponseStatus”:”Failure”,
“data”:{
“txnID”:” INBde92b19862e44226b2ff318c8462c6a9”,
"merchantId":"9999",
“terminalId”:” 719519273055705102”
}
}

3.2 Onboard Check Status API

Merchant creation will be done using Merchant Onboarding API. If response is not received for Merchant onboard API then MERCHANT will initiate check Merchant onboard check status API to know the status of on-boarding. Aggregator will pass required parameters mentioned below and Infrasoft will provide response for the same.

Sample URL **: URL PROVIDED BY INFRASOFT at the time of integration**
Method **: Post**
Content-Type **: application/JSON**

3.2.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Min: 10 Max: 12	Entity Id of the bank it should be always ‘INB
2	mobileNo	String	M	Min: 10 Max: 12	Mobile number of a merchant
3	paymentAddress	String	M	Min: 1 Max: 99	Payment address of a merchant which will link to QR
4	merchantAccountNo	String	M	Min: 1 Max: 20	Account number of a merchant

5	accountType	String	M	Min: 1 Max: 20	Account type of a merchant like “SAVINGS”, “CURRENT” etc.
6	IFSC	String	M	Min: 1 Max: 11	Ifsc code of account.
7	merchantLegalName	String	M	Min: 1 Max: 100	Legal name of a merchant.
8	channelId	String	M	Min: 1 Max: 100	Channel of a transaction like UPI, USSD etc. For MERCHANT it will be UPI.
9	aggregatorCode	String	M	Min: 1 Max: 20	Code of Aggregator. Incase of MERCHANT, they have to pass “MERCHANT”.
10	merchantId	String	M	Min: 1 Max: 15	Merchant ID of merchant given by aggregator to individual merchant.
11	MCC	String	M	Max 4	Merchant category code allocated to merchant.
12	terminalId	String	M	Min: 1 Max: 100	Terminal ID generated by agreegator while on boarding merchant
13	Checksum	String	M		Calculated as given below

The response code will be sent as mentioned below:

Status	Response code
SUCCESS	00
FAILURE	01
SUSPECTED	91

3.2.2 Sample Request

Encrypted Request:

```
{
        "data":"< encrypted json data…….>"
}
```

After Decryption Sample Request Posted

```
{
        "entityId":"INB",
        "mobileNo":" 918888899999",
        "paymentAddress":"abc@indianbk ",
        "merchantAccountNo":"123456789",
        "accountType":"Saving",
        "IFSC":"IDIB0000001",
        "merchantLegalName ":"ABC vendor Ltd.",
        "channelId":"UPI",
        "aggregatorCode":"MERCHANT",
        "merchantId":"9999",
        "MCC":"9890",
        "terminalId ":" 719519273055705102",
        "Checksum":
        "6E00E309B6EBF2BE49A64DE331C4F3030263F8F86AF97CEF8873C4996C9553D55FD51A9A
        917E1EBB7092E8C063233AC61ECE3D3A775A0B64516A7281B9F1F6F99CAACE3DC14D54AC
        C127AFF16DF2639D378F4862C26AD45167113EB64E0C8AE3AC04F310C560FFBA448D67079
        C9CDBAD0B502AD69156E26F75AEDDDAFDFF7A7E80CEF759F889F16DE33A1F08E54DA2B2
        1A2C543790B1BDC320120A1BB777AD76DF7B89631B6B3D9AC90C072B9E5A9D59D465B940
        9A8F39BC08E78BE5916ED42DDD3C60676EE7A196ACBA02C3537829A74B1B0B769D0E67B1
        CD222F626100BAC749CD08C70C72890382C98073C3B41D813D054BBDD3F8F1C2F4F6FB75
        AE8603D6"
}
```

Note: Request message (in JSON format) will be encrypted with MERCHANT key as per the logic mentioned in section "***Encryption and Request Checksum Logic"***

3.2.3 Response Parameter:

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	responseCode	String	M	Min:1 Max: 10	Reponse code for the request processed at infrasoft end
2	ResponseStatus	String	M	Min:1 Max: 10	Status of response. SUCCESS / FAILURE.
3	failureResaon	String	M (in case of failure status	Min:1 Max: 35	Reason of failure for merchant onboarding.

4	txnID	String	M	Min:1 Max: 35	Transaction ID of a request generated by Infrasoft.
5	merchantId	String	M	Min:1 Max: 15	Merchant ID of merchant given by aggregator to individual merchant.
6	terminalId	String	M	Min:1 Max: 35	Unique terminalId generated by aggregator while initiating request for merchant on boarding.

3.2.4 Sample Response:

Success sample response:

```
{
“responseCode”:”00”,
“ResponseStatus”:”Success”,
“data”:{
        “txnID”:” INB11111111111111222222222222”,
        "merchantId":"9999",
        “terminalId”:” 719519273055705102”
        }
}
```

Failure sample response:

```
{
“responseCode”:”01”,
        “ResponseStatus”:”Failure”,
        “data”:{
        “failureResaon”:”VPA already exist”,
         “txnID”:” INBde92b19862e44226b2ff318c8462c6a9”,
        "merchantId":"9999",
        “terminalId”:” 719519273055705102”
        }
}
```

3.3 Call Back API

Call back URL API helps merchant to receive the transaction status from the bank UPI server.

API URL	[https://ipaddress:Port/upi/merchant/approval] https://txuat.merchantpay.com:443/QR/InfraUPI/Callback
HTTP Method	POST

Request Type	application/json
Response Type	application/json

Example for the URL for UAT environment

https://txuat.merchantpay.com:443/QR/InfraUPI/callback.php?
msg={"resp":"51361DFA5EE8CD5EA3AD06C43C5392BA0A22
3DBE09067A2F3E6521BB79768A5AF7A1D99661F78E0AF8966254484F7963D55065ABD050BD0C241BB83B
C4D"}

3.2.1 Request Parameter

Request			
Parameter Name	M - Mandatory O - Optional C - conditional	Type(Size)	Description
PSPRefNO	O	30	Original Transaction Reference Number sent by Merchant
TransID	M	35	Transaction ID
CustRefNo	M	12	Reference No print in receipt as RRN
Amount	M	16,2 (Numeric)	Transaction Amount
TxnAuthDate	M	Date	DATE IN THE FORMAT YYYYMMDDHH24MISS
responseCode	M	2	Response Code received from NPCI 91 for suspected.
approvalNumber	M	6	Transaction approval number (Core bank ref number) This wil be hardcoded as “000000”
Status	M	30	Status of the transaction
AddInfo	O		FUTURE USE
Payer_VPA	M	99	Payer Virtual Private Address
Payee_VPA	M	99	Payee Virtual Private Address
OrderNo	M	30	Order Number will consist the following TID[8] invoice[6] stan[6] Ref ID received from NPCI will be passed. In case of static QR code, REF ID will be common.
CurrentStatusDesc	O	100	Transaction status description
TransactionNote	C	12	Future Use Remarks received from NPCI will be passed

