भारतीय राष्ट्रीय भुगतान निगम
NATIONAL PAYMENTS CORPORATION OF INDIA

NPCI/2024-25/NACH/003

October 20, 2024

To

All NACH member banks

Emandate Authentication through Simplified Aadhaar

Reference may be taken from Circular No. 9 (NPCI / 2021-22 / NACH / Circular no. 009, dated Feb 10, 2022) regarding Emandate Authentication through Aadhaar based authentication and Circular No. 3 (NPCI / 2023-24 / NACH / 003, dated July 21, 2023) regarding E-mandate simplification and harmonization of the limit of all variants of mandates.

In addition to Aadhaar based authentication where applicable limit is of Rs. 1.00 crore (Rupees one crore), we have introduced Simplified Aadhaar with limit of Rs. 15,000 where UIDAI authentication will not be required and banks shall send OTP to the customer on registered mobile number for authentication, post successful OTP validation mandate will be registered.

Validation of Aadhaar linkage to the account: It is mandatory for the banks to validate that the given Aadhaar number in the mandate registration request is linked to account number in which mandate is registered. Register mandate only if Aadhaar is already linked to the account. This validation is mandatory for Aadhaar based mandates as well as for Simplified Aadhaar mandate.

Aadhaar based mandate (where mandate value is above 15,000 up to Rs. 1 Cr): It may please be noted that in case of Aadhaar based mandate the "uidaiAuthenticated" flag value shall be 'Y' the banks must necessarily validate this value before taking up the mandate request for processing and registration. If the value is anything other than 'Y' such registration requests shall be rejected by the bank.

Simplified Aadhaar mandate (where mandate value is up to Rs. 15,000): For authentication under Simplified Aadhaar mechanism banks shall follow the existing validations of Aadhaar based authentication with the only difference of "uidaiAuthenticated" flag value can be N or Y.

NPCI shall depending on the regulatory and other approvals will decide on the following:

1. UIDAI validation requirement for different categories of Aadhaar based mandate
2. Limit of amount for simplified mandate.

Detailed process flow is provided in Annexure I and Technical specifications are provided in the Annexure-II. Member banks are advised to take note and disseminate the information to all concerned for implementation.

With warm regards,

SD
Giridhar G. M
Chief - Customer Success

1001A, The Capital, B Wing, 10th Floor,
Bandra Kurla Complex, Bandra (E), Mumbai 400 051.
T: +91 22 40009100 F: +91 22 40009101
contact@npci.org.in www.npci.org.in
CIN: U74990MH2008NPL189067

Annexure-I

भारतीय राष्ट्रीय भुगतान निगम
NATIONAL PAYMENTS CORPORATION OF INDIA

NPCI Mandate Approval Gateway Service

Bank Specification Document

Version 1.0.

DOCUMENT RELEASE NOTICE

Document Details

Name	Version No.	Date	Description
Bank Specification Document for Simplified Aadhaar	First	18-10-2024	Provides technical & operation specification for Banks to develop compatible application at their end for communicating with the Mandate Authorization application

Contents

1. Introduction 5
Abbreviation 5
2. Interface specification details for Mandate Approval 6
Registration with NPCI 6
3. Simplified Aadhaar Based Authentication Flow 6
4. API services 16
API to get Transaction Status for Banks 16

1. Introduction

This document details the requirement for destination banks to develop the required interface for interacting with the Mandate Authorization gateway service.

The file formats for request & response are covered in this document.

Abbreviation

The below abbreviations are used in the document.

