NPCI
भारतीय राष्ट्रीय भुगतान निगम
NATIONAL PAYMENTS CORPORATION OF INDIA

NPCI/2021-22/NACH/Circular no. 009

Feb 10,2022

To,

All NACH Member banks

E-Mandate through Aadhaar based authentication

Presently, E-mandate registration can be done through Internet Banking or Debit Card based authentication now in order to make the e mandate variant available the segment of the society that does not have access to internet banking or debit card it has been decided to extend the authentication mechanism to Aadhaar Number and OTP based authentication.

The customer will be authenticated first using Aadhaar + OTP through UIDAI, on successful authentication the mandate registration request will be routed to the bank with additional tag of Aadhaar number. The banks will verify whether the Aadhaar is linked to the account number provided in the request, if Aadhaar is linked then send the OTP to the registered mobile number of the customer. On successful verification of OTP mandate will be registered. The detailed technical specification is provided in Annexure I.

Member banks are advised to take note and disseminate the information to all the concerned and start the development process as per the specification provided to get on-boarded.

For any clarification please write to below

Name	Email id	Contact
Mr. Rajasekar K	Rajasekar.Kuppusamy@npci.org.in	99411 34593
Mr. Sivaji Kanumuri	sivaji.kanumuri@npci.org.in	95505 71509
Mr. Pramod Sagar	pramod.sagar@npci.org.in	99086 73657

With warm regards,

Giridhar GM

(Chief- Offline product operations & run technology)

Annexure I - Process flow document E-Mandate through Aadhaar based authentication

eMandate serves as an underlying infrastructure for businesses in India to collect recurring payments without any human intervention. A mandate is a standard instruction that you provide to your issuing bank and other institutions allowing them to automatically debit the mentioned amount from your bank account. eMandate is a convenient way for businesses and their customers to easily manage all the recurring payments like insurance premiums, SIPs, loan instalment collections, etc. This eliminates the hassles of reminders and late penalty charges.

Recurring payments primarily authorized by a physical form that the customer had to fill, sign and submit. This was an operationally expensive and manual process, which took days to process. API based eNACH more commonly known as eMandate. Currently, E-mandate registration can be done through Internet banking or Debit Card based authentication.

Additionally, now a new variant through Aadhaar based authentication is introduced. The registration here will be done through Aadhaar and OTP based authentication.

Major advantages of eMandates:

- **Increase customer retention**: One-time digital authentication allows you to auto-debit your customer's bank account helping your customer enjoy uninterrupted service–in the longer run building customer loyalty and customer retention. No constant payment reminders to your customers

- **Reduces friction in payments:** eMandate allows a business to auto-debit a recurring payment, thereby reducing the friction of a customer logging on to the website/app to make a payment. This assures the business of continuous cash flow that it can rely upon

- **Reduces administrative costs:** As the amount is auto-debited, the business teams do not have to chase the customer to make the payment. This cuts down the operational effort, invoicing effort and cost to the company

- **Auto-reconciliation:** Most of the details are tracked and captured online, hence reducing the time, effort and money involved in buying and maintaining multiple tools

- **Simple and seamless process:** The entire process is simple and just needs an active bank account and Aadhaar number (mobile registered with UIDAI as well as the bank). One-time enrollment for a lifetime of hassle-free service

- **Flexibility in plan:** With eMandate, the ability to debit a customer in the hands of businesses. This allows them to change the payment cycle or skip a cycle if need be. This adds to good user experience

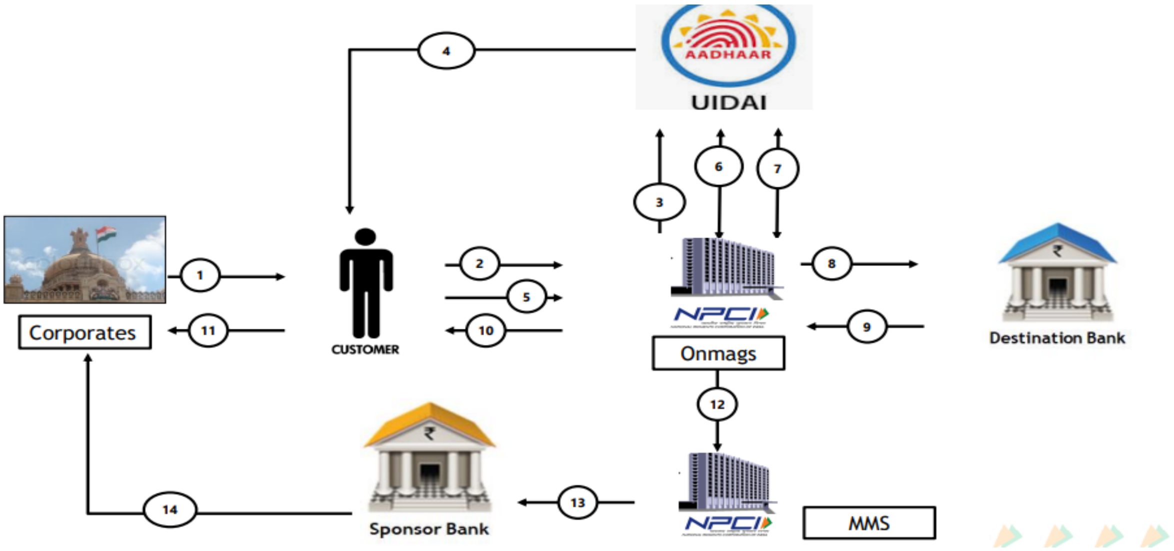
Process flow:

1. Corporates share the link to the customer for mandate registration.
2. Customer clicks the link to register the mandate, provides mandate registration details and selects authentication mode as "Aadhaar based". Once it is selected, the page will be routed to ONMAGS page, and the below will be user experience.
    a) The customer will be able to view the mandate related information.

b) Customer to enter details and Aadhaar number in online mandate page and click on mandate registration process.

3. NPCI will initiate OTP validation of the customer to UIDAI.
4. UIDAI to send the OTP to the registered mobile number of the customer.
5. Customer to enter the OTP on NPCI page.
6. The OTP will be sent to UIDAI for verification and the response will be provided back to ONAMGS.
7. If the OTP authentication is successful, mandate registration API request will be sent to destination bank.
8. Destination bank to validate the mandate details, check if Aadhaar number is linked to account number and other details as per banks internal policy. Post successful validation bank to send an OTP to customer on his registered mobile number as per bank CBS.
9. Customer to enter the OTP in the ONMAGS page.
10. The OTP will be sent to bank for verification and the final response will be received from bank on the acceptance/rejection of the mandate.
11. The page will be re-directed to corporate page along with the status of the mandate.
12. ONMAGS will send update to MMS system to update the mandate details along with UMRN.
13. MMS system at the agreed cut off will send the response file to sponsor bank.
14. Sponsor bank will share the final response to corporate.

Process flow diagram:



DOCUMENT RELEASE NOTICE

Document Details

Name	Version No.	Date	Description
Bank Specification Document	Draft	24-02-2017	Provides technical & operation specification for Banks to develop compatible application at their end for communicating with the Mandate Authorization application
NPCI Mandate Authorization Specification for Banks	1.0	01-03-2017	Updated for Debit Card
NPCI Mandate Authorization Specification for Banks	1.1	22-03-2017	Covered the specification for Signing, Check Sum & Encryption. XML Specification, XSD & XML Samples attached as zip
NPCI Mandate Authorization Specification for Banks	2.0	27-03-2017	Updated for Error Scenarios, HTTP Status codes.
NPCI Mandate Authorization Specification for Banks	3.0	26-05-2017	API to get live destination banks for e-mandate Separate URL's for Net banking & Debit Card Corporate mapping to the destination banks
NPCI Mandate Authorization Specification for Banks	3.1	08-06-2017	Updated the process flow to include bank selection in the merchant page.
NPCI Mandate Authorization Specification for Banks	3.2	24-06-2017	Addition of Dbtr tag in Request XML. Changes in Error Response XML's, Error Codes & Failure Scenarios (In Appendix)
NPCI Mandate Authorization Specification for Banks	3.3	03-Jul-2017	Error Codes & Failure Scenarios Sheet Updated.

			Encryption of Debtor field instead of Creditor. Changes in Server to Server communication specification.
NPCI Mandate Authorization Specification for Banks	3.4	14-Jul-2017	Changes in Request & Response XML formats and Error XML format.
NPCI Mandate Authorization Specification for Banks	3.5	07-Aug-2017	Handling of Timeout Scenario Added
NPCI Mandate Authorization Specification for Banks	3.5	20-Dec-2017	Encryption methodology updated Updates to Offline API's Error Codes Updated
NPCI Mandate Authorization Specification for Banks	3.6	18-Sep-2018	AuthMode added as additional Parameter from Merchant. Flow changes based on this parameter.
NPCI Mandate Authorization Specification for Banks	3.7	15-APR-2019	Change in API "Posting list of Open Transactions to Bank"
NPCI Mandate Authorization Specification for Banks	3.8	17-May-2019	Changes in Error XML Structure from Bank to NPCI and from NPCI to Merchant (Appendix 9.1) Changes in lengths and data types of XML elements in Merchant Request. Changes in Error Response from Bank Change in live bank list api
NPCI Mandate Authorization Specification for Banks	4.0	12-DEC-2019	Changes in Merchant request XML, Bank request XML & Merchant response XML. Encryption of additional fields Encryption of Request XML and Response XML Additional parameter in the form post for Merchant

NPCI Mandate Authorization Specification for Banks	4.1	29-JUL-2020	Introduction of New Debit Card Flow BANKID & AUTHMODE mandatory in merchant request
NPCI Mandate Authorization Specification for Merchant	4.2	29-DEC-2021	Introduction of Aadhaar flow

This document and any revised pages are subject to document control. Please keep them up-to-date using the release notices from the distributor of the document.