TransactionType	O	10,2	Future Use
RefURL	O	50	Future Use
checksum	M		Calculated as given below

The response code will be sent as mentioned below;

Status	Response code
SUCCESS	00
FAILURE	Response code received from NPCI which may be 3 characters in some cases
DEEMED	Response code received from NPCI which may be 3 characters in some cases
SUSPECTED	91

3.2.2 Sample Request

Encrypted Request:

{
"data":"< encrypted json data…….>"
}

After Decryption Sample Request Posted

{
"TransId": "IND4521EDFDF113434131442134D34D34FF",
"custRefNo": "701245124574",
"amount": "0.0",
"txnAuthDate": "NA",
"responseCode": "00",
"approvalNumber": "000000",
"status": "Pending",
"payerVPA": "sk@indus",
"payeeVPA": "rk@indus",
"orderNo": "1495085619883943",
"txnNote": "NA",
“checksum”:”6E00E309B6EBF2BE49A64DE331C4F3030263F8F86AF97CEF8873C4996C9553D55FD51A9A91
7E1EBB7092E8C063233AC61ECE3D3A775A0B64516A7281B9F1F6F99CAACE3DC14D54ACC127AFF16DF2

639D378F4862C26AD45167113EB64E0C8AE3AC04F310C560FFBA448D67079C9CDBAD0B502AD69156E26F75AEDDDAFDFF7A7E80CEF759F889F16DE33A1F08E54DA2B21A2C543790B1BDC320120A1BB777AD76DF7B89631B6B3D9AC90C072B9E5A9D59D465B9409A8F39BC08E78BE5916ED42DDD3C60676EE7A196ACBA02C3537829A74B1B0B769D0E67B1CD222F626100BAC749CD08C70C72890382C98073C3B41D813D054BBDD3F8F1C2F4F6FB75AE8603D6"

}

3.2.3 Response Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	ChannelID	String	M	Max 50	Merchant channel ID / name
2	responseCode	String	M	Min:1 Max: 10	Reponse code for the request processed at infrasoft end.
3	Status	String	M	Min:1 Max: 50	Status of response. Success / Failed
4	TxnRefNo	String	M	Max: 12	Call back request transaction ref number

3.2.4 Response Parameter

Sample Response

{
"ChannelID": " merchantPAY ",
"RefNo": "23456789012345",
"ResponseCode" : "L000"
"Status" : "SUCCESS"
}

Request message (in JSON format) will be encrypted with merchant key as per the logic mentioned in section "***Encryption and Request Checksum Logic"***

3.4 Query Transaction

Sample URL : https://ip:port/QueryTransactionRefId/
(Url will be changed at the time of production)

Method : Post

Content-Type : application/JSON

3.4.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Min: 10 Max: 12	Entity Id of the bank. It should be always 'indianbank'
2	refId	String	M	Max-: 35	Merchant unique reference passed in dynamic QR as a "tr" tag
3	txnRefNumber	String	M	Max 35	Merchant generated transaction reference number for Pay or Collect initiation
4	checkSum	String	M		Calculated as given below

The response code will be sent as mentioned below.

3.4.2 Sample Request

Encrypted Request

```
{
        "data":"< encrypted json data…….>"
}
```

After Decryption Sample Request Posted


```
{
   "refId": "XXXF0HO464QQS2PFKQ6H7VBJB6L16",
   "entityId": "inb"
"checkSum":"UvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dauUvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dau"
                }
```


3.4.3 Response Parameter

The switch will query transaction based on refId (refId is received in NPCI ReqPay_Credit as REFID tag. It is payer psp's responsibility to populate refId as "tr" tag of dynamic QR).

Sr No	Request Fields	Datatype	Mandatory	Length	Description

1.	entityId	String	NA	15	It should be “inb” for UAT and “inb” for Prod
2.	appId	String	NA	15	e.g.: “FI00001”
3.	mobileNo	String	NA	12	Subscriber mobile number
4.	txnId	String	NA	Max: 35	NPCI Transaction ID
5.	requestTime	DateTime	NA		Date and time of the transaction
6.	responseStatus	String	NA	Max 50	The status of the transaction. It can be 'COMPLETED', 'PENDING', 'REJECT' or 'FAILED'.
7.	refId	String	NA	35	Merchant unique reference id received in request

3.4.3 Sample Response:


```
{
  "code": "00",
   "result": "Success",
"data": {
"entityId": "inb",
  "appId": "FI00001",
  "mobileNo": "9999999999",
  "txnId": "XXXF0HO464QQS2PFKQ6H7VBJB6L16",
"requestTime": "21/12/2016 16:15:57",
"responseStatus": "PENDING",
  " txnRefId ": "9999999999"
}
}
```

3.5 Refund API

Sample URL : **https://ip:port/refund/ (Url will be changed at the time of production)**
Method : **Post**
Content-Type : **application/JSON**

3.4.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Min: 10 Max: 12	Entity Id of the bank. It should be always ‘inb’
2	txnId	String	M	Max-: 35	NPCI Transaction ID
3	txnRefID	String	C	Max 12	Transaction reference number
4	txnAmount	String	M	Max: 40	Refund amount received in the request
5	OrgTxnRefNo	String	M	Max 12	Orgnial transaction reference no
6	checkSum	String	M		Calculated as given below

The response code will be sent as mentioned below.