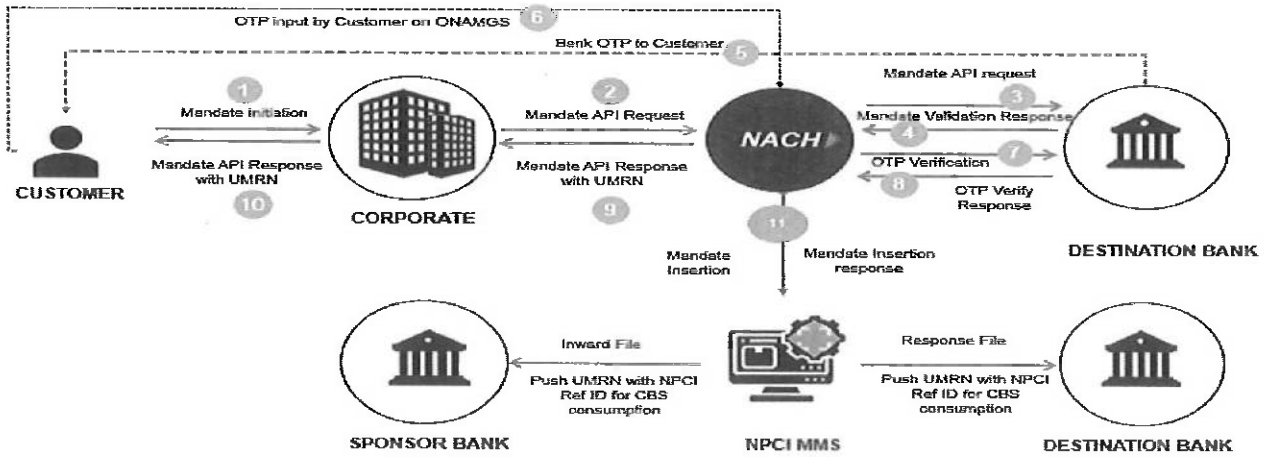
NPCI	National Payments Corporation of India
ONMAGS	Online Mandate Approval Gateway Service
UIDAI	Unique Identification Authority of India

2. Interface specification details for Mandate Approval

Registration with NPCI

The destination banks who want to leverage the service need to be registered with NPCI and get certified.

3. Simplified Aadhaar Based Authentication Flow



Step1: Customer has initiated the request via Merchant Portal i.e., Web Browser

Step2: Customer will be redirected to ONMAGS Platform to enter the details required for Aadhaar authentication. Customer enters Aadhaar Number along with required details.

Step 3: ONMAGS will send an API request to customer's Bank to verify the customer details.

Step 4: Bank will validate the mandate request and send to NPCI

Step 5: Customer will be landed on ONMAGS OTP page. Banks will generate the OTP and send it to customer for Authentication.

Step 6: Customer will enter the Bank OTP in ONMAGS platform for Authentication.

Step 7: ONMAGS platform will forward that OTP to destination bank for Verification.

Step 8: If OTP verification is successful then only the Bank needs to mark the mandate as accepted at their end. Until OTP validation is passed the mandate would be in non-accepted state at the Bank end.

Step 9: ONMAGS Platform in turn redirects the response to Merchant Web Page where customer can view the response.

Step 10: Response is shared by corporate to customer

Step 11: Mandate will be inserted in MMS System and Inward & Response will be shared with Sponsor and Destination Bank

Annexure II

Auth mode: Aadhaar
Privilege: Initiated by ONMAGS (NPCI)
API type: Sync
Request Type: JSON
HTTP Method: POST

Parameter Specification

Parameters	Data Type	Description
mandateAuthDtls	JSON Object	This will contain mandate Request details and aadhaar Info
transactionID	String	This is used for the complete transaction for mandate registration. ALPNUM String with Length is 20.
mandateRequestDtl	JSON Object	This will contain Encrypted mandate Request Doc XML and Encrypted checksum value.
MandateReqDoc	String	See below table for Mandate Request Doc.
CheckSumVal	String	How to generate Checksum value is mentioned above.
authMode	String	authMode value will be Aadhaar and user will get aadhaarInfo JSON Object in request.
aadhaarInfo	JSON Object	This will contain the aadhaar details and flag indicating that the customer authentication has been successful though UIDAI. In case of simplified mandate this value can be either 'Y' or 'N'.
aadhaarNo	String	Last four digit of aadhaar number
uidaiAuthenticated	Char	(In case of simplified mandate this value can be either 'Y' or 'N'.)

Mandate Request to Bank

```
{
   "mandateAuthDtls": {
      "transactionID": "<Transaction ID>",
      "mndtType":"<CREATE / AMEND /CANCEL / SUSPEND/ REVOKE /CUSTOM_CANCEL
attributes>",
      "mandateRequestDtl": {
         "MandateReqDoc": "<Encrypted and Signed request XML>",
         "CheckSumVal": "<Check sum value of secure attributes>"
      },
      "authMode":"Aadhaar",
      "aadhaarInfo": {
         "aadhaarNo": "<Encrypted Aadhaar Number>",
         "uidaiAuthenticated" : "Y or N"
      }
   }
}
```

Note:- Checksum will be validated only for Create & Amend flow

Unencrypted and Unsigned request XML for MandateReqDoc Key:

Element Name	Validation	Data Type	Length	Remarks
Xmlns	Namespace tag. This is mandatory tag. Value cannot be empty. Namespace value should be "http://npci.org/ONMAGS/schema"	Alpha Numeric		
NPCI_RefMsgId	NPCI_RefMsgId from NPCI should be unique	Alpha Numeric	35	Message ID for NPCI Reference
CreDtTm	Should be in ISO Date time format. E.g.2017-02-09T15:11:39	Alpha Numeric	25	
ID	Request Initiating Party ID. In this case it will be Corporate / Merchant ID. Should not be null. Will be validated if this is a valid Merchant ID with the master.	Alpha Numeric	18	ID & UtilCode value would be the same.
UtilCode	Utility Code would be validated against the masters. It should be 18 digit Utility code.	Alpha Numeric	18	ID & UtilCode value would be the same.
CatCode	Identifies under which category the mandate is created. Will be validated against the masters maintained by NPCI	Alpha Numeric	4	
Name	Should not be empty	Alpha Numeric	40	Corporate Name.
Spn_Bnk_Nm	Corporate Sponsor Bank Name	Alpha Numeric	140	Should be a valid Bank Name as per MMS
CatDesc	Category Description should correspond to Category Code in the Master	Alpha Numeric	50	
MndtReqId	Mandate Req ID length should be <= 35. Should be unique for the day	Alpha Numeric	35	
MndtId	This tag will contain the UMRN generated in MMS for the mandate.	Alpha Numeric	20	UMRN
Mndt_Type	Mandate Type	Alpha	35	Should be DEBIT
Schm_Nm	Scheme Name / Plan Reference Number	Alpha Numeric	20	
SeqTp	Allowed values are RCUR or OOFF	Alpha Numeric	4	
Frqcy	This is an optional field. If present should adhere to the list value available in MMS Masters.	Alpha Numeric	4	Allowed Values are: ADHO, INDA, DAIL, WEEK, MNTH, QURT, MIAN, YEAR, BIMN
FrstCollitnDt	Date of First Collection. Mandatory Field. This field is in ISODate Format	Alpha Numeric	16	
FnlCollitnDt	Date of Final Collection. Optional Field. This field is in ISODate Format	Alpha Numeric	16	If this field is left blank then deduction will happen until Cancelled.
CollitnAmt	Either of CollitnAmt or MaxAmt is mandatory. Amount Should be given as 100.00	Alpha Numeric	13	

MaxAmt	Either of ColltnAmt or MaxAmt is mandatory Amount Should be given as 100.00	Alpha Numeric	13	
Debtor Nm	Customer name should be maximum of 35 digit	Alpha Numeric	40	
Debtor AccNo	Customer Account Number should be maximum of 35 digit.	Alpha Numeric	35	
Acct_Type	Debtor Account Type	Alpha	35	Should be either of SAVINGS or CURRENT
Cons_Ref_No	Consumer Reference Number	Alpha Numeric	20	
Phone	Phone Number of the Customer	Alpha Numeric	34	Should be given in the format +91-xxx-xxxxxxx. +91- is mandatory.
Mobile	Mobile Number of the Customer	Alpha Numeric	34	Should be given in the format +91-xxxxxxxxx. +91- is mandatory.
Email	Email ID of the Customer	Alpha Numeric	50	Should be valid email id
Pan	Pan Number of the Customer	Alpha Numeric	27	Should be in Valid PAN format
Creditor Nm	Corporate Name. Length will be 40	Alpha Numeric	140	
Creditor AccNo	Will be the 18 digit Corporate ID	Alpha Numeric	18	
MmbId	Will be 11 digit IFSC code	Alpha Numeric	11	IFSC Code of the Sponsor Bank which is available in the ONMAG Live Bank list
MndtId	Will be 20 digit UMRN	Alpha Numeric	20	Except Create Flow
ReasonCode	will be 4 digit Reason code	Alpha Numeric	4	Except Create Flow

Bank needs to first verify the mandate request details

a) If the destination bank is unable to parse the mandate request it will send the response in the below format. Bank need not validate the aadhaar details if sending failure response (because of request XML validation failure at bank end).