Table of Contents

1. Introduction

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

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

1.1 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

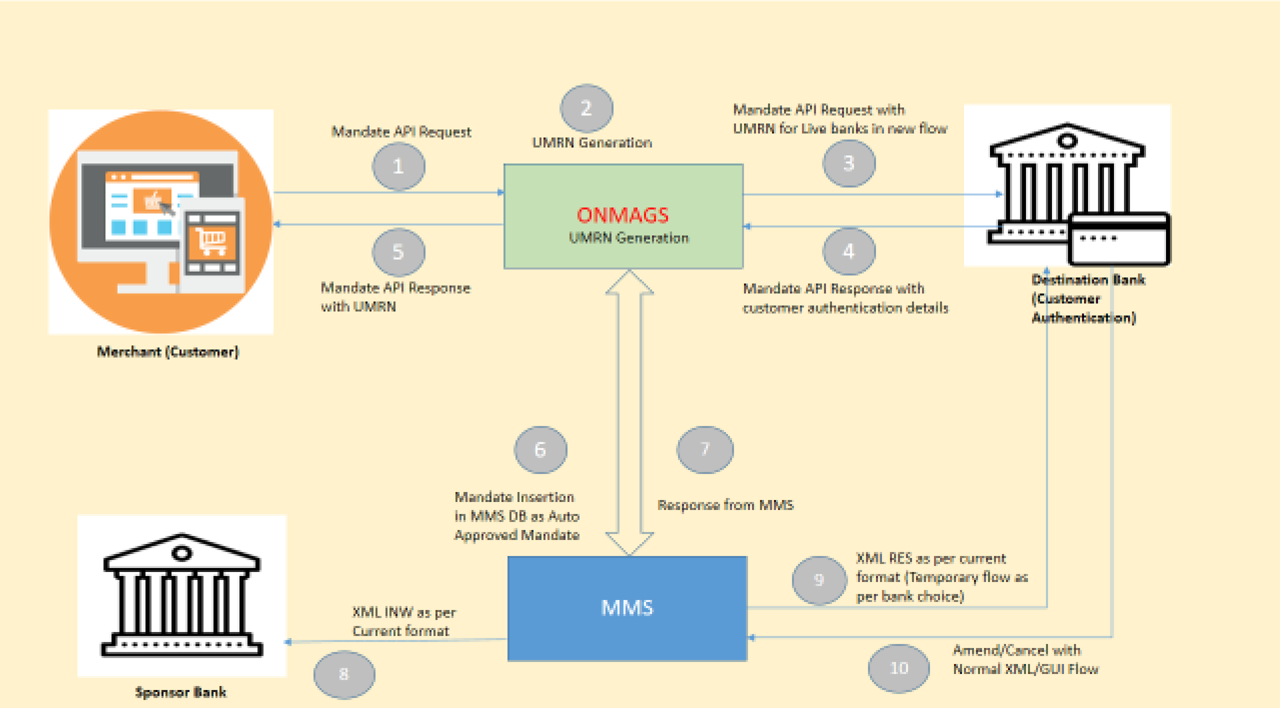
2.1 Registration with NPCI

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

2.2 Mandate Approval function flow (For Net Banking & Debit Card mode of authentication)

The mandate approval flow is initiated from the Merchant end, request validated at NPCI end and forwarded to the Bank for authorization. The confirmation provided back by the Destination Bank is replied back to the merchant.

Mandates created through ONMAGS will be auto registered in MMS. The overall flow and the integration between ONMAGS and MMS systems is explained by the below diagram.

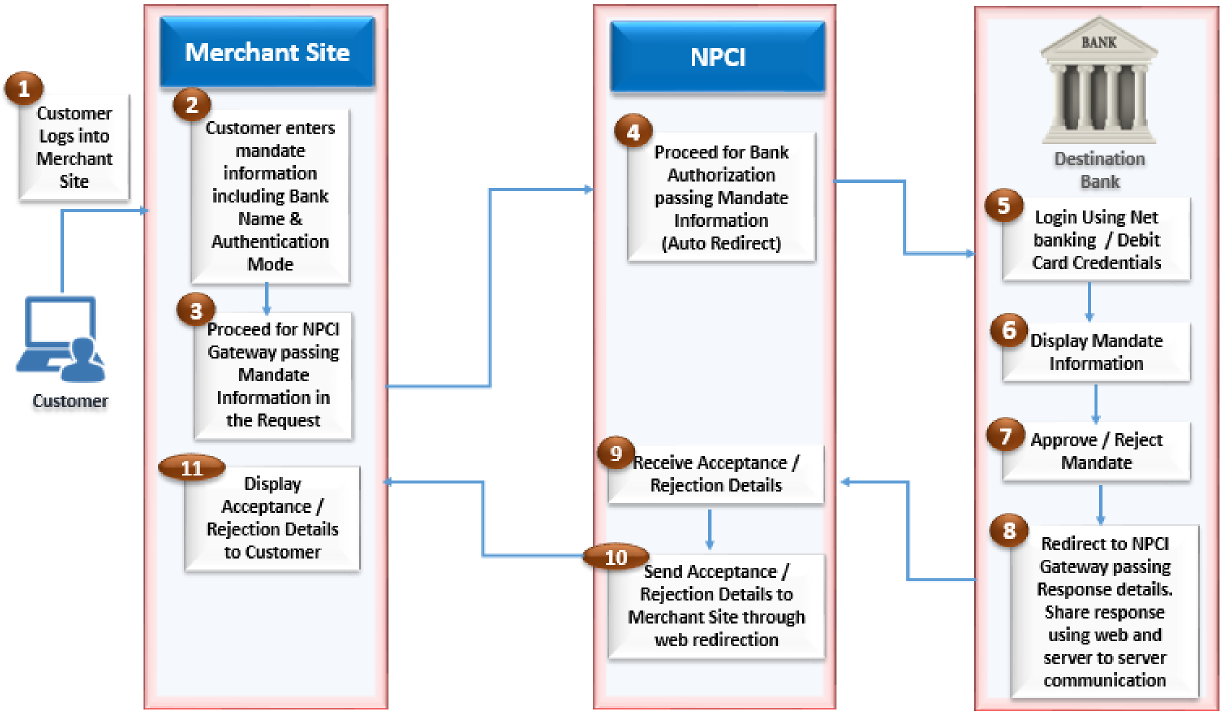

Mandate API Request
1
2
UMRN Generation
Mandate API Request with UMRN for Live banks in new flow
3
ONMAGS
UMRN Generation
5
Mandate API Response with UMRN
4
Mandate API Response with customer authentication details
Merchant (Customer)
Destination Bank (Customer Authentication)
6
Mandate Insertion in MMS DB as Auto Approved Mandate
7
Response from MMS
MMS
9
XML RES as per current format (Temporary flow as per bank choice)
XML INW as per Current format
8
10
Amend/Cancel with Normal XML/GUI Flow
Sponsor Bank

The process flow is mentioned in the next section.

Note:
From version 4.1 BankID & AuthMode are mandatory in the merchant request

2.2.1 End to End Process Flow (For Net Banking & Debit Card mode of authentication)

The below diagram illustrates the functional flow of mandate authorization when Bank ID & Authentication Mode are passed from Merchant. This will be the default flow from version 4.1.


Merchant Site
NPCI
BANK
Destination Bank
1 Customer Logs into Merchant Site
Customer
2 Customer enters mandate information including Bank Name & Authentication Mode
3 Proceed for NPCI Gateway passing Mandate Information in the Request
4 Proceed for Bank Authorization passing Mandate Information (Auto Redirect)
5 Login Using Net banking / Debit Card Credentials
6 Display Mandate Information
7 Approve / Reject Mandate
8 Redirect to NPCI Gateway passing Response details. Share response using web and server to server communication
9 Receive Acceptance / Rejection Details
10 Send Acceptance / Rejection Details to Merchant Site through web redirection
11 Display Acceptance / Rejection Details to Customer

- Customer logins to the merchant site where he/she would be shown the mandate Information
- Specific details of the mandate along with deduction details needs to be shown.
- Customer can proceed with accepting the mandate if he/she finds the information displayed is correct (Customer needs to enter the Bank account number before proceeding)
- Merchant site needs to provide the option for selecting Bank & Authentication Mode. Merchant site has to ensure that the Banks which are listed for authentication are listed in the authorized Bank list of NPCI registered for ONMAGS.
- Customer would be redirected to NPCI ONMAGS interface.
- NPCI Interface would show an intermittent page while processing happens in the back ground.
- If the validation is successful, then NPCI will auto redirect to Bank's authentication page based on the Bank ID & Authentication Mode selected by the end user in the merchant site.
- If the validation fails, then NPCI will redirect back to the Merchant Site posting the Error XML response.
- Once the customer is redirected to Bank's page , the summary of the mandate will be displayed to customer and provided an option for accepting or rejecting the mandate.
- If the customer selected 'Proceed/Accept', further authentication will be done by the Bank. The result of the authentication, either success/failure will then auto redirected to the merchant.
- If the customer selected 'Reject/Cancel', customer will be redirected to merchant site as customer did not want to proceed with the mandate registration.

- Merchant site will display the status of Mandate Approval to the customer.

2.3 Interface Layer

Following certificates needs to be installed at NPCI & Merchant Site to process the transactions.

3. Signing Certificate

4. Specification Format for Request & Response

Appendix 9.1

Lists the XML file format for the request & response.

The specification for below request / response are listed in the document.

The data format would be XML. Schema structure and sample XML's can he found in Appendix 9.3.

- Merchant Mandate Request to NPCI

Merchant site when sending request to NPCI ONMAGS need to send the request in the specified format.

- Response from NPCI to Merchant

NPCI ONMAGS will use this format for sending response back to Merchant Site.

5. Technical Integration Specification

The below section lists few of the technical requirements for the implementation.

5.1 Merchant Site Integration Requirements

5.1.1 Request from Merchant to NPCI

Merchant site will display specific details of the Mandate to the end user.

The below information should mandatorily be displayed apart from other details the merchant site proposes to display.

- Utility Code of Merchant
- Corporate Name
- Consumer Reference Number
- Consumer Name
- Mandate Reference Number
- Amount of Deduction
- Debit Type (Fixed Amount / Maximum Amount)
- Frequency of Deduction
- Start & End Period of Deduction/Valid until Cancelled
- Category

Following information should be obtained from the end user.

- **Bank A/c Number**

  Bank A/c Number of the customer. This would be used by the destination bank to authenticate whether the account which customer logs in by providing the credentials matches with the account number provided in the merchant site.

- **A/c Holder Name**

  Name of the account holder.

- **Destination Bank Selection**

  In case of the Corporates already mapped to specific Destination banks, then the merchant site can provide an option for the user to select the destination bank in the merchant portal itself.

- **Authentication Mode Selection**

Customer should be given an option to choose the mode of authentication that he/she would like to go for. All three authentication modes, which are 'Net Banking', 'Debit Card', 'Aadhaar Number' should be available for selection.

➢ **Account Type**

The account type for the account based on which end user is going to authenticate in Net Banking mode. Possible options are SAVINGS / CURRENT/OTHER.

➢ **Phone Number (Optional)**

End user Phone Number provided for the account based on which authentication is going to be done.

➢ **Mobile Number (Optional)**

End user Mobile Number provided for the account based on which authentication is going to be done.

➢ **Email ID (Optional)**

End user Email ID provided for the account based on which authentication is going to be done.

➢ **PAN Number (Optional)**

PAN Number of the end user

Once user enters the above information they can proceed for Bank Authentication.

A link should be provided for the customer to proceed for Bank Authentication / Approval. On submitting the link, the merchant site should redirect to NPCI ONMAGS interface.

There are two ways NPCI can post information to ONMAGS.

❖ Browser redirection

This will be the URL that merchant needs to use who are redirecting through browser.

Production_URL :

Merchant:

https://enach.npci.org.in/onmags/sendRequest

Bank:

https://enach.npci.org.in/onmags/bankResponse

UAT_URL:

Merchant:

https://103.14.161.144:8086/onmags/sendRequest

BANK:

https://103.14.161.144:8086/onmags/bankResponse

- API Call

For the specific requirement of certain merchants accessing the ONMAGS application as an API call we have provided a separate URL. This will be helpful to merchants who access ONMAGS from their respective app. The response information provided for the API call has to be displayed in the app.