3.4.2 Sample Request

Encrypted Request

```
{
        "data":"< encrypted json data…….>"
}
```

After Decryption Sample Request Posted

```
{
  "txnId": "XXXF0HO464QQS2PFKQ6H7VBJB6L16",
   “entityId”: ”inb”,
```

"txnAmount":"10.00",
"txnRefID":"150220221234",
"txnId":"XXXF0HO464QQS2PFKQ6H7VBJB6L16",
"OrgTxnRefNo":"123345567812"
"checkSum":"UvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dauUvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dau"
}

3.4.3 Response Parameter:

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	NA	15	It should be "inb" for UAT and "inb" for Prod
2	orgTxnId	String	NA	35	Unique transaction Id generated by merchant while initiating request for collect Request
3	responseStatus	String	NA	150	The status of the transaction. It can be 'COMPLETED', 'PENDING', 'REJECT' or 'FAILED'.
4	txnAmount	String	NA	14	The amount refunded to the payer
5	txnRefID	String	NA	12	The refund transaction refer ID

3.4.3 Sample Response:

{
"code": "00",
"result": "Success",
"data": {
"entityId": "inb",
"txnId": "XXXF0HO464QQS2PFKQ6H7VBJB6L16",
"responseStatus": "Success",
"txnAmount": "10.00",
"txnRefID": "123345567812"
}
}

3.6 Collect API

Sample URL : **https://ip:port/collect/ (Url will be changed at the time of production)**
Method : **Post**
Content-Type : **application/JSON**

3.6.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Min: 10 Max: 15	Entity Id of the bank. It should be always 'INB'
2	appId	String	M	Min: 10 Max: 15	App Id of the bank. It should be always 'INB'
4	merchantCode	String	M	Min: 10 Max: 25	Merchant code of the merchant. Ex. "M123456"
5	subMerchantCode	String	O	Min: 10 Max: 15	Submerchant code of the submerchant.Ex. "SM123456"
6	mcc	String	M	Max: 4	mcc code at the time of registration
7	payerAddr	String	M	Min: 10 Max: 99	Payer VPA
8	payerName	String	M	Min: 10 Max: 150	Payer Name
9	payerMobileNo	String	M	12	Payer mobile number (10 digits)
10	payerAccNo	String	O	12	Payer account number
11	Expdate	Date	M	Date	Collect Expiry date & time (07-12-2021 14:15)
12	txnAmount	String	M	Max 14	Txn amount that was specified while raising the collect request.
13	currencyCode	String	M	Min: 1 Max: 18	The currency of the transaction defaulted to INR
14	Remark	String	M	Min: 1 Max: 50	Remark

15	Refurl	String	M	Min: 1 Max: 50	Reference URL of the transaction http:// merchant.com
16	txnRefNumber	String	M	Min 12	Merchant Transaction reference number generated by merchant.
17	TxnId	String	C	Max 35	Transaction ID generated by the merchant for processing collect request.
18	authToken	String	O	12	Token received at the time of authentication.
19	checkSum	String	M		Calculated as given below

3.6.2 Sample Request

Encrypted Request:

```
{
    "data":"< encrypted json data…….>"
}
```

After Decryption Sample Request Posted

```
{
  "entityId": " INB'",
  "appId": " INB'",
  "merchantCode":"M123456",
  "subMerchantCode":"SM123456"
  "mcc": "8891",
  "payerAddr": "infra1@indbank",
  "payerName": "NA",
  "payerMobileNo": "NA",
  “payerAccNo”:”234456679012”,
  “expdate”:” 07-12-2021 14:15”,
  "txnAmount": "1.15",
  "currencyCode": "INR"
  "remark": "payment made",
  "refUrl": "NA",
  "txnRefNumber": "400004941051",
  "authToken": "Test",
  "checkSum":"UvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dauUvVlR//FlV
4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dau"
}
```

3.6.3 Response Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	Code	String	M	Min:1 Max: 10	The response code for the request is processed at the infrasoft end.
2	Result	String	M	Min:1 Max: 10	Status of response. Success / Failed
4	OrgTxnId	String	M	Min:1 Max: 35	Transaction ID generated by infrasoft for processing collect request by Merchant.
5	TxnRefNumber	String	M	Max: 12	Unique transaction Id generated by the merchant while initiating the request for collect Request
6	Amount	String	M	Max: 14	Amount raised in collect request generated by merchant app/PG

3.6.4 Sample Response


```
{
    "code": "00",
     "result": "Success",
  "data": {
        "orgTxnId": "CBIWLG5UKR4S0PB23MDDBB6MPMN9ET4"
        "txnRefNumber": "400004941051",
         "amount": "1.15",
  }
  }
```


Failure sample response:


```
{
"code": "01",
   "result": " Failed",
   "data": {
      "orgTxnId": "CBIWLG5UKR4S0PB23MDDBB6MPMN9ET4"
      "txnRefNumber": "400004941051",
       "amount": "1.15",
   }
}
```

3.7 Pay API

Sample URL : **https://ip:port/pay/ (Url will be changed at the time of production)**
Method : **Post**
Content-Type : **application/JSON**

3.7.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Min: 10 Max: 15	Entity Id of the bank. It should be always 'INB'
2	appId	String	M	Min: 10 Max: 15	App Id of the bank. It should be always 'INB'
4	merchantCode	String	M	Min: 10 Max: 25	Merchant code of the merchant. Ex. "M123456"
5	subMerchantCode	String	O	Min: 10 Max: 15	Submerchant code of the submerchant. Ex. "SM123456"
6	Mcc	String	M	Max: 4	MCC code at the time of registration
7	payerAddr	String	M	Min: 10 Max: 99	Payer VPA
8	payerName	String	M	Min: 10 Max: 150	Payer Name
9	payerMobileNo	String	M	12	Payer mobile number (10 digits)
10	payerAccNo	String	M	12	Payer account number
11	payerCode	String	M	4	Payer code
12	payeeAddr	String	M	Min: 10 Max: 99	Payee address
13	payeeName	String	C	Max: 150	Payee name
14	payeeCode	String	C	4	Paye code
15	txnAmount	String	M	Max 14	Txn amount that was specified while raising collect request.
16	currencyCode	String	M	Min: 1 Max: 6	The currency of the transaction defaulted to INR
17	Remark	String	M	Min: 1 Max: 50	Remark

18	Refurl	String	M	Min: 1 Max: 50	Reference URL of the transaction http:// merchant.com
19	txnRefNumber	String	M	35	Transaction reference number generated by merchant.
20	Timestamp	Date	M		Current timestamp of the request.Should be in YYYY-MM-ddTHH:mi:SSG
21	InitiationMode	Numeric	C	2	Initiation Mode
22	PurposeCode	Numeric	C	2	Purpose Code
23	authToken	String	O	12	Token received at the time of authentication.
24	checkSum	String	M		Calculated as given below