Parameters	Datatypes	Description
mandateVerifyDtls	JSON Object	Mandate verify details contains transaction ID, mandate Validation and mandate reject details
transactionID	String	This is the same transaction ID which Is passed in request for mandate registration. ALPNUM String with Length is 20.

mndtType	String	This will contain the operation AMEND / CANCEL / SUSPEND/ REVOKE /CUSTOM_CANCEL. mndtType will not be present for Create Flow.
mandateValidation	String	This will return either success or failure.
aadhaarValidation	String	This will return either success or failure.
mandateRejectDtl	JSON Object	This will contain error code and error desc
ErrorCode	Integer	This will be between 000 to 999
ErrorDesc	String	This will be the corresponding error description for the error code.
signature	String	The Response payload will be signed with bank's private key and algorithm used as ***RSA_USING_SHA256***
checkSumVal	String	Generate checksum on the entire payload. We will use SHA-2 as the hash function

Error Response from Bank for Mandate request:

```
{⊟
   "mandateVerifyDtls": {⊟
      "transactionID": "<Transaction ID>",
      "mndtType":"<CREATE / AMEND /CANCEL / SUSPEND/ REVOKE /CUSTOM_CANCEL
attributes>",
      "mandateValidation": "failure",
      "aadhaarValidation": "none",
      "mandateRejectDtl": {⊟
         "ErrorCode": "<Error Code>",
         "ErrorDesc": "<Error Description>"
      }
},
      "signature": "<Encrypted and Signed response JSON>",
      "checkSumVal": "<Check sum value of complete payload>"


}
```

Note:- Attribute values Mandate Validation, Aadhaar Validation, Error Code & ErrorDesc needs to be encrypted. Bank needs to encrypt using NPCI public key.

b) If destination bank is able to successfully parse the mandate request XML but business validation of XML fails, then bank needs to send the response in the below format. Aadhaar details need not be validated in such a scenario.

```
{
   "mandateVerifyDtls": {
      "transactionID": "<Transaction ID>",
      "mndtType":"<CREATE / AMEND /CANCEL / SUSPEND/ REVOKE /CUSTOM_CANCEL
attributes>",
      "mandateValidation": "failure",
      "aadhaarValidation": "none",
```

```
        "mandateRejectDtl": {
           "ReasonCode": "<Reason Code>",
           "ReasonDesc": "<Reason Description>"
        }
    },
        "signature": "<Encrypted and Signed response JSON>",
        "checkSumVal": "<Check sum value of complete payload>"

    }
```

Note:- Attribute values Mandate Validation, Aadhaar Validation, Reason Code, Reason Desc & checkSumVal needs to be encrypted

c) Aadhaar Validation

1. Aadhaar number of debtor should matches with the "Aadhaar linked with the Debtor AccNo" provided in the mandate Request XML
2. Aadhaar number should be linked with the debtor Account Number.

If the above validation fails, then the bank needs to provide the response as above format 2nd type.

```
{⊟
   "mandateVerifyDtls": {⊟
      "transactionID": "<Transaction ID>",
      "mndtType":"<CREATE / AMEND /CANCEL / SUSPEND/ REVOKE /CUSTOM_CANCEL
attributes>",
      "mandateValidation": "success",
      "aadhaarValidation": "failure",
      "mandateResponseDtl": {⊟
         "accptRefNo": "<Accept Reference Number>",
          "dbtrIfsc": "<Debtor IFSC>",
          "dbtrAcctType": "<Debtor Account Type>"
      },
      " aadhaarRejectDtl ": {⊟
         "ReasonCode": "<Reason Code>"
      }
},
       "signature": "<Encrypted and Signed response JSON>",
       "checkSumVal": "<Check sum value of complete payload>"

}
```

The below table provides the error codes for different failure reasons.

Failure Reason	Reason Code
Aadhaar number Does not Match with debtor Account number	AP48
Aadhaar Number not linked with the Debtor Account Number	AP51

d) If all the above validation passes then the bank needs to provide the success response as below:-

Success Response for Mandate request to Bank:

```
{
   "mandateVerifyDtls": {
      "transactionID": "<Transaction ID>",
      "mndtType":"<CREATE / AMEND /CANCEL / SUSPEND/ REVOKE /CUSTOM_CANCEL
attributes>",
      "mandateValidation": "success",
       "aadhaarValidation": "success",
      "mandateResponseDtl": {
         "accptRefNo": "<Accept Reference Number>",
          "dbtrIfsc": "<Debtor IFSC>",
          "dbtrAcctType": "<Debtor Account Type>"
      },
      "aadhaarVerifyDtl": {
         "successCode": "<Success Code>"
      }
},
      "signature": "<Encrypted and Signed response JSON>",
      "checkSumVal": "<Check sum value of complete payload>"


}
```

The below table provides the code for the success

Success Reason	Success Code
Aadhaar number matches with Aadhaar linked with debtor account number Validation Passed	000

Note:-

i. Attribute values mandateValidation, aadhaarValidation, AccptRefNo , successCode &b checkSumVal needs to be encrypted.

ii. Bank needs to store the mandate details received along with the transaction ID for the subsequent OTP validation.