Production:

https://enach.npci.org.in/onmags/sendApiRequest

UAT:

https://103.14.161.144:8086/onmags/sendApiRequest

NPCI will validate the source of the request before processing the request.

For securing the data passed between domains the data passed in the request should be encrypted & signed.

Below are the steps to be done for securing the content of the Request Data passed from Merchant to NPCI:

1. Generating checksum for the secure information in the XML

   The below attributes needs to be concatenated for the purpose of generating Checksum:
   a) Debtor Account Number
   b) First Collection Date
   c) Final Collection Date
   d) Collection Amount
   e) Max Amount

   The above attributes need to be concatenated with “|” symbol appended as the delimiter. The order of the attributes needs to be as mentioned above. In case any of the attribute is null then during concatenation the particular attribute will be replaced by an empty string.

   Since either of MaxAmt or ColltnAmt is mandatory the value of the empty field should be set to empty string before concatenating.

   Same applies to Final Collection Date which can be empty.

   Example:
   If the below are the values for the fields to be concatenated:

Debtor AccNO	: 1023344333
First Collection Date	: 2019-04-29+05:30
Final Collection Date	: 2019-04-29+05:30
Collection Amount	:
Max Amount	: 1000

Then the concatenated string for generating checksum will look as below:
1023344333|2019-04-29+05:30|2019-04-29+05:30||1000

If the below are the values for the fields to be concatenated:

Debtor AccNO	: 1023344333
First Collection Date	: 2019-04-29+05:30
Final Collection Date	:
Collection Amount	: 1000
Max Amount	:

Then the concatenated string for generating checksum will look as below:
1023344333|2019-04-29+05:30||1000|

Note:
The attributes to be concatenated can be changed at later point of time. Please refer the latest version of the document for any revision on the attributes that needs to be marked for encryption.

Generate checksum on the concatenated values. We will use SHA-256 as the hash function.

2. Encrypt secure information in the XML. The below attributes in the XML needs to be encrypted.

- Debtor AccNO
- First Collection Date
- Final Collection Date
- Collection Amount
- Max Amount
- Phone
- Mobile
- E-mail
- PAN

The attributes mentioned above needs to be encrypted individually and placed in the respective XML tags.

Note: -

- Phone, Mobile, Email & Pan are encrypted but not considered in the checksum computation.
- The optional fields Phone, Mobile, Email & Pan needs to encrypted and included in the XML only if value is available for these fields.
- Empty or blank values for these fields should not be added in the XML.

We will use the below methodology for encryption of secure information.

Encryption Methodology – Asymmetric

Hashing Algorithm – SHA256

Cryptography – RSA/ECB/OAEPWithSHA-256AndMGF1Padding 2048 bits

Encryption needs to be done using the Public Key certificate shared by NPCI.

3. Signing of the Request XML

The request XML got from Step-2 has to be signed using the Private Key certificate of the merchant.

Merchant needs to send the below data as MIME content with type as “application/x-www-form-urlencoded” in the request body. The following key-value pair needs to be posted in the body of the request.

Key	**Value**
MerchantID	Participant ID of the Merchant in NACH
SPID (Optional)	In case a merchant has multiple certificates for each of its service providers then the ID of the Service Provider shared with NPCI has to be provided against this parameter
MandateReqDoc	Output of the Step-3
CheckSumVal	Encrypted Output of Step-1
BankID	Valid Participant ID of the Bank.
AuthMode	Will be either of “NetBanking” or “DebitCard” or "Aadhaar"

Note: -

The Merchant ID & SP ID combination would be used to identify the signing certificate of the Merchant. Also the response URL (URL of response to merchant from NPCI) will as well identified by this combination.

Merchant should share the Service Provider ID (SP ID) with NPCI when they go for multiple Certificate’s and URLS for Single Merchant ID.

CheckSumVal should be encrypted using the public key certificate shared by NPCI.

MerchantID in the MIME message should be the same as the <ID> tag under <ReqInitPty> in the Request XML.

Upon Mandate Approval / Rejection at the banking site by the end user NPCI ONMAGS would send the response shared by the bank with the merchant site. In case the request is rejected at NPCI ONMAGS layer itself NPCI will generate the response and send.

NPCI will perform the following validations:

1. Verify if MerchantID is valid

2. If ServiceProviderID is provided verify if it is a valid service provider ID registered against the merchant in NPCI.

3. Verify if Bank ID is a valid Bank ID

4. Verify if AuthMode is a valid AuthMode.

5. Validate the MandateReqDoc xml data.

The format of the merchant request XML is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="http://npci.org/ONMAGS/schema">
   <MndtAuthReq>
      <GrpHdr>
         <MsgId></MsgId>
         <CreDtTm></CreDtTm>
         <ReqInitPty>
            <Info>
               <Id> </Id>
               <CatCode> </CatCode>
               <UtilCode> </UtilCode>
               <CatDesc> </CatDesc>
               <Name></Name>
                                <Spn_Bnk_Nm></Spn_Bnk_Nm>
            </Info>
         </ReqInitPty>
      </GrpHdr>
      <Mndt>
         <MndtReqId></MndtReqId>
                             <Mndt_Type></Mndt_Type>
                             <Schm_Nm><Schm_Nm>
         <Ocrncs>
            <SeqTp></SeqTp>
            <Frqcy></Frqcy>
            <FrstColltnDt></FrstColltnDt>
            <FnlColltnDt></FnlColltnDt>
         </Ocrncs>
         <ColltnAmt Ccy="INR"></ColltnAmt>
         <Dbtr>
            <Nm></Nm>
            <AccNo></AccNo>
                             <Acct_Type></Acct_Type>
                             <Cons_Ref_No></Cons_Ref_No>
                             <Phone></Phone>
                             <Mobile></Mobile>
                             <Email></Email>
                             <Pan></Pan>
         </Dbtr>
         <CrAccDtl>
            <Nm></Nm>
            <AccNo></AccNo>
            <MmbId></MmbId>
         </CrAccDtl>
      </Mndt>
   </MndtAuthReq>
```

```
</Document>
```

Given below are the validations done on the MandateReqDoc xml data. For more details refer to the Appendix Section.

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		
MsgId	MSG ID from the merchant.	Alpha Numeric	35	
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 7 digit OLD ICS or 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	
Mndt_Type	Mandate Type	Alpha Numeric	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 SeqTp is OOF. If SeqTp is RCUR Mandatory field 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
FrstColltnDt	Date of First Collection. Mandatory Field. This field is in ISODate Format (date and time/ only date)	Alpha Numeric	16	Example: 2012-05-17+05:30 / 2012-05-17
FnlColltnDt	Date of Final Collection. Optional Field. This field is in ISODate Format(date and time/ only date)	Alpha Numeric	16	If this field is left blank, then deduction will happen until Cancelled.

				Example: 2012-05-17+05:30 / 2012-05-17
ColltnAmt	Either of ColltnAmt or MaxAmt is mandatory.	Alpha Numeric	13	Entry of 12000 will be considered 120.00 while an entry of 120.00 will be considered as 120.00
MaxAmt	Either of ColltnAmt or MaxAmt is mandatory	Alpha Numeric	13	Entry of 12000 will be considered 120.00 while an entry of 120.00 will be considered as 120.00
Debtor Nm	Customer name	Alpha Numeric	40	Should be maximum of 40 digit
Debtor AccNo	Customer Account Number	Alpha Numeric	35	Should be maximum of 35 digit.
Acct_Type	Debtor Account Type	Alpha Numeric	35	Should be either of SAVINGS or CURRENT
Cons_Ref_No	Consumer Reference Number	Alpha Numeric	35	
Phone	Phone Number of the Customer	Alpha Numeric	34	Should be given in the format +91-xxx-xxxxxxxx. +91- is mandatory.
Mobile	Mobile Number of the Customer	Alpha Numeric	34	Should be given in the format +91-xxxxxxxxxx. +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 140	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. Sponsor Bank mentioned should be mapped to the Corporate at NPCI end. Else the request will be rejected

6. Verify if ChecksumVal matches the decrypted CheckSum value of the encrypted fields.

5.1.1.1 Encoding of Mandate Request XML

As part of prevention of malicious attack from external source, it is recommended to send the XML data in encoded format. This will prevent any malicious content to be introduced in the request XML. All merchants are advised to send XML content in encoded format only going ahead. NPCI will maintain list of merchants who have been certified for sending XML in encoded format and will accept information from the merchant in that format only.

The encoded XML will be in the below format:

```
&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
```

```
&lt;Document xmlns=&quot;http://npci.org/ONMAGS/schema&quot;&gt;
    &lt;MndtAuthReq&gt;
        &lt;GrpHdr&gt;
            &lt;MsgId&gt;&lt;/MsgId&gt;
            &lt;CreDtTm&gt;&lt;/CreDtTm&gt;
            &lt;ReqInitPty&gt;
                &lt;Info&gt;
                    &lt;Id&gt; &lt;/Id&gt;
                    &lt;CatCode&gt; &lt;/CatCode&gt;
                    &lt;UtilCode&gt; &lt;/UtilCode&gt;
                    &lt;CatDesc&gt; &lt;/CatDesc&gt;
                    &lt;Name&gt;&lt;/Name&gt;
                                            &lt;Spn_Bnk_Nm&gt;&lt;/Spn_Bnk_Nm&gt;
                &lt;/Info&gt;
            &lt;/ReqInitPty&gt;
        &lt;/GrpHdr&gt;
        &lt;Mndt&gt;
            &lt;MndtReqId&gt;&lt;/MndtReqId&gt;
                                        &lt;Mndt_Type&gt;&lt;/Mndt_Type&gt;
                                        &lt;Schm_Nm&gt;&lt;Schm_Nm&gt;
            &lt;Ocrncs&gt;
                &lt;SeqTp&gt;&lt;/SeqTp&gt;
                &lt;Frqcy&gt;&lt;/Frqcy&gt;
                &lt;FrstColltnDt&gt;&lt;/FrstColltnDt&gt;
                &lt;FnlColltnDt&gt;&lt;/FnlColltnDt&gt;
            &lt;/Ocrncs&gt;
            &lt;ColltnAmt Ccy=&quot;INR&quot;&gt;&lt;/ColltnAmt&gt;
            &lt;Dbtr&gt;
                &lt;Nm&gt;&lt;/Nm&gt;
                &lt;AccNo&gt;&lt;/AccNo&gt;
                                        &lt;Acct_Type&gt;&lt;/Acct_Type&gt;
                                        &lt;Cons_Ref_No&gt;&lt;/Cons_Ref_No&gt;
                                        &lt;Phone&gt;&lt;/Phone&gt;
                                        &lt;Mobile&gt;&lt;/Mobile&gt;
                                        &lt;Email&gt;&lt;/Email&gt;
                                        &lt;Pan&gt;&lt;/Pan&gt;
            &lt;/Dbtr&gt;
            &lt;CrAccDtl&gt;
                &lt;Nm&gt;&lt;/Nm&gt;
                &lt;AccNo&gt;&lt;/AccNo&gt;
                &lt;MmbId&gt;&lt;/MmbId&gt;
            &lt;/CrAccDtl&gt;
        &lt;/Mndt&gt;
    &lt;/MndtAuthReq&gt;
&lt;/Document&gt;
```

Encoding needs to be applied on the final request XML after encryption of secure tags and signing of XML.

5.1.2 Response from NPCI to Merchant

NPCI will send the below data as MIME content to Merchant with type as “application/x-www-form-urlencoded” in the request body. The request body will contain the following key-value pair.

Key	Value
MandateRespDoc	Encrypted and Signed response XML
CheckSumVal	Check sum value of secure attributes
RespType	Will be either of ErrorXML / RespXML

Merchant Site would get the response either in the format mentioned in sheet “Response from NPCI to Merchant” or in the format “Error XML Resp from NPCI to Mer”.

The scenarios based on which either of RespXML or ErrorXML would be send is listed in the Appendix 9.2.

Merchant site needs to display the status of mandate approval in their mandate confirmation page. The URL for mandate confirmation page needs to be shared with NPCI.

Merchant site needs to do the below steps for validating and reading the XML:

If the RespType is ErrorXML then the response XML would be in the format as mentioned in the sheet “Error XML from NPCI to Merchant”.

ErrorCode contains the error code for the error, and ErrorDesc contains the Error Description of the error.

The ErrorXML will be send in plain XML format.

In case of RespType value being “RespXML” then the response XML will be of the format “Response from NPCI to Merchant”.

Merchant site should un-sign the XML using the public key of NPCI and then decrypt the key fields using the private key of the merchant.

The tag Accptd will specify if the request was Accepted or Rejected (true/false)

In case of Rejection, ReasonCode contains the error code and ReasonDesc contains the description of the errors. In case multiple errors identified then the ReasonCode will list all the error codes as comma separated. ReasonDesc will have the text “Multiple errors detected”.

5.1.2.1 Encoding of Response XML

ONMAGS will send the merchant response in encoded format for merchants who are certified to send request XML in encoded format. Hence merchants who are certified for sending request XML in encoded format also needs to accept response/error XML in encoded format only.

5.2 NPCI Gateway Specification

NPCI ONMAGS will act as the gateway layer during forward flow from merchant site to Bank Site as well as during reverse flow from Bank Site to Merchant Site.

5.2.1 Forward Flow specification from Merchant to NPCI

Upon user submitting the page in the Merchant site for proceeding with Bank Authentication, merchant site will redirect to the NPCI ONMAGS passing the necessary parameters in the Request body. If the source of the request is approved one, NPCI ONMAGS will proceed with validating the request XML. If the validation fails, then response is send back to the merchant site with the error code and error description. The tag <RejectBy> will have the value “NPCI” meaning the rejection happened at NPCI gateway layer. The error response will be in the format “Error XML Resp from NPCI to Merchant”.

- NPCI ONMAGS will verify if the specified BankID is in the approved list. In case the BankID is not part of the approved list the request is rejected and the Merchant will get an error response.
- The valid values for AuthMode are ‘NetBanking’ or ‘DebitCard’ or ‘Aadhaar’.
- In case BankID and AuthMode are not passed in the merchant request then the request will be rejected and the Merchant will get an error response.

Once all the validations are successful then based on the BankID/AuthMode parameter value being sent in the request further processing happens.

An intermittent loading page of NPCI would be shown briefly and then the User will be auto redirected to the Bank’s Authentication Page.

5.2.2 Return Flow specification from NPCI to Merchant

The bank site should redirect back to NPCI ONMAGS layer both on successful or failed authentication. NPCI will share the URL for redirection. The response XML wound be send to NPCI ONMAGS layer in the request body. Bank should send the response in the either of the below format:

- ➢ Response from Bank to NPCI
- ➢ ErrorXML Resp from Bank to NPCI

NPCI ONMAGS layer would validate the response XML received from the Bank. Based on the validation result NPCI ONMAGS would send either of Response XML or Error XML to the Merchant.

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

1. **Generating checksum for the secure information in the XML**

   The below attributes needs to be concatenated for the purpose of generating Checksum:
   a) Accptd
   b) AccptRefNo
   c) ReasonCode
   d) ReasonDesc
   e) RejectBy

   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 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.

2. **Replace the secure information in the XML with the encrypted text.**

   The attributes mentioned above needs to be encrypted individually and placed in the respective XML tags. Encryption should be done using the public key of the certificate which NPCI shares.

   We will use the below methodology for encryption of secure information.

   Encryption Methodology – Asymmetric

   Hashing Algorithm – SHA256

   Cryptography – RSA/ECB/OAEPWithSHA-256AndMGF1Padding 2048 bits.

   Encryption will be done using the Public Key of the certificate shared by the Merchant.

3. **Signing of the Response XML**

The response XML got from Step-2 will be signed using the Private Key certificate of NPCI.

4. **Encoding of Response XML**

The response XML shared to the merchant would be encoded. This applies to the Success Response XML and the Error Response XML.

5.3 Authentication Modes

5.3.1 Net Banking and Old Debit Card Mode of Authentication

In case of Net banking or if the Bank has opted for the old debit card flow, then NPCI ONMAGS would redirect to Bank Page. The URL for redirection for Net banking / Debit Card should be made available to NPCI by the banks.

Once the authentication process completes at bank site, Mandate process continues. If the authentication at Bank site fails, corresponding error message will be sent to the user.

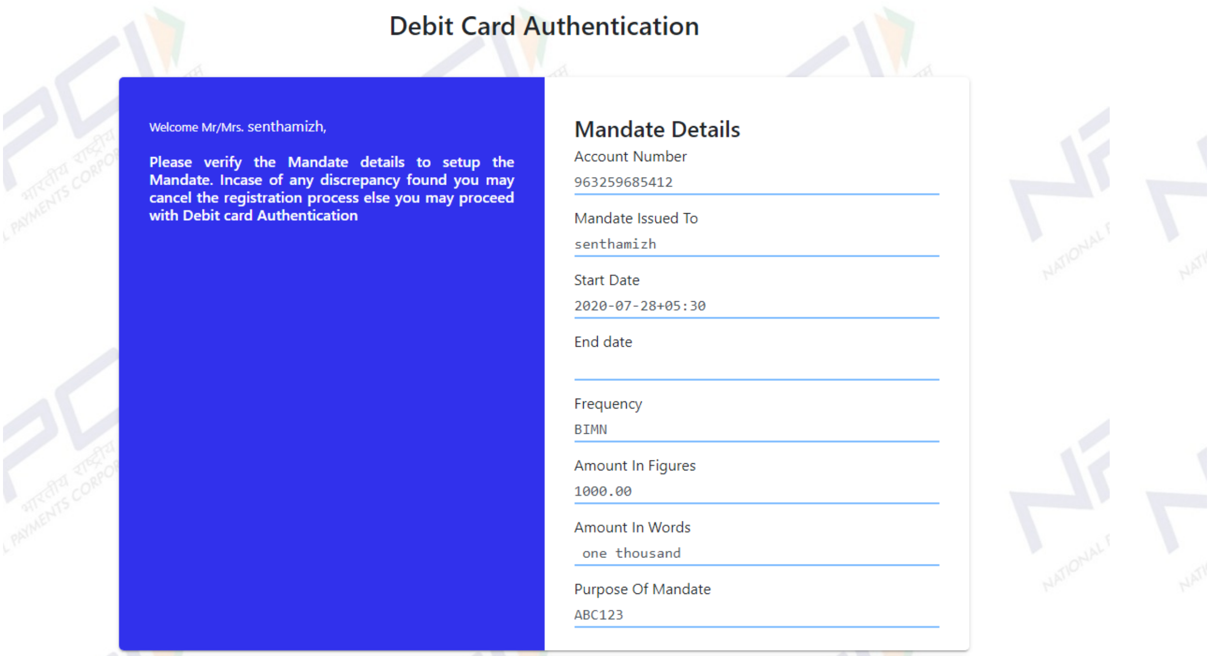
5.3.2 New Debit Card Flow

In case Bank has opted for the new Debit Card Flow, then on redirection from the merchant, the user will be landing in the ONMAGS Debit Card authentication page.

Debit Card Information will be accepted in ONMAGS page itself and validated with Bank through server to server call. The steps in this flow is described below:

On Redirection from merchant the user will be landing on the ONMAGS debit card authentication page.

The mandate information passed by the merchant will be displayed in the top portion of the page.



Debit Card Authentication

Welcome Mr/Mrs. senthamizh,

Please verify the Mandate details to setup the Mandate. Incase of any discrepancy found you may cancel the registration process else you may proceed with Debit card Authentication

Mandate Details

Account Number: 963259685412

Mandate Issued To: senthamizh