3.7.2 Sample Request

Encrypted Request:


```
{
        "data":"< encrypted json data…….>"
}
```


After Decryption Sample Request Posted


```
{
  "entityId": " INB'",
  "appId": " INB'",
  "merchantCode":"M123456",
  "subMerchantCode":"SM123456"
  "mcc": "8891",
  "payerAddr": "infra1@indbank",
  "payerName": "NA",
  "payerMobileNo": "NA",
  “payerAccNo”:”234456679012”,
  “payerCode”:”0000”,
  “payeeAddr”:”james@inb”,
  “payeeName”:”JAMES”,
  “payeeCode”:”0000”,
  "txnAmount": "1.15",
  "currencyCode": "INR"
  "remark": "payment made",
  "refUrl": "NA",
  "txnRefNumber": "UCB40000494105",
 “Timestamp”:”2022-03- 15T13:33:52+05:3 0”,
“initiationMode”:”00”,
“purposeCode”:”00”,
"authToken": "Test",
"checkSum":"UvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dauUvVlR//FlV4
FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dau"
```

}

3.7.3 Response Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	Code	String	M	Min:1 Max: 10	Response code for the request processing at the infrasoft end.
2	Result	String	M	Min:1 Max: 10	Status of response. Success / Failed
4	OrgTxnId	String	M	Min:1 Max: 35	Transaction ID generated by infrasoft for processing pay requests by Merchant.
5	TxnRefNumber	String	M	Max: 12	Unique transaction Id generated by the merchant
6	Amount	String	M	Min:1 Max: 14	Transaction amount
7	Timestamp	Date	M	Current timestamp of the response	Current timestamp of the request.Should be in YYYY-MM-ddTHH:mi: SSG

3.7.4 Sample Response


```
{
  "code": "00",
   "result": "Success",
   "data": {
        "orgTxnId": "INBWLG5UKR4S0PB23MDDBB6MPMN9ET4"
        "txnRefNumber": "UCB40000494105",
         "amount": "1.15",
        “Timestamp”:”2022-03- 15T13:33:54+05:3 0”
      }
      }
```


Failure sample response:


```
{
"code": "01",
    "result": " Failed",
    "data": {
       "orgTxnId": "INBWLG5UKR4S0PB23MDDBB6MPMN9ET4"
       "txnRefNumber": "UCB40000494105",
       "amount": "1.15",
       “Timestamp”:”2022-03- 15T13:33:54+05:3 0”
     }
```

}

3.8 Validate Address API

Sample URL : **https://ip:port/validateadd/ (Url will be changed at the time of production)**
Method : **Post**
Content-Type : **application/JSON**

3.8.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Min: 10 Max: 15	Entity Id of the bank. It should be always 'INB'
2	appId	String	M	Min: 10 Max: 15	App Id of the bank. It should be always 'INB'
3	merchantCode	String	M	Min: 10 Max: 25	Merchant code of the merchant. Ex. "M123456"
4	subMerchantCode	String	O	Min: 10 Max: 15	Submerchant code of the submerchant. Ex. "SM123456"
5	mcc	String	M	Max: 4	MCC code at the time of registration
6	payerAddr	String	M	Min: 10 Max: 99	Payer VPA
7	payerMobileNo	String	M	12	Payer mobile number (10 digits)
8	payerAccNo	String	M	12	Payer account number
9	payerCode	String	M	4	Payer code
10	payeeAddr	String	M	Min: 10 Max: 99	Virtual Address of the Payee for resolution of registered Name
11	Remark	String	M	Min: 1 Max: 50	Remark
12	Refurl	String	M	Min: 1 Max: 50	Reference URL of the transaction http:// merchant.com
13	txnRefNumber	String	M	35	Transaction reference number generated by merchant.
14	Timestamp	Date	M		Current timestamp of the request.Should be in YYYY-MM-ddTHH:mi:SSG
15	authToken	String	O	12	Token received at the time of authentication.
16	checkSum	String	M		Calculated as given below

3.8.2 Sample Request

Encrypted Request:

{
"data":"< encrypted json data…….>"
}

After Decryption Sample Request Posted

{
"entityId": " INB'",
"appId": " INB'",
"merchantCode":"M123456",
"subMerchantCode":""
"mcc": "8891",
"payerAddr": "infra1@indbank",
"payerMobileNo": "NA",
“payerAccNo”:”234456679012”,
“payerCode”:”0000”,
“payeeAddr”:”james@inb”,
"remark": "payment made",
"refUrl": "NA",
"txnRefNumber": "UCB40000494105",
“Timestamp”:”2022-03- 15T13:33:52+05:3 0”,
"authToken": "Test",
"checkSum":"UvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dauUvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dau"
}

3.8.3 Response Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	Code	String	M	Min:1 Max: 10	Response code for the request processing at the infrasoft end.
2	Result	String	M	Min:1 Max: 10	Status of response. Success / Failed
3	OrgTxnId	String	M	Min:1 Max: 35	Transaction ID generated by infrasoft for processing pay requests by Merchant.
4	Timestamp	Date	M	Current timestamp of the response	Current timestamp of the request.Should be in YYYY-MM-ddTHH:mi: SSG
5	verifiedName	String	M	Max 150	Verified Name of the Payee registered in the CBS
6	payeeCode	String	M	Max 4	Payee category code
7	IFSC	String	C	Max 11	Payee IFSC

8	accType	String	C	Max 20	Payee account type

3.8.4 Sample Response


```
{
  "code": "00",
   "result": "Success",
   "data": {
        "orgTxnId": "INBWLG5UKR4S0PB23MDDBB6MPMN9ET4"
        "txnRefNumber": "UCB40000494105",
       “Timestamp”:”2022-03- 15T13:33:54+05:3 0”,
      “verifiedName”:”JAMES”,
      “payeeCode”:”0000”,
      “IFSC”:”INBO00011”
      “accType”:”SAVING”
      }
      }
```


Failure sample response:


```
{
"code": "01",
   "result": " Failed",
   "data": {
      "orgTxnId": "INBWLG5UKR4S0PB23MDDBB6MPMN9ET4"
      "txnRefNumber": "UCB40000494105",
      “Timestamp”:”2022-03- 15T13:33:54+05:3 0”,
        “verifiedName”:”JAMES”,
        “payeeCode”:”0000”,
        “IFSC”:”INBO00011”
        “accType”:”SAVING”
   }
}
```

3.9 VPA Deactivation API

This API is used to deactivate the merchant-created VPA.