For scenarios (a), (b) and (c) ONMAGS will construct the merchant rejection response and redirect to the merchant. Bank needs to mark the mandate as rejected at their end for these scenarios. For scenario (d) if bank has opted for OTP validation then mandate status will be "In Process" for the bank until the OTP verification is completed, else mandate status will be "Accept" and send the response back to ONMAGS.

For scenario (d) ONMAGS will redirect to the OTP verification page.

Below are the steps to be done for securing the content of the Response JSON:

1. Generating checksum for the secure information in the Response JSON (Mandate and Aadhaar validation).

The below attributes need to be concatenated for the purpose of generating Checksum:

A. Transaction ID
B. Mandate Validation
C. Accepted Ref No.

D. Dbtr Account type
E. Dbtr IFSC
F. Reason Code
G. Reason Desc
H. Error Code
I. Error Desc
J. Aadhaar Validation
K. Success Code
L. Aadhaar Reason Code
M. Aadhaar Error Code
N. JSON Web Signature

2. Generating checksum for the secure information in the Response JSON (OTP Validation).

The below attributes need to be concatenated for the purpose of generating Checksum:

A. Transaction ID
B. Verify Status
C. Error Code
D. Reason Code
E. JSON Web Signature

The above attributes need to be concatenated with "|" symbol appended as the delimiter. The order of the attributes needs to be as mentioned above.

Note: The attributes to be concatenated might be changed at a later point of time. Please refer the latest version of the document for any revision on the attributes that needs to be marked for Generate checksum on the concatenated values. We will use SHA-2 as the hash function.

3. Signing of the Response JSON.

- The complete response we are going to use as a payload.
- The response JSON has to be signed using the Private Key certificate of the Bank.
- Json Web Signature is used for generating digital signatures and the same will be validated at the NPCI end.

Note :

- Except transaction ID, Dbtr Account Type and Dbtr IFSC field, all the fields are encrypted. For generating checksum, we are going to use encrypted values. If value is not present in response, then we will use empty string for that key.
- Since we are using Signature value while generating Checksum, so that first we need to sign the response then generate checksum.

Bank OTP Verification Request for Same Mandate request:-

Parameters	DataTypes	Description
otpInfo	JSON Object	This will contains the transaction Id same used in OTP generation and Encrypted OTP which is received on registered mobile in bank
transactionID	String	Same transaction ID used in mandate request to bank. ALPNUM String with Length is 20.
otp	String	Encrypted OTP received on registered mobile in the bank. Length is 4.

```
{
   "otpInfo": [
      {
         "transactionID": "<Transaction ID>",
         "otp": "<Encrypted OTP Value>"
      }
   ]
}
```

- In case of retry request also, the request will be posted to bank in the above mentioned format only.

The encryption on the OTP will follow the existing encryption methodology. Bank needs to decrypt the OTP and verify it based on the transaction ID. The OTP verification status needs to be sent in the below json format by the bank.

Response From Bank for Bank OTP Verification for the same Mandate request:

Parameters	DataTypes	Description
otpVerifyInfo	JSON Object	This will contain the same transaction Id which is sent in mandate request to bank and encrypted status as success or failure.
transactionID	String	Transaction Id is the same Which is sent in verify request bank OTP. ALPNUM String with Length is 20.
optVerifyStatus	String	Encrypted OTP verification status. It will be either success / failure

a) If OTP verification at bank end is success then the response will be as below:

```
{
   "otpVerifyInfo":
      {
         "transactionID": "<Transaction ID>",
         "optVerifyStatus": "success",
        "errorCode" : "",
         "reasonCode": ""
      },
      "signature": "<Encrypted and Signed response JSON>",
      "checkSumVal": "<Check sum value of complete payload>"

}
```

b) If OTP verification failed at bank end, then response will be as below:

```
{
        "otpVerifyStatus": {
                "transactionID" : "<Transaction ID>",
                "optVerifyStatus":"failure",
                "errorCode":"",
                "reasonCode" : <Reason Code>
        },
        "signature":"<Encrypted and Signed response JSON>",
        "checkSumVal" : "<Check sum value of complete payload"
}
```

Failure Reason	Reason Code
Invalid Bank OTP	AP39
Maximum tries exceeded for OTP	AP40
Time expired for OTP	AP41
Bank Aadhaar OTP Verification response Failed	AP50

If OTP verification is successful only Bank needs to mark the mandate as accepted at their end. Until OTP validation is passed the mandate would be in non-accepted state at the Bank end.