Start Date: 2020-07-28+05:30

End date:

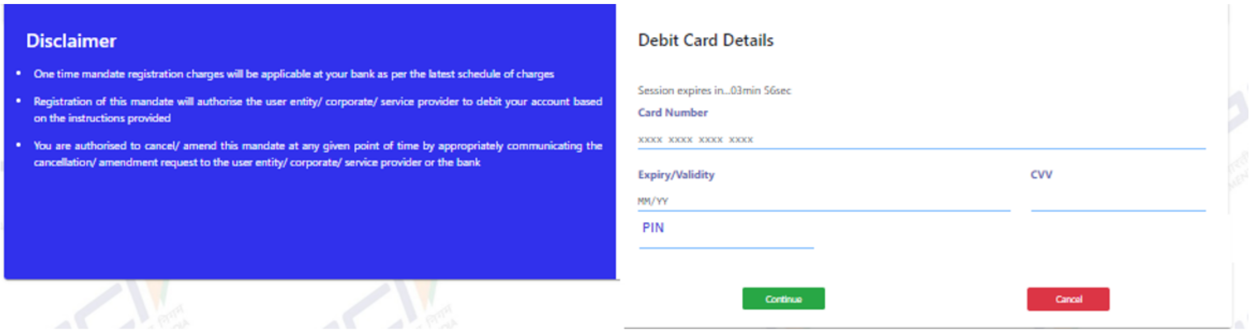
Frequency: BIMN

Amount In Figures: 1000.00

Amount In Words: one thousand

Purpose Of Mandate: ABC123

User needs to verify the mandate information displayed in the Mandate Details section. Once mandate information are verified by the user he can proceed with entering the Debit Card information in the lower section of the page.



Disclaimer

- One time mandate registration charges will be applicable at your bank as per the latest schedule of charges
- Registration of this mandate will authorise the user entity/ corporate/ service provider to debit your account based on the instructions provided
- You are authorised to cancel/ amend this mandate at any given point of time by appropriately communicating the cancellation/ amendment request to the user entity/ corporate/ service provider or the bank

Debit Card Details

Session expires in...03min 56sec

Card Number: xxxx xxxx xxxx xxxx

Expiry/Validity: MM/YY

CVV

PIN

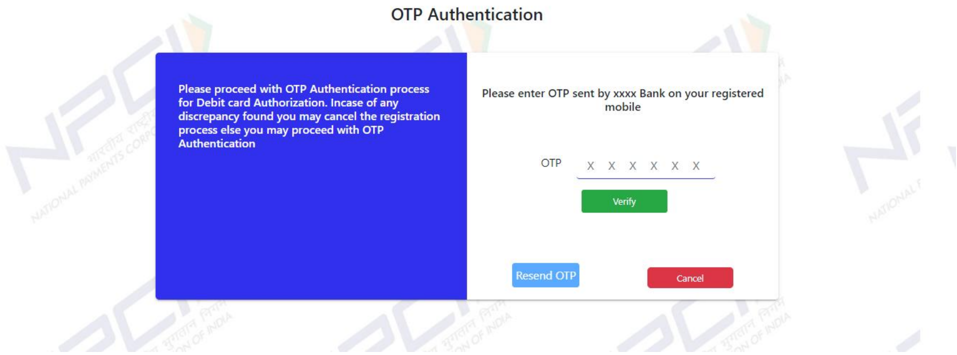
Below are the validation done related to the entered Debit Card Details

- Card Number should be 16 digit Numerical.
- Expiry Year and Month should be current month or future year month
- Expiry period cannot be greater than 10 years
- CVV will be 3 digit Numeric.
- PIN number is optional field and its bank’s specific
- CVV/PIN is mandatory (Banks can opt for CVV+PIN (or) either CVV/PIN

On entering the Debit Card Information user can click on Continue, to proceed with Debit Card Verification.

In case the user do not want to proceed further with authentication then he/she can click on Cancel. On clicking on Cancel, the transaction is cancelled, merchant response gets generated redirects to the Merchant response page. Once the User clicks on Continue, ONMAGS will construct the below JSON request and post to the Bank. Debit card validation will happen at Bank site

If Debit card validation fails, corresponding error message will be shown to the user. If the debit card validation gets passed at Bank site, then ONMAGS will redirect to the OTP verification page.



OTP Authentication

Please proceed with OTP Authentication process for Debit card Authorization. Incase of any discrepancy found you may cancel the registration process else you may proceed with OTP Authentication

Please enter OTP sent by xxxx Bank on your registered mobile

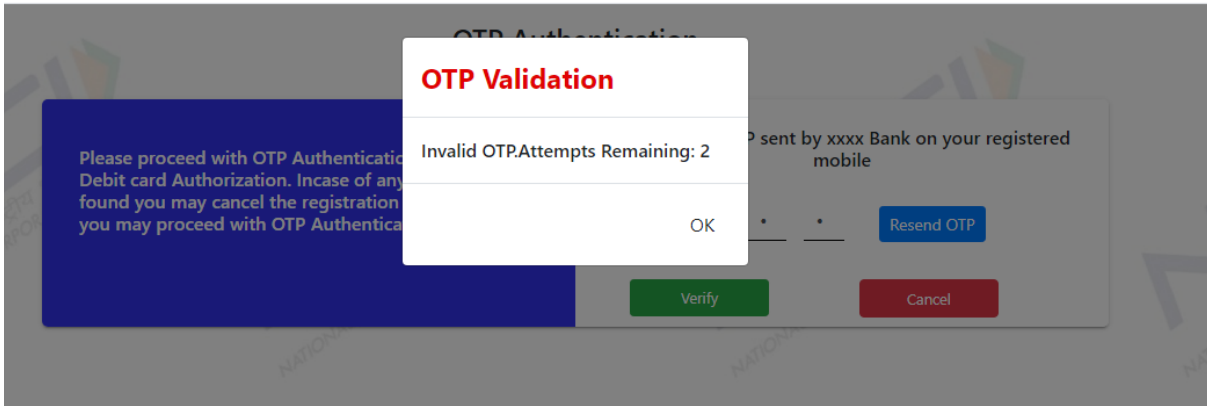
OTP X X X X X X

- OTP will be a 6 digit numeric number
- In case User did not receive OTP, there is an option to Resend OTP which the user can retry maximum of 3 times.

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

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.



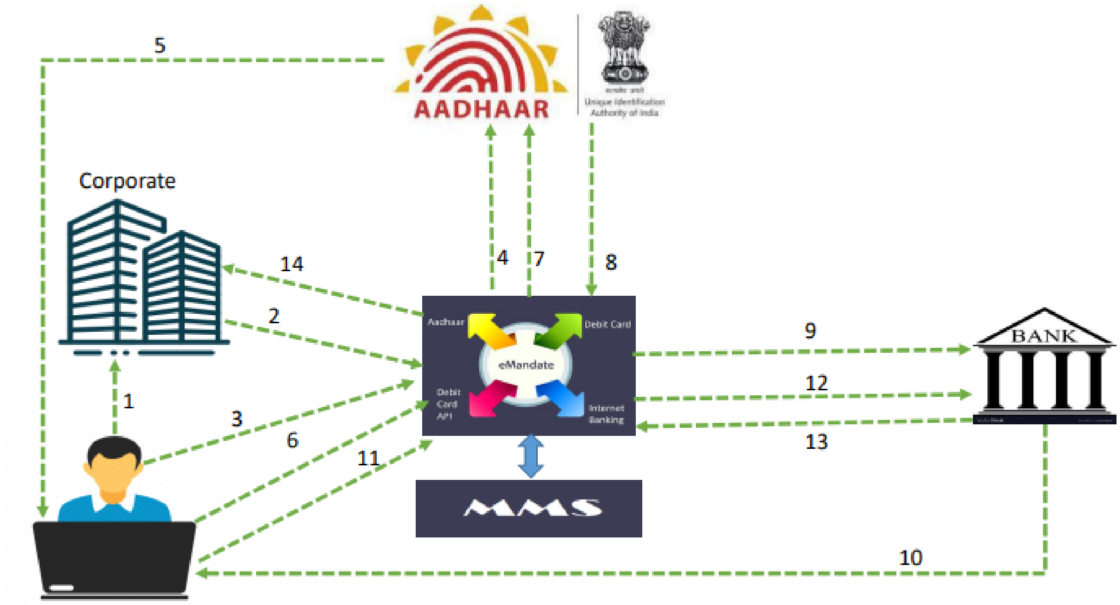
OTP Validation

Invalid OTP.Attempts Remaining: 2

Please proceed with OTP Authenticatic Debit card Authorization. Incase of any found you may cancel the registration you may proceed with OTP Authentica

P sent by xxxx Bank on your registered mobile

5.3.3 Aadhaar 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.

Step3: Customer enters Aadhaar Number along with required details.

Step4: ONMAGS Platform will forward that request to UIDAI for Customer Authentication via OTP generation

Step5: UIDAI will generate the OTP and send it to Customer's registered mobile number for Authentication

Step6: Customer will enter the OTP in the ONMAGS OTP page

Step7: ONMAGS Platform will forward that OTP to UIDAI for Verification

Step8: UIDAI sends response for OTP Verification. If the request is not authenticated by UIDAI then the flow ends here by showing the error message in Merchant Portal.

Step9: Once the customer is successfully authenticated, then the ONMAGS platform will send the mandate request to destination bank. If customer bank doesn't opt for additional OTP authentication then skip Step 10, Step 11 and Step 12.

Step10: After customer successfully authenticated by UIDAI, he/she will be landed on ONMAGS OTP page. ONMAGS will send an API request to customer's Bank to verify the customer details will generate the Bank OTP and send it to customer for Authentication.

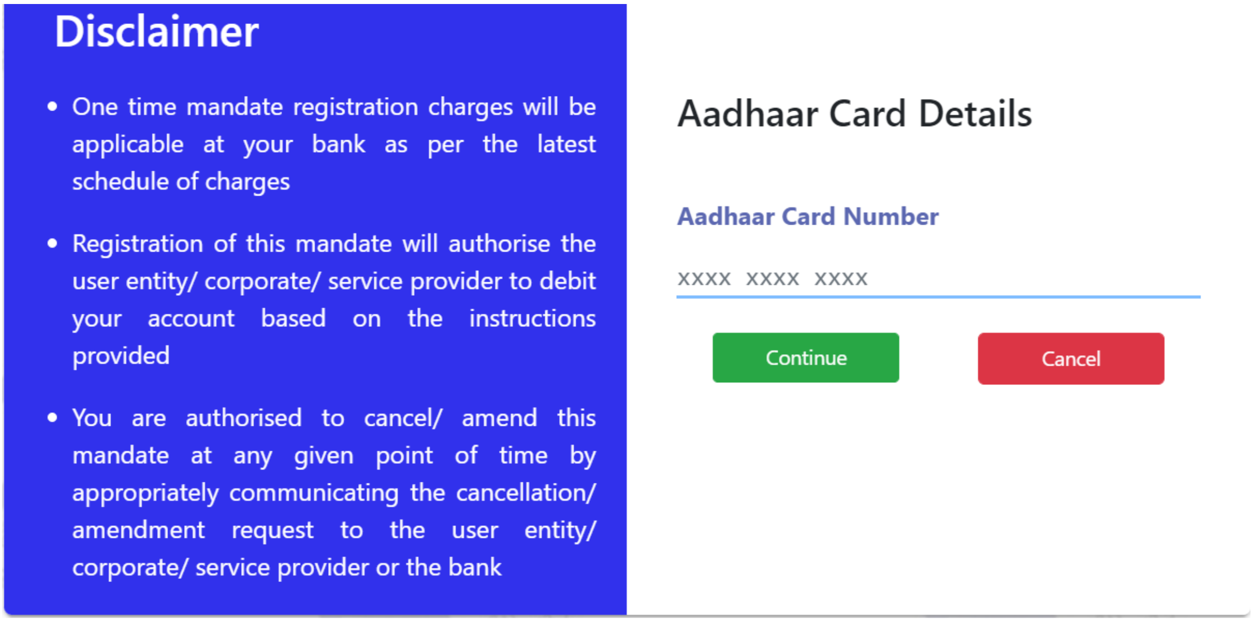
Step11: Customer will enter the Bank OTP in ONMAGS platform for Authentication.

Step12: ONMAGS platform will forward that OTP to destination bank for Verification.

Step13: 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.

Step 14: ONMAGS Platform in turn redirects the response to Merchant Web Page where customer can view the response

ONMAGS page where customer enters his/her Aadhar Number



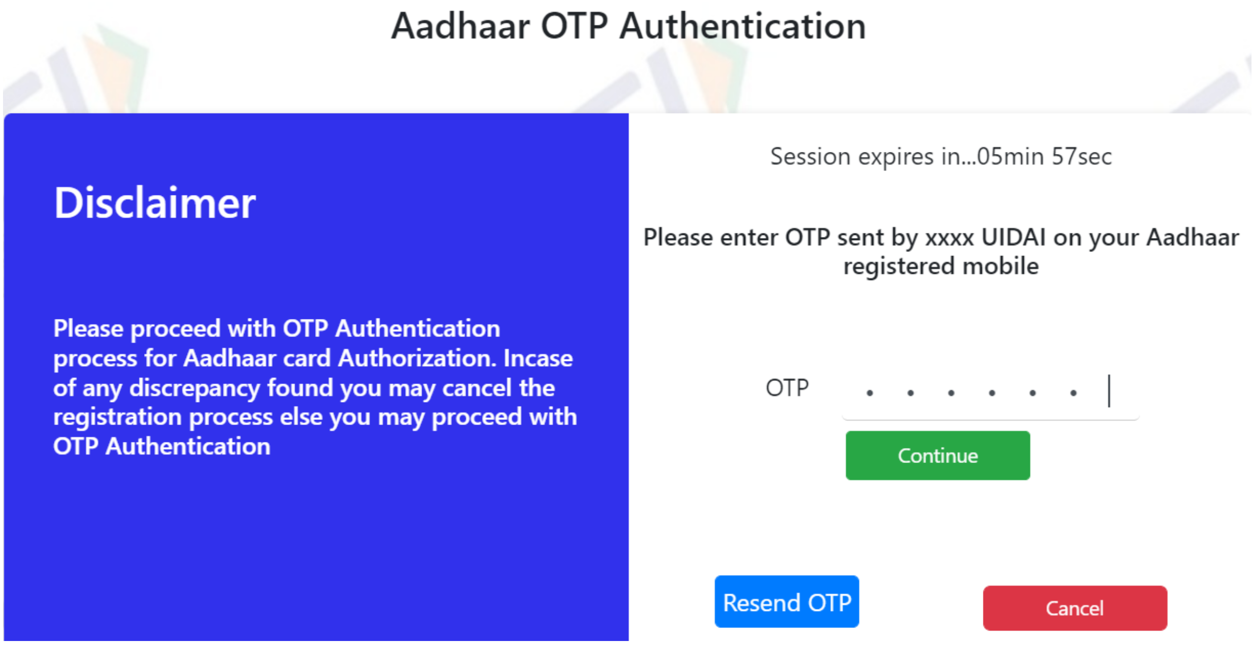
Disclaimer

- One time mandate registration charges will be applicable at your bank as per the latest schedule of charges
- Registration of this mandate will authorise the user entity/ corporate/ service provider to debit your account based on the instructions provided
- You are authorised to cancel/ amend this mandate at any given point of time by appropriately communicating the cancellation/ amendment request to the user entity/ corporate/ service provider or the bank

Aadhaar Card Details

Aadhaar Card Number: xxxx xxxx xxxx

Aadhaar OTP Authentication:



Aadhaar OTP Authentication

Disclaimer

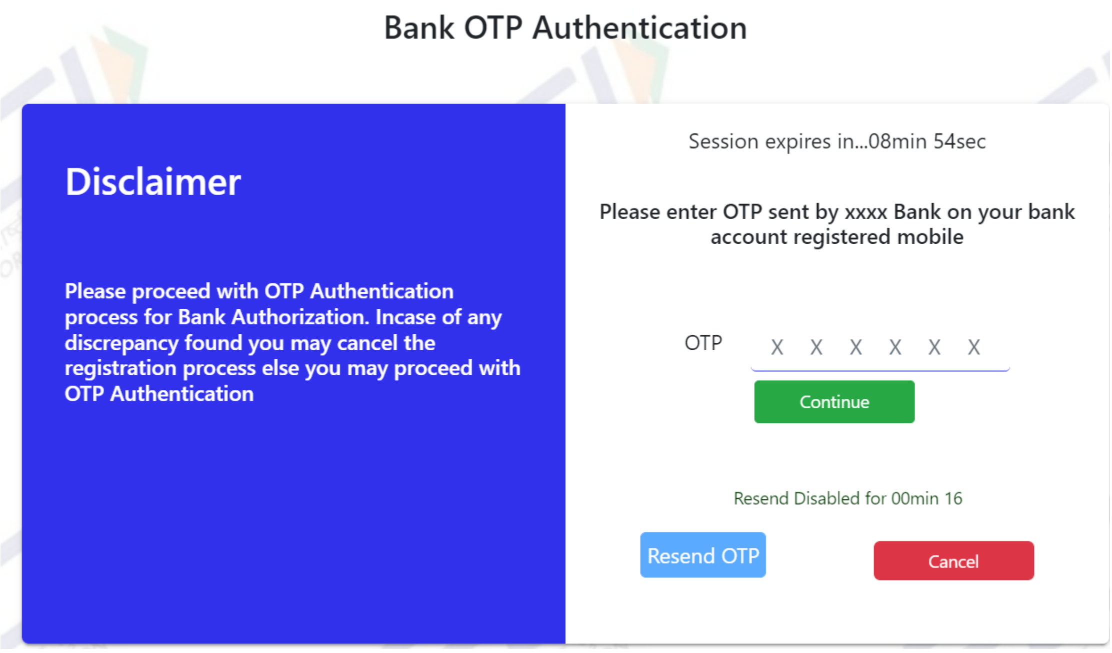
Please proceed with OTP Authentication process for Aadhaar card Authorization. Incase of any discrepancy found you may cancel the registration process else you may proceed with OTP Authentication

Session expires in...05min 57sec

Please enter OTP sent by xxxx UIDAI on your Aadhaar registered mobile

OTP: • • • • • •

Bank OTP Authentication:



Bank OTP Authentication

Disclaimer

Please proceed with OTP Authentication process for Bank Authorization. Incase of any discrepancy found you may cancel the registration process else you may proceed with OTP Authentication

Session expires in...08min 54sec

Please enter OTP sent by xxxx Bank on your bank account registered mobile

OTP X X X X X X

Resend Disabled for 00min 16

5.4 Signing and Encryption process

Below is the process for encryption & signing during the various flows.

- **Merchant to NPCI**
  - Encryption will be done using the public key of the certificate shared by NPCI.
  - Signing Using Private key certificate of the merchant
- **NPCI to Merchant**
  - Encryption will be done using the Public Key of the certificate shared by the Merchant.
  - Signing Using Private key certificate of NPCI

5.5 Encoding Guideline

The request XML & response XML posted to NPCI and received from NPCI should in encoded format. As part of encoding specific characters would be replaced by escaped character of those.

Symbol	Spelled	Escaped Character
‘	Single Quotes	&apos;
“	Double Quotes	&quot;
&	Ampersand	&amp;
<	Less Than	&lt;
>	Greater Than	&gt;

5.6 Handling of Time out / not reachable Scenarios

The following timeout verification is being maintained by NPCI while the mandate request posted to Bank for authentication

Flow	Auth Mode/Request	Timeout	Remarks
NPCI to Bank	**Old Net Banking/Debit Card**	30Min	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI. (merchant can use Status API to know the status of the request)
	New Debit Card – Submit Card Details	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	New Debit Card – OTP Verification	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at

			NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Aadhaar verification by UIDAI	90 Sec	No response from UIDAI for original request. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Aadhaar OTP Authentication	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Account verification by Bank	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Bank OTP Authentication	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code

Explained below are the action taken at NPCI ONMAGS layer for timeouts happening at various levels.

- **Merchant to NPCI**

  Scenario-1: User has not acted in the NPCI ONMAGS page within the timeout limit or has submitted the pager after the defined timeout.

  Action: The request will be auto closed as Failed at NPCI end after the specified duration. Merchant will not receive any communication from NPCI. Merchant needs to make use of the offline API’s available to understand the status of such requests.

- **NPCI to Merchant**

  Scenario-1: NPCI has not responded to Merchant within the timeout period.

  Action: Merchant needs to make use of the offline API’s available to understand the status of such requests.

  Scenario-2: Merchant is not reachable while posting response through browser.

Action: Merchant needs to make use of the offline API's available to understand the status of such requests.

6. Miscellaneous Features

6.1 API to get live destination banks for e-mandate

A Rest API would be made available which Corporates and Banks can access for getting the details on live destination banks.

The Rest API needs to be invoked as a Get Request. The response of the Rest API would be JSON Object of multiple arrays. An example output is given below:

```
{
  "liveBankList":[
    {
      "bankID":"SBIN",
      "bankName":"State Bank Of India",
      "ifsc":"SBIN0004343",
      "netBankStatus":"Active",
      "nbActiveFrom":"24-May-2017",
      "debitCardStatus":"InActive",
      "dcActiveFrom":"24-May-2017"
    },
    {
      "bankID":"HDFC",
      "bankName":"HDFC Bank LTD",
      "ifsc":"HDFC0012747",
      "netBankStatus":"Active",
      "nbActiveFrom":"24-May-2017",
      "debitCardStatus":"Active",
      "dcActiveFrom":"22-May-2017"
    }

  ]

}
```

For getting the live bank list the following API service has to be invoked:

Production:

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

UAT:

https://enachuat.npci.org.in:8086/apiservices/getLiveBankDtls

6.2 API to get Transaction Status for Merchant

For the purpose of getting the transaction status of a particular transaction or group of transactions for Merchant, 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:

```
{
  " mandateReqIDList ":[
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"000f0f29dc27f00000101b09c52b8e50037",
      "ReqInitDate":" 017-02-09"
    },
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"000f0f29dc27f00000101b09c52b8e50037",
      "ReqInitDate":" 2017-02-09"
    }
  ]
}
```

Note:

- MerchantID - Should match the <ID> tag present in <ReqInitPty> node of the Request XML from Merchant
- MndtReqId – This should match the value present in the tag <MndtReqId>
- ReqInitDate – Date format would be in “yyyy-mm-dd format”. This date should correspond to the date present in the tag “CreDtTm” of the Request XML.

Sample Output JSON:

```
{
  "tranStatus ":[
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"000f0f29dc27f00000101b09c52b8e50037",
      "ReqInitDate":" 017-02-09",
      "NpciRefMsgID":"000f0f29dc27f00000101b09c5227457f17",
      "MndtId":"xxxxxxxxxxxxxxxxxxxx",
      "Accptd":"false",
      "AccptRefNo":"tranid3432kkkeke",
      "ReasonCode":"343",
      "ReasonDesc":"Stale Account",
      "RejectBy":"Bank",
      "ErrorCode":"000",
      "ErrorDesc":"NA"
    },
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"000f0f29dc27f00000101b09c52b8e50037",
      "ReqInitDate":" 2017-02-09",
      "NpciRefMsgID":"NULL",
      "MndtId":"NULL",
      "Accptd":"NULL",
      "AccptRefNo":"NULL",
      "ReasonCode":"NULL",
      "ReasonDesc":"NULL",
      "RejectBy":"NULL",
      "ErrorCode":"453",
      "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/getTransStatusForMerchant

UAT:

https://enachuat.npci.org.in:8086/apiservices/getTransStatusForMerchant

6.3 API to Get Response posted to Merchant

This API is provided to get the response posted to Merchant against the NPCI Reference ID. The API accepts a single NPCI Reference ID / Mandate Request ID and provides the response posted to the merchant. The XML in the response would be either of Response XML or Error XML. There will be a limitation on the number of items posted per request. Currently the limit is set as 10.