Sample URL : **https://ip:port/vpadeactivation/ (Url will be changed at the time of production)**
Method : **Post**
Content-Type : **application/JSON**

3.9.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	3	Entity Id of the bank. It should be always 'INB'
2	mobileNo	String	M	12	merchant mobile number (12 digits)
3	paymentAddress	String	M	99	Merchant VPA ID
4	merchantID	String	C	15	Merchant ID
5	checkSum	String	M		Calculated as given below

3.9.2 Sample Request

Encrypted Request:

```
{
        "data":"< encrypted json data…….>"
}
```

After Decryption Sample Request Posted


```
{
    "subAction": "vpaDeactivation",
    "action": " vpaDeactivation ",
    "entityId": "INB",
    "inputParam": {
        "paymentAddress": "navnath221@indianbk",
        "mobileNo": "919421991523",
      "merchantID":"12345678",
      "checkSum":"UvVlR//FlV4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dauUvVlR//FlV
      4FjqrhOeLAfRzqxRE1782Prv2L7WejKuMedFYepnrK0b7dau"
    }
}
```

3.9.3 Response Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	responseCode	String	M	Min:1 Max: 10	The response code for the request is processed at the bank's end
2	result	String	M	Min:1 Max: 10	Status of response. SUCCESS / FAILURE.
3	txnTimedate	Date	M	Timestamp	Transaction date and time
4	txnRefId	String	M	12	VPA deactivation transaction reference no

3.9.4 Sample Response

```
{
    " responseCode": "00",
     "result": "Success",
     "data": {
          " txnRefId ": "UCB40000494105",
           “txnTimedate”:”2022-03- 15T13:33:54+05:3 0”,
         }
}
```

3.10 QR Generation API

This API is consumed by the middleware server to generate the dynamic QR code for cash withdrawal. Below are the details of the API request and response.

API URL	**https://inbuat.kiya.ai:6060/ICCWUtil/Atm/QRservice**
HTTP Method	POST
Request Type	application/json
Response Type	application/json
Pre requisites	UPI switch will provide the SSL certificate

3.10.1 Request Parameter

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	entityId	String	M	Max: 12	Entity Id of the bank. It should be always 'INB'
2	mode	String	C	Max: 2	ATM/MATM Identifier Initiation mode (18)
3	purpose	String	M	Max: 2	Type of ATM. ATM- Metro (12) / Non-Metro (13)
4	merchantId	String	M	Max: 20	Merchant ID of the merchant given by the aggregator to the individual merchant.
5	refID	AN	M	Max 12	Transaction Reference no. 12 digit RRN format. Merchants can refer as order no
6	terminalId	String	C	Max 50	ATM terminal ID to be passed from ATM middleware to UPI switch
7	atm_pincode	String	O	Max 20	Postal PIN code of ATM/POS
8	amount	Numeric	M	Max 16	cash withdrawal amount
9	Checksum	String	M		Calculated as given below.

3.10.2 Sample Request

Encrypted Request {

{"**data**":
"pBmbpBgWF7chSma/jGQDCL2bTgoZpYF1ZyUi8uUV2jFxw9srtZgZzknvnHototG6OXIa3vziGF6MH5xWg1CfcUw/D0AfbSSadLcpMvr6QEm9mYKtAUloskdq3SZMJfiNWwQD5HaEMaYIWMwfHEcaoAGTQWiMwySMh3SXHQlXMjK/RcqsKgFcqSl0CXptW+1EFpuhfWUZDJ2QA/DLTRusO2oq3MDhk9+NnRPNqaW7Ni1qJWILHfzcmECavfcIBLceakqwRm5ROqYGRV7vyd6JgSQ/nSkINGp3ONbFlaEhghIwbV3Z1v/D/Xp4v71aJI2gcAftsavuz0N0/PlR12giwQsfS5txDVPWXjn2BVJetyIjyFdPOHGiyY3rcIGLz5quSF2xqvzR87Cxnb0okMLPsKD5lmRqP3qpbp05pS6QipV6aBIhqk4pG0x7WLlrt87VOVEgmvSXeYFOk0BO/H5Q6F90mbA0mEsxDGTxl9w="}

}

After Decryption Sample Request Posted {

decryptedData={
"entityId": "INB",
"mode": "18",
"purpose": "12",
"merchantId": "12345678904",
"refID": "202206011234",
"terminalId": "INB1234T45",
"atm_pincode": "600044",
"amount": "1000",
"**checkSum**": "75AB7CACE86D4B95935BE1CF7ACCB514A78462C4179B25320A83169D46831D066B567EB25167D387EDF5DD17AFA84F8FF6103184F5122C7F87D5EFB3C5A89DD4"
}

}

Note: Request message (in JSON format) will be encrypted with MERCHANT key as per the logic mentioned in section "***Encryption and Request Checksum Logic"***

3.10.3 Response Parameter:

Sr No	Request Fields	Datatype	Mandatory	Length	Description
1	responseCode	String	M	Min:1 Max: 10	Reponse code for the request processed at Infrasoft end
2	ResponseStatus	String	M	Min:1 Max: 10	Status of response. SUCCESS / FAILURE
4	txnID	String	M	Min:1 Max: 35	Transaction ID of a request generated by UPI switch.
5	refID	AN	M	Max 12	Transaction Reference no. 12 digit RRN format. Merchant can refer as order no
6	merchantId	String	M	Max: 20	Merchant ID of merchant given by aggregator to individual merchant.
7	terminalId	String	O	Max 50	ATM terminal ID to be passed from ATM middleware to UPI switch
8	QRDateTime	Date	M	Max 20	QR generation date and time eg format :DDMMYYHHMMSS
9	QRdata	AN	M	Max 500	Signed QR string - upi://pay?pa=inbatm@indianbk&pn=INBATMCashWithdrawal&tid=INB4b1798d08c0c4701bded20deaff0f761&tr=025119026916&tn=INB01408&am=500.00&mode=18&purpose=12&orgid=159054&msid=010620221982&mc=6013&mid=600006&sign=MEUCIBw+1HOZ+ZE54M50bCOdsrNqRr+vDBZLIW8yVyFdQvlIAiEAxDTKZkzdAqXNVmzWZ8TMzZm43AIfOW8NE9oqZOPF4lM=

The response code will be sent as mentioned below:

Status	Response code
SUCCESS	00
FAILURE	01

3.10.4 Sample Response:

Success sample response:

{
"responseCode":"00",
"ResponseStatus":"Success",

“data”:{
“txnID”:” INBde92b19862e44226b2ff318c8462c6a9”,
"refID":"202206011234",
"merchantId":"12345678904",
"terminalId":"INB1234T45",
“QRDateTime”:”010622100222”,
“QRData”:”
upi://pay?pa=inbatm@indianbk&pn=INBATMCashWithdrawal&tid=INB4b1798d08c0c4701bded20deaff0f761&tr=025119026916&tn=INB01408&am=500.00&mode=18&purpose=12&orgid=159054&msid=010620221982&mc=6013&mid=600006&sign=MEUCIBw+1HOZ+ZE54M50bCOdsrNqRr+vDBZLIW8yVyFdQvlIAiEAxDTKZkzdAqXNVmzWZ8TMzZm43AIfOW8NE9oqZOPF4lM=”
}
}

Failure sample response:

{
“responseCode”:”01”,
“ResponseStatus”:”Failure”,
“data”:{
“txnID”:” INBde92b19862e44226b2ff318c8462c6a9”,
"refID":"202206011234",
"merchantId":"12345678904",
"terminalId":"INB1234T45"
}
}

4 Checksum and Request Encryption Logic

All the requests and responses will have checksum and encryption applied.

4.1 Checksum Logic

Only for callback use The SHA256 Hashing algorithm to create a checksum for other API calls use the SHA512 Hashing algorithm to create a checksum. The hashing is applied to the input parameters and then converted to hexadecimal equivalent.

Example:

If an API has below mentioned properties in JSON.

Onboarding – Sample logic (reset of the API merchant has to check and implement the same)

generateCheckSumKey ("entityId-" + entityId + "|merchantId-" + merchantId + "|mobileNo-" + mobileNo + "|paymentAddress-"+ paymentAddress + "|merchantAccountNo-" + merchantAccountNo + "|accountType-" + accountType+ "|IFSC-" + IFSC + "|merchantLegalName-" + merchantLegalName + "|channelId-" + channelId+ "|aggregatorCode-" + aggregatorCode + "|MCC-" + MCC + "|terminalId-" + terminalId, "73127505498180881483015890950210");

Java sample code refers under the code snippet section.

CheckStatus

```
public static String checkOnboardStatus() {
  check = generateCheckSumKey("entityId-" + entityId + "|merchantId-" + merchantId + "|mobileNo-" +
  mobileNo + "|paymentAddress-"+ paymentAddress + "|merchantAccountNo-" + merchantAccountNo +
  "|accountType-" + accountType+ "|IFSC-" + IFSC + "|merchantLegalName-" + merchantLegalName +
  "|channelId-" + channelId+ "|aggregatorCode-" + aggregatorCode + "|MCC-" + MCC + "|terminalId-" +
  terminalId,"73127505498180881483015890950210");
  return check;
}
```

Validate

```
public static String validateVPA() {
validation = generateCheckSumKey("entityId-" + entityId + "|appId-" + appId + "|merchantCode-" +
merchantCode+ "|subMerchantCode-" + subMerchantCode + "|mcc-" + mcc + "" + "|payerAddr-" +
payerAddr+ "|payerMobileNo-" + payerMobileNo + "|" + "payerAccNo-" + payerAccNo + "|payerCode-" +
payerCode+ "|payeeAddr-" + paymentAddress + "|remark-" + remark + "|" + "refUrl-" + refUrl +
"|txnRefNumber-"+ txnRefNumber + "|" + "Timestamp-" + Timestamp + "|authToken-" + authToken +
"|payerName-" + payerName+ "|payeeName-" + payeeName, "73127505498180881483015890950210");
return validation;
}
```

Collect

public static String Collect() {

collect = generateCheckSumKey("entityId-" + entityId + "|appId-" + appId + "|merchantCode-" + merchantCode+ "|subMerchantCode-" + subMerchantCode + "|mcc-" + mcc + "|payerAddr-" + payerAddr + "|payerName-"+ payerName + "|payerMobileNo-" + payerMobileNo + "|" + "txnAmount-" + txnAmount + "|currencyCode-"+ currencyCode + "|remark-" + remark + "|refUrl-" + refUrl + "|txnRefNumber-" + txnRefNumber+ "|authToken-" + authToken + "|ExpDate-" + ExpDate, "731275054981808814830158909502l0");

return collect;

}

Query

public static String queryTransaction() {

queryTrn = generateCheckSumKey("entityId-" + entityId + "|txnId-" + txnId, "E92C1F814B3AE367E4E1476A29398B10");

return queryTrn;

}

Refund

public static String reFund() {

refund = generateCheckSumKey("entityId-" + entityId + "|txnId-" + txnId + "|txnAmount-" + refundAmt + "", "731275054981808814830158909502l0");

return refund;

}

Deactivate

public static String deactivateVPA(){

deactive = generateCheckSumKey("entityId-" + entityId + "|mobileNo-" + demobileNo + "|" +"paymentAddress-" + paymentAddress + "","731275054981808814830158909502l0");

return deactive;

}

Call back sample logic

Data 1: TransId + custRefNo + amount + txnAuthDate + responseCode + approvalNumber + status + payerVPA + payeeVPA + orderNo+TransactionNote

generateCheckSumKey("TransId-" + TransId + "|custRefNo-" + custRefNo + "|amount-" + amount+ "|txnAuthDate-" + txnAuthDate + "|responseCode-" + responseCode + "|approvalNumber-" + approvalNumber+ "|status-" + status + "|" + "payerVPA-" + payerVPA + "|payeeVPA-" + payeeVPA + "|orderNo-" + orderNo+ "|" + "txnNote-" + txnNote, "7312750549818088148301589095021**0")**

Data 2: KEY for Checksum (Checksum generator: 7312750549818088148301589095021**0**
Encryption and Decryption: 01428169FE856B02EA3998A4E0C92D84)

Then checksum will be calculated by concatenating this to get checksum.