If OTP validation is failure User would be provided with option of reattempting OTP validation further 2 times. An alert message as below will be shown to the user. User can then proceed with entering the correct OTP again and re-verify.

Request for Resend Bank OTP:

Parameters	Datatypes	Description
mandateAuthDtls	JSON Object	This will contains transaction Id same which is sent in the first generate bank OTP request and encrypted aadhaar number
transactionID	String	transaction Id same which is sent in the mandate request to bank. ALPNUM String with Length is 20.
aadhaarInfo	JSON Object	This will contains Encrypted aadhaar number
aadhaarNo	String	Encrypted aadhaar number only last four digit.

JSON Request:

```
{
   "aadhaarAuthDtls": {
      "transactionID": "<Transaction ID>",
      "aadhaarInfo": {
         "aadhaarNo": "< Encrypted Aadhaar Number>"
      }
   }
}
```

- Response for Resend Request will be '200' status code.
- If OTP verification is successful only Bank needs to register the mandate as accepted at their end.
- In case OTP verification fails in all the attempts bank can mark the mandate as rejected at their end.

Bank will not generate any OTP, skip the OTP verification step and need to mark the mandate as accepted at their end.

Technical Integration requirement for Aadhaar Authentication

1. Connectivity:

Communication between NPCI to Bank Server with specific port

2. Certificates
   - Bank SSL certificate(FQDNS)
   - Bank Signing certificate
   - One way API handshake

3. Keys exchange for UIDAI Authentication
   - Bank should share their AUA Keys
   - Bank has to share the keys as part of onboarding process, else we will use NPCI AUA Key

4. API services

API to get Transaction Status for Banks

For the purpose of getting the transaction status of a particular transaction or group of transactions for Banks, NPCI ONMAGS would expose a rest service which will accept list of NPCI Transaction Reference Numbers in JSON format. The response of this API will also be in JSON Format. There will be a limitation on the number of items posted per request. Currently the limit is set as 50.

Sample Input JSON:

```
{
  "npcirefmsgID":[
    "000f0f29dc27f00000101b09c5227457f17",
    "000f0f29dc27f00000101b09c5227457E23",
    "000f0f29dc27f00000101b09c5227453S42"
  ]
}
```

Sample Output JSON:


```
{
  " tranStatus ":[
    {
      "npcirefmsgID":"000f0f29dc27f00000101b09c5227457f17",
      "Accptd":"false",
      "AccptRefNo":"tranid3432kkkeke",
      "MndtId":"xxxxxxxxxxxxxxxxxxx",
      "ReasonCode":"343",
      "ReasonDesc":"Invalid Account",
      "RejectBy":"Bank",
      "ErrorCode":"000",
      "ErrorDesc":"NA"
    },
    {
      "npcirefmsgID":"000f0f29dc27f00000101b09c5227457E23",
      "Accptd":"true",
      "AccptRefNo":"tranid352254221",
      "MndtId":"xxxxxxxxxxxxxxxxxxx",
      "ReasonCode":"000",
      "ReasonDesc":"NA",
      "RejectBy":"NA",
      "ErrorCode":"000",
      "ErrorDesc":"NA"

      .
```

```
  },
  {
    "npcirefmsgID":"000f0f29dc27f00000101b09c5227453S42",
    "Accptd":"NULL",
    "AccptRefNo":"NULL",
    "MndtId":"NULL",
    "ReasonCode":"NULL",
    "ReasonDesc":"NULL",
    "RejectBy":"NULL",
    "ErrorCode":"452",
    "ErrorDesc":"No Details available for the requested parameters. Please check the values provided"
  }
 ]
}
```


In case the details provided in the request are invalid then ErrorCde & ErrorDesc will have the corresponding error code & description. For the valid request ErrorCode would be "000" and "ErrorDesc" would be "NA". **API URL would be of the below format:**

https://enach.npci.org.in/apiservices/getTransStatusForBanks

UAT:

https://103.14.161.144/8086/apiservices/getTransStatusForBanks