Note: -

This service will be used by the Merchant.

Sample Request JSON format:

```
        {
  "getRespForNPCIRefID":[
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"000f0f29dc27f00000101b09c52b8e50037",
      "ReqInitDate":"2018-09-19",
      "NpciRefMsgID":"NULL"
    },
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"NULL",
      "ReqInitDate":"NULL",
      "NpciRefMsgID":"acdf0f29dc27f3456101b09c52b8e39004"
    }
  ]
}
```

Note:

- MerchantID - Should match the <ID> tag present in <ReqInitPty> node of the Request XML from Merchant
- MndtReqId – This should match the value present in the tag <MndtReqId>
- ReqInitDate – Date format would be in “yyyy-mm-dd format”. This date should correspond to the date present in the tag “CreDtTm” of the Request XML
- NpciRefMsgID – Will be the NpciRefMsgID generated by NPCI for the request.
- Either of NpciRefMsgID or (MerchantID , MndtReqId, ReqInitDate) combination is mandatory
- **Sample Response JSON format:**

Output JSON would be the same format provided to Merchant using browser to browser communication apart from two more columns Error code & Error Description.

```
        {
  "responseDtl":[
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"000f0f29dc27f00000101b09c52b8e50037",
      "ReqInitDate":"2018-09-19",
      "NpciRefMsgID":"NULL",
      "MndtId":"xxxxxxxxxxxxxxxxxxxx",
      "MandateRespDoc":"<Encrypted and Signed response XML>",
      "CheckSumVal":"<Check sum value of secure attributes>",
      "RespType":"<Will be either of ErrorXML / RespXML>",
      "ErrorCode":"000",
      "ErrorDesc":"NA"
    },
    {
      "MerchantID":"ABC22333",
      "MndtReqId":"NULL",
      "ReqInitDate":"NULL",
      "NpciRefMsgID":"acdf0f29dc27f3456101b09c52b8e39004",
      "MndtId":"xxxxxxxxxxxxxxxxxxxx",
      "MandateRespDoc":"NULL",
      "CheckSumVal":"NULL",
      "RespType":"NULL",
      "ErrorCode":"455",
      "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/respPostedToMerchant

UAT:

https://enachuat.npci.org.in:8086/apiservices/respPostedToMerchant

6.4 Restricting Duplicate Transactions

ONMAGS has introduced a feature to identify & to restrict the duplicate mandate requests. This will help the eco system by reducing the processing efforts of additional load. These duplicate requests will be identified & restricted at the initial level itself, before the request even sent to the destination bank. Details are as follows.

Identification of Duplicate request:

A request will be identified as a ‘Duplicate Request’, If the customer has raised a mandate request already on the given day and same customer sends another request with same details such as

- **Corporate Utility Code**
- **Corporate Sponsor Bank**
- **Category Code**
- **Customer Account Number**
- **Max Amount/Fixed Amount**
- **Destination Bank**

on the same day, this second request will be treated as a duplicate request and the fate of this duplicate request will be decided based on the status of its original request.

The action taken on duplicate request will be as follows.

S. No	Status of Original Request	Action taken on Duplicate Request	Customer Action
1	Successful	No restriction, duplicate request will be allowed	Customer can initiate duplicate request
2	Rejected by Bank	Request will be rejected by NPCI with error code 608. (error description will consist the reject reason of the original request) **Note**: Based on the reason code with which the original request rejected, customer will be either restricted on their first duplicate request or will be allowed for 2 or 3 retries. Reason codes allowed for representation are mentioned in document attached in Appendix section.	Customer should wait for 24 Hrs. and retry on the next day.
3	Rejected by NPCI	No restriction, duplicate request will be allowed	Customer can initiate duplicate request
4	Pending	If the duplicate request initiated within 5min of its original request, then the request will have rejected by NPCI with error code 607 (Previous Request in progress).	Customer should wait for 5Min and retry.
		If the duplicate request initiated after 5min of its original request, no restriction. Duplicate request will be allowed.	Customer can re-initiate the transaction

Note: Merchant will receive the duplicate rejection message in Error XML format.

7. About NPCI Gateway

NPCI Gateway Page will be mobile responsive Page. The page has been built to be lightweight. The application has gone through the required security testing as mandated by RBI.

7. Appendix

7.1 Request & Response XML Specification for Merchants

Validation_Sheet Merchant

Validation Sheet.xlsx

7.2 Sample XML Formats and Schemas

Request Response Files_V7.zip

7.3 Reason & Error Codes

merchant error codes

Error_codes.xls

Reason Code List_Duplicate Reques

Bank_Rej_Reason_codes.xls

DOCUMENT RELEASE NOTICE

Document Details

Name	Version No.	Date	Description
Bank Specification Document	Draft	24-02-2017	Provides technical & operation specification for Banks to develop compatible application at their end for communicating with the Mandate Authorization application
NPCI Mandate Authorization Specification for Banks	1.0	01-03-2017	Updated for Debit Card
NPCI Mandate Authorization Specification for Banks	1.1	22-03-2017	Covered the specification for Signing, Check Sum & Encryption. XML Specification, XSD & XML Samples attached as zip
NPCI Mandate Authorization Specification for Banks	2.0	27-03-2017	Updated for Error Scenarios, HTTP Status codes.
NPCI Mandate Authorization Specification for Banks	3.0	26-05-2017	API to get live destination banks for e-mandate Separate URL's for Net banking & Debit Card Corporate mapping to the destination banks
NPCI Mandate Authorization Specification for Banks	3.1	08-06-2017	Updated the process flow to include bank selection in the merchant page.
NPCI Mandate Authorization Specification for Banks	3.2	24-06-2017	Addition of Dbtr tag in Request XML.

			Changes in Error Response XML’s, Error Codes & Failure Scenarios (In Appendix)
NPCI Mandate Authorization Specification for Banks	3.3	03-Jul-2017	Error Codes & Failure Scenarios Sheet Updated. Encryption of Debtor field instead of Creditor. Changes in Server to Server communication specification.
NPCI Mandate Authorization Specification for Banks	3.4	14-Jul-2017	Changes in Request & Response XML formats and Error XML format.
NPCI Mandate Authorization Specification for Banks	3.5	07-Aug-2017	Handling of Timeout Scenario Added
NPCI Mandate Authorization Specification for Banks	3.5	20-Dec-2017	Encryption methodology updated Updates to Offline API’s Error Codes Updated
NPCI Mandate Authorization Specification for Banks	3.6	18-Sep-2018	AuthMode added as additional Parameter from Merchant. Flow changes based on this parameter.
NPCI Mandate Authorization Specification for Banks	3.7	15-APR-2019	Change in API “Posting list of Open Transactions to Bank”
NPCI Mandate Authorization Specification for Banks	3.8	17-May-2019	Changes in Error XML Structure from Bank to NPCI and from NPCI to Merchant (Appendix 9.1)

			Changes in lengths and data types of XML elements in Merchant Request. Changes in Error Response from Bank Change in live bank list api
NPCI Mandate Authorization Specification for Banks	4.0	12-DEC-2019	Changes in Merchant request XML, Bank request XML & Merchant response XML. Encryption of additional fields Encryption of Request XML and Response XML Additional parameter in the form post for Merchant
NPCI Mandate Authorization Specification for Banks	4.1	29-JUL-2020	Introduction of New Debit Card Flow BANKID & AUTHMODE mandatory in merchant request
NPCI Mandate Authorization Specification for Banks	4.2	29-DEC-2021	Introduction of Aadhaar flow

This document and any revised pages are subject to document control. Please keep them up-to-date using the release notices from the distributor of the document.

Table of Contents

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.

1.1 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

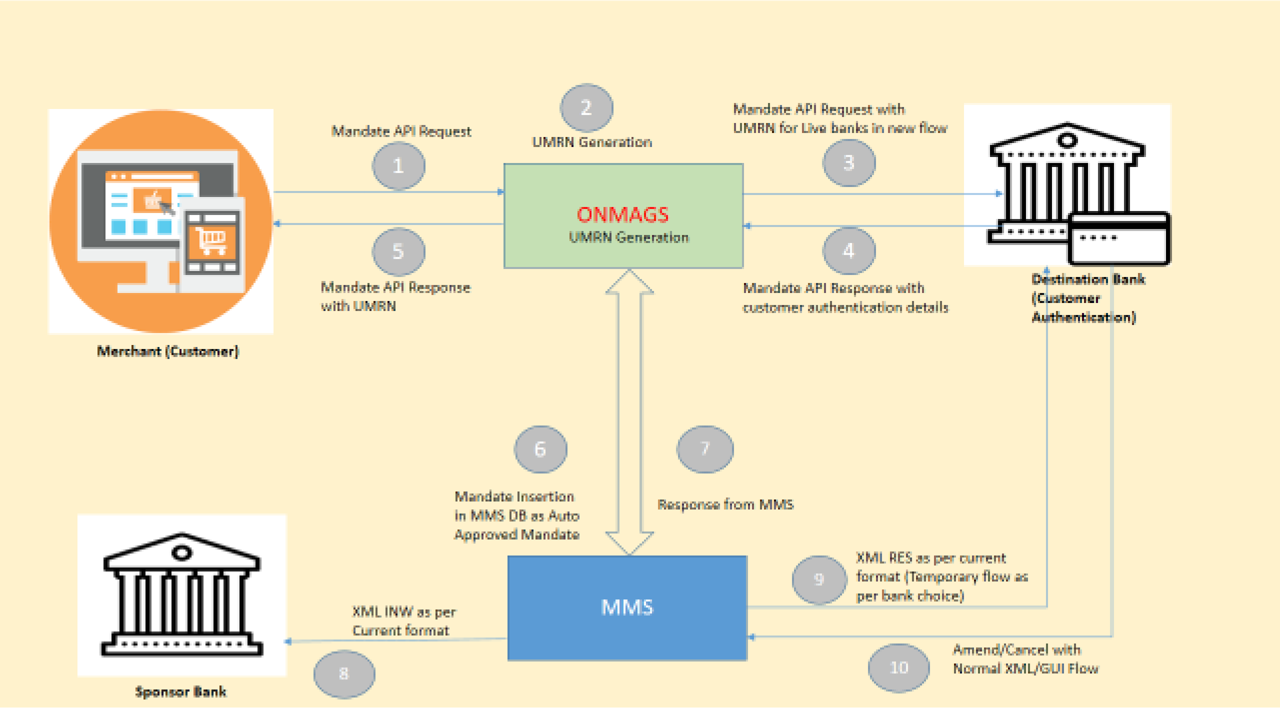
2.1 Registration with NPCI

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

2.2 Mandate Approval function flow (for Net Banking & Debit Card authentication modes)

The mandate approval flow is initiated from the Merchant end, request validated at NPCI end and forwarded to the Bank for authorization. The confirmation provided back by the Destination Bank is relayed back to the merchant.

Mandates created through ONMAGS will be auto registered in MMS. The overall flow and the integration between ONMAGS and MMS systems is explained by the below diagram.



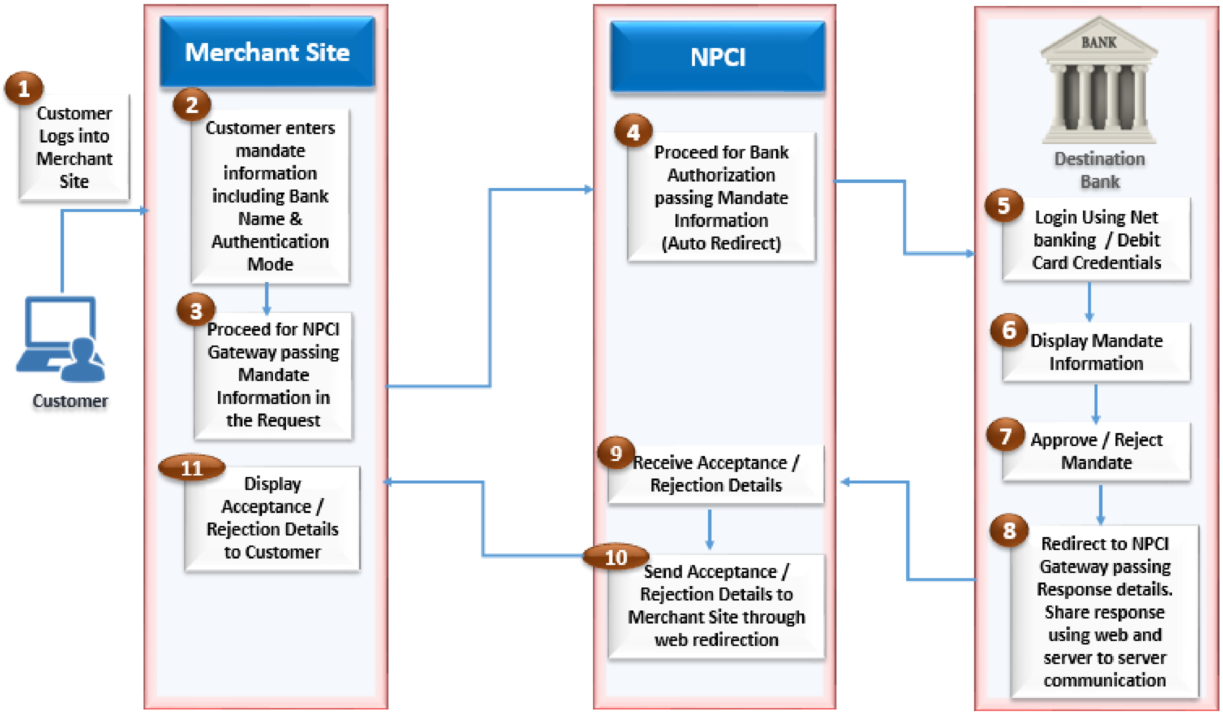
The process flow is mentioned in the next section.

Note:

From version 4.1 BankID & AuthMode are mandatory in the merchant request

2.2.1 End to End Process Flow (for Net Banking & Debit Card Authentication Modes)

The below diagram illustrates the functional flow of mandate authorization when Bank ID & Authentication Mode are passed from Merchant. This will be the default flow from version 4.1.


Merchant Site
NPCI
BANK
Destination Bank
1 Customer Logs into Merchant Site
Customer
2 Customer enters mandate information including Bank Name & Authentication Mode
3 Proceed for NPCI Gateway passing Mandate Information in the Request
4 Proceed for Bank Authorization passing Mandate Information (Auto Redirect)
5 Login Using Net banking / Debit Card Credentials
6 Display Mandate Information
7 Approve / Reject Mandate
8 Redirect to NPCI Gateway passing Response details. Share response using web and server to server communication
9 Receive Acceptance / Rejection Details
10 Send Acceptance / Rejection Details to Merchant Site through web redirection
11 Display Acceptance / Rejection Details to Customer

Note:-

In case of new debit card/Aadhaar flow there will not be any redirection to Bank. The debit card/Aadhaar authentication will happen in NPCI side itself. For this ONMAGS will interact with Banks through API calls for validating the mandate and debit card/Aadhaar information. (Detailed flow explained in section 4.2.3)

- Customer logins to the merchant site where he/she would be shown the mandate Information
- Specific details of the mandate along with deduction details needs to be shown.
- Customer can proceed with accepting the mandate if he/she finds the information displayed is correct (Customer needs to enter the Bank account number before proceeding)
- Merchant site needs to provide the option for selecting Bank & Authentication Mode (NetBanking, Debit Card, Aadhaar Card).
- Customer would be redirected to NPCI ONMAGS interface.
- NPCI Interface would show an intermittent page while processing happens in the back ground.

- If the validation is successful, then NPCI will auto redirect to Bank's authentication page based on the Bank ID & Authentication Mode selected by the end user in the merchant site.

- If the validation fails, then NPCI will redirect back to the Merchant Site posting the Error XML response.

- Bank will display the authentication Page based on the Auth mode selected by the user. (

- In the Banks page customer will authenticate either using the user's net banking credential or Debit card credentials based on the authentication mode user had selected in the Merchant page.

- Bank need to validate whether the Account Number passed in the request XML matches the Account Number through which the customer has authenticated the login.

- Once verified Bank Page will display the summary of the mandate and provide option for accepting or rejecting the mandate

- Once the customer has selected either of Approve / Reject link he would be redirected back to NPCI ONMAGS interface

- The NPCI ONMAGS interface will auto redirect to the merchant site

- Merchant site will display the status of Mandate Approval

2.3 Interface Layer

Necessary ports need to be opened between NPCI servers & Bank servers. Also required certificates needs to be installed at NPCI & Bank Site servers.

For API flow (Direct Debit card/Aadhaar) authentication below details are required.

1. NPCI to Bank connectivity with specified port
2. Bank SSL certificate (FQDNS is preferred)
3. URL's (mandate validation, verify OTP and resend OTP)

3. Specification Format for Request & Response

Appendix 9.1

Lists the XML file format for the request & response.

The specification for below request / response are listed in the document.

The data format would be XML. Schema structure and sample XML's can he found in Appendix 9.3.

- NPCI Mandate Request to Bank

NPCI ONMAGS will send the request to bank in the specified format

- Response from Bank to NPCI

Destination Bank will use this format for sending response bank to NPCI ONMAGS.

4. Technical Integration Specification

The below section lists a few of the technical requirements for the implementation.

4.1 Forward Flow specification from NPCI to Bank

This flow applies to Net Banking mode of authentication or for the Old Debit Card flow authentication.

Below are the steps done for securing the content of the Request data posted to the Bank from NPCI

1. The request XML to bank with all the tags present will be in the below format: -

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="http://npci.org/ONMAGS/schema">
   <MndtAuthReq>
        <GrpHdr>
             <NPCI_RefMsgId></NPCI_RefMsgId>
             <CreDtTm></CreDtTm>
             <ReqInitPty>
                  <Info>
                       <Id></Id>
                       <CatCode></CatCode>
                       <UtilCode></UtilCode>
                       <CatDesc></CatDesc>
                       <Name></Name>
                       <Spn_Bnk_Nm></Spn_Bnk_Nm>
                  </Info>
             </ReqInitPty>
        </GrpHdr>
        <Mndt>
             <MndtReqId></MndtReqId>
             <MndtId>UMRN</MndtId>
             <Mndt_Type></Mndt_Type>
             <Schm_Nm><Schm_Nm>
             <Ocrncs>
                  <SeqTp></SeqTp>
                  <Frqcy></Frqcy>
                  <FrstColltnDt></FrstColltnDt>
                  <FnlColltnDt></FnlColltnDt>
             </Ocrncs>
             <ColltnAmt Ccy="INR"></ColltnAmt>
             <MaxAmt Ccy="INR"></MaxAmt>
             <Dbtr>
                  <Nm></Nm>
                  <AccNo></AccNo>
                  <Acct_Type></Acct_Type>
                  <Cons_Ref_No></Cons_Ref_No>
                  <Phone></Phone>
                  <Mobile></Mobile>
                  <Email></Email>
```

```
                   <Pan></Pan>
              </Dbtr>
              <CrAccDtl>
                   <Nm></Nm>
                   <AccNo></AccNo>
                   <MmbId></MmbId>
              </CrAccDtl>
        </Mndt>
   </MndtAuthReq>
</Document>
```