Checksum :

6E00E309B6EBF2BE49A64DE331C4F3030263F8F86AF97CEF8873C4996C9553D55FD51A9A917E1EB
B7092E8C063233AC61ECE3D3A775A0B64516A7281B9F1F6F99CAACE3DC14D54ACC127AFF16DF26
39D378F4862C26AD45167113EB64E0C8AE3AC04F310C560FFBA448D67079C9CDBAD0B502AD6915
6E26F75AEDDDAFDFF7A7E80CEF759F889F16DE33A1F08E54DA2B21A2C543790B1BDC320120A1B
B777AD76DF7B89631B6B3D9AC90C072B9E5A9D59D465B9409A8F39BC08E78BE5916ED42DDD3C60
676EE7A196ACBA02C3537829A74B1B0B769D0E67B1CD222F626100BAC749CD08C70C72890382C98
073C3B41D813D054BBDD3F8F1C2F4F6FB75AE8603D6

Code Snippet

```
public static String generateCheckSumKey(String stcheckSumdata, String stShaKey) {
        String chkSumKey = null;
        String stCheckType = "SHA-512";
        CheckSumGeneratorMswipe hmac = new CheckSumGeneratorMswipe();
        chkSumKey = hmac.Sha(stcheckSumdata, stShaKey, stCheckType);
        return chkSumKey;
 }

 public String Sha(String checkSumdata, String key, String CheckType) {
        Mac sha512_HMAC = null;
        String result = null;
        String HMAC_SHA = null;
```

```
            try {
                byte[] byteKey = key.getBytes("UTF-8");
                if (CheckType.equals("SHA-512")) {
                    HMAC_SHA = "HmacSHA512";
                }
                if (CheckType.equals("SHA-256")) {
                    HMAC_SHA = "HmacSHA256";
                }
                sha512_HMAC = Mac.getInstance(HMAC_SHA);
                SecretKeySpec keySpec = new SecretKeySpec(byteKey, HMAC_SHA);
                sha512_HMAC.init(keySpec);
                byte[] mac_data = sha512_HMAC.doFinal(checkSumdata.getBytes("UTF-8"));
                result = bytesToHex(mac_data);
                System.out.println(result);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return result;
    }
    public static String bytesToHex(byte[] bytes) {
            final char[] hexArray = "0123456789ABCDEF".toCharArray();
            char[] hexChars = new char[bytes.length * 2];
            for (int j = 0; j < bytes.length; j++) {
                int v = bytes[j] & 0xFF;
                hexChars[j * 2] = hexArray[v >>> 4];
                hexChars[j * 2 + 1] = hexArray[v & 0x0F];
            }
            return new String(hexChars);
    }
```

4.2 Encryption

The AES encryption will be used to transfer the data between the bank and MERCHANT. MERCHANT shall provide the key for encryption and decryption while integration of this solution.

Example of Request

For any API, the data in encrypted format will be as follows:

Step 1 : data + checksum

TransId: INB4521EDFDF113434131442134D34D34FF,
custRefNo: 701245124574,
amount: 0.0,
txnAuthDate: NA,
responseCode: 00,
approvalNumber: 000000,
status: Pending,
payerVPA: sk@indiabnk,
payeeVPA: rk@indianbnk,
orderNo: 1495085619883943,
txnNote: NA,
checksum:6E00E309B6EBF2BE49A64DE331C4F3030263F8F86AF97CEF8873C4996C9553D55FD51A9A917E1EBB7092E8C063233AC61ECE3D3A775A0B64516A7281B9F1F6F99CAACE3DC14D54ACC127AFF16DF2639D378F4862C26AD45167113EB64E0C8AE3AC04F310C560FFBA448D67079C9CDBAD0B502AD69156E26F75AEDDDAFDFF7A7E80CEF759F889F16DE33A1F08E54DA2B21A2C543790B1BDC320120A1BB777AD76DF7B89631B6B3D9AC90C072B9E5A9D59D465B9409A8F39BC08E78BE5916ED42DDD3C60676EE7A196ACBA02C3537829A74B1B0B769D0E67B1CD222F626100BAC749CD08C70C72890382C98073C3B41D813D054BBDD3F8F1C2F4F6FB75AE8603D6

Step 2 - Convert request into JSON format

"{
"TransId": "IND4521EDFDF113434131442134D34D34FF",
"custRefNo": "701245124574",
"amount": "0.0",
"txnAuthDate": "NA",
"responseCode": "00",

"approvalNumber": "000000",
"status": "Pending",
"payerVPA": "sk@indianbnk ",
"payeeVPA": "rk@indianbnk ",
"orderNo": "1495085619883943",
"txnNote": "NA",
“checksum”:”6E00E309B6EBF2BE49A64DE331C4F3030263F8F86AF97CEF8873C4996C9553D55FD51A9A917E1EBB7092E8C063233AC61ECE3D3A775A0B64516A7281B9F1F6F99CAACE3DC14D54ACC127AFF16DF2639D378F4862C26AD45167113EB64E0C8AE3AC04F310C560FFBA448D67079C9CDBAD0B502AD69156E26F75AEDDDAFDFF7A7E80CEF759F889F16DE33A1F08E54DA2B21A2C543790B1BDC320120A1BB777AD76DF7B89631B6B3D9AC90C072B9E5A9D59D465B9409A8F39BC08E78BE5916ED42DDD3C60676EE7A196ACBA02C3537829A74B1B0B769D0E67B1CD222F626100BAC749CD08C70C72890382C98073C3B41D813D054BBDD3F8F1C2F4F6FB75AE8603D6”
}

Step 3 - Encrypted request using AES encryption:

data:wxp3XXcqmY9JD4f+onyy1z2wze98b09h5d7BQFvjQflig4JRqKXEnapaD+tDwbl2Mnp/JJ4yzHRj\r\ns9t9yT9wy9nuN2VI+rElSyv2iS1BD0VymFpcY8tLPD3zE7vmPyd949R6sesZ+fqa3cxcigdJqi4Q\r\nt2l7trbZxwZUe0shFGAVeyRAa0eaiOfEYF/CxCUq41OVksYVXvjkGBuZ1TH54ZtTPCHQ9Reb6ojJ\r\nCQOL1uqs1KJAPKQCIesaK7vXQi0MTnwrzxY70ICS7tFLGcMIthHh9PH5KK0FPqbrJerVjT4/sEAL\r\nV6JUbJaTqLm2PXFS/6gyZ/r9ZwOSQtS+xcb5o99Pz8Gbg1AIMRrjWYVXqmbZt4++NX8IbwAH3WFS\r\nMe8BhEZ8QTi4DS79+ldjbGpIcDkFatrrZPwR9gLiiCNRevMozjvlEhMlCf5G5PXs/qQVb/wQ/vRL\r\neyee/02rNEtb0JByU6kcSoGkpt84Q9XEJzF8NlwozNxvOWb12if9lZVBljNUHBPA/W1+ONHEduF+\r\n3mG0o4TaxoFtX+Cl11d/D873yu9Fth2OyKS7giweC8ulMIuVI/sNZyjgEjO/LKPrG1cZQqpDGkyH\r\nkvD33mVaqeDAIxlfBQn+9y+HjtFjtBQripN2Zqq6Y3ffjspgue/rRzqTzW4KimgAA6xnYZhheTjw\r\nGrIyH1u6NTyisSDRUd1Abls0m69cLRk42kY6hnvw3Mif1NtIB+kXgahUpdsmfrGtOcjr0FFWFLX5\r\nEtmHNeKrz8s9Y4n2xM472IcmC/Hw+kaPJifA2hAq65IpkVkwHMmm29H7SoMgNTxgKsSQsNV/VlAx\r\nXHLCYcUTeC1QIuVmH36o+MxOEK4xnHbHlwgdUDqRUT/NHPwxgFL6DWNIwxwyhrrCc0tJyhGCmDp+\r\n7XHUze5siX9ItrEMcPSfbEU6U3xmzlOiF7QTe1N2DoGlbNldD9xapjNvVKon/dj48XMWQKDoe6Rq\r\nLVW1BVebGixFgYCCvl1QNu8Kq4KtRkylJeQh/N1dc5Dp2XClfA7QigRZQhpv8dFysiomEgWwXIuy\r\nA1zHIiLX/OQVkBEXPEL5dvA+dVoI2HZgOsw3TSab47qoNBMBB13AiDliayL1rwBlf3qpqqq8FuzR\r\ngQ41U8sMBKrUVtVVkbr1cswZalo7IH+I8A== "

Step 4 – Convert encrypted request into JSON format

{
"data":"wxp3XXcqmY9JD4f+onyy1z2wze98b09h5d7BQFvjQflig4JRqKXEnapaD+tDwbl2Mnp/JJ4yzHRj\r\ns9t9yT9wy9nuN2VI+rElSyv2iS1BD0VymFpcY8tLPD3zE7vmPyd949R6sesZ+fqa3cxcigdJqi4Q\r\nt2l7trbZxwZUe0shFGAVeyRAa0eaiOfEYF/CxCUq41OVksYVXvjkGBuZ1TH54ZtTPCHQ9Reb6ojJ\r\nCQOL1uqs1KJAPKQCIesaK7vXQi0MTnwrzxY70ICS7tFLGcMIthHh9PH5KK0FPqbrJerVjT4/sEAL\r\nV6JUbJaTqLm2PXF

S/6gyZ/r9ZwOSQtS+xcb5o99Pz8Gbg1AIMRrjWYVXqmbZt4++NX8IbwAH3WFS\r\nMe8BhEZ8QTi4DS79+
ldjbGpIcDkFatrrZPwR9gLiiCNRevMozjvlEhMlCf5G5PXs/qQVb/wQ/vRL\r\neyee/02rNEtb0JByU6kcSoGkpt
84Q9XEJzF8NlwozNxvOWb12if9lZVBljNUHBPA/W1+ONHEduF+\r\n3mG0o4TaxoFtX+Cl11d/D873yu9Fth
2OyKS7giweC8ulMIuVI/sNZyjgEjO/LKPrG1cZQqpDGkyH\r\nkvD33mVaqeDAIxlfBQn+9y+HjtFjtBQripN2Zq
q6Y3ffjspgue/rRzqTzW4KimgAA6xnYZhheTjw\r\nGrIyH1u6NTyisSDRUd1Abls0m69cLRk42kY6hnvw3Mif1
NtIB+kXgahUpdsmfrGtOcjr0FFWFLX5\r\nEtmHNeKrz8s9Y4n2xM472IcmC/Hw+kaPJifA2hAq65IpkVkwHM
mm29H7SoMgNTxgKsSQsNV/VlAx\r\nXHLCYcUTeC1QIuVmH36o+MxOEK4xnHbHlwgdUDqRUT/NHPwx
gFL6DWNIwxwyhrrCc0tJyhGCmDp+\r\n7XHUze5siX9ItrEMcPSfbEU6U3xmzlOiF7QTe1N2DoGlbNldD9xa
pjNvVKon/dj48XMWQKDoe6Rq\r\nLVW1BVebGixFgYCCvl1QNu8Kq4KtRkylJeQh/N1dc5Dp2XClfA7QigR
ZQhpv8dFysiomEgWwXIuy\r\nA1zHIiLX/OQVkBEXPEL5dvA+dVoI2HZgOsw3TSab47qoNBMBB13AiDliay
L1rwBlf3qpqqq8FuzR\r\ngQ41U8sMBKrUVtVVkbr1cswZalo7IH+I8A== "
}

Code snippet

Encryption

INIT_VECTOR= 9568463295684632

KEY= 01428169FE856B02EA3998A4E0C92D84

```
public static String encryptMaster(String data, String key) {
String encryptedValue = "";
String initVector = AppConstants.INIT_VECTOR;
int GCM_TAG_LENGTH =16;

try {
GCMParameterSpec gcmParameterSpec = new GCMParameterSpec(GCM_TAG_LENGTH * 8,
initVector.getBytes());
SecretKeySpec skp = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
cipher.init(Cipher.ENCRYPT_MODE, skp, gcmParameterSpec);
byte[] encrypted = cipher.doFinal(data.getBytes());
encryptedValue = Base64.getEncoder().encodeToString(encrypted);

} catch (Exception e) {
e.printStackTrace();
}
return encryptedValue;
}
```

Decryption

```
public static String decryptMaster(String data, String key) {
```

```
String decryptedData = "";
String initVector = 9568463295684632;
int GCM_TAG_LENGTH =16;
try {
GCMParameterSpec gcmParameterSpec = new GCMParameterSpec(GCM_TAG_LENGTH * 8,
initVector.getBytes());
SecretKeySpec skp = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
cipher.init(Cipher.DECRYPT_MODE, skp, gcmParameterSpec);
byte[] decrypted = cipher.doFinal(Base64.getDecoder().decode(data));
decryptedData = new String(decrypted);

} catch (Exception e) {
e.printStackTrace();
}
return decryptedData;
}

public static String encryptMaster(String data, String key) {
String encryptedValue = "";
String initVector = 9568463295684632;
int GCM_TAG_LENGTH =16;

try {
GCMParameterSpec gcmParameterSpec = new GCMParameterSpec(GCM_TAG_LENGTH * 8,
initVector.getBytes());
SecretKeySpec skp = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
cipher.init(Cipher.ENCRYPT_MODE, skp, gcmParameterSpec);
byte[] encrypted = cipher.doFinal(data.getBytes());
encryptedValue = Base64.getEncoder().encodeToString(encrypted);

} catch (Exception e) {
  e.printStackTrace();
}
return encryptedValue;
}
```