2. Generating checksum for the secure information in the XML

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

   - Debtor Account Number
   - First Collection Date
   - Final Collection Date
   - Collection Amount
   - Max Amount

   The above attributes need to be concatenated with "|" symbol appended as the delimiter. The order of the attributes needs to be as mentioned above. In case any of the attribute is null then during concatenation the particular attribute will be replaced by an empty string.

   Note:
   The attributes to be concatenated might be changed at later point of time. Please refer the latest version of the document for any revision on the attributes that needs to be marked for encryption.

   Generate checksum on the concatenated values. We will use SHA-2 as the hash function.

3. Replace the secure information in the XML with the encrypted text. Below are the attributes which will be encrypted in the request XML

   - Debtor Account Number
   - First Collection Date
   - Final Collection Date
   - Collection Amount
   - Max Amount
   - Phone
   - Mobile
   - Email
   - Pan

   The attributes mentioned above needs to be encrypted individually and placed in the respective XML tags. We will use the below methodology for encryption of secure information.

Encryption Methodology – Asymmetric

Hashing Algorithm – SHA256

Cryptography – RSA/ECB/OAEPWithSHA-256AndMGF1Padding 2048 bits.

Encryption will be done using the Public Key of the certificate shared by Bank.

4. Signing of the Request XML

   The request XML got from Step-2 will be signed using the Private Key certificate of NPCI.

NPCI will send the below data as MIME content to Merchant with type as “application/x-www-form-urlencoded” in the request body.

Key	Value
MandateReqDoc	Output of the Step-3
CheckSumVal	Encrypted Output of Step-1

4.1.1 Encoding of Request XML for Banks

The request XML from NPCI to Bank will be encoded to prevent any malicious attack. Banks will need to accept the encoded xml content at their end then decode it to get the original content.

The encoded request XML will look as below: -

```
&lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?&gt;
&lt;Document xmlns=&quot;http://npci.org/ONMAGS/schema&quot;&gt;
   &lt;MndtAuthReq&gt;
         &lt;GrpHdr&gt;
               &lt;NPCI_RefMsgId&gt;&lt;/NPCI_RefMsgId&gt;
               &lt;CreDtTm&gt;&lt;/CreDtTm&gt;
               &lt;ReqInitPty&gt;
                     &lt;Info&gt;
                          &lt;Id&gt;&lt;/Id&gt;
                          &lt;CatCode&gt;&lt;/CatCode&gt;
                          &lt;UtilCode&gt;&lt;/UtilCode&gt;
                          &lt;CatDesc&gt;&lt;/CatDesc&gt;
```

```
                         &lt;Name&gt;&lt;/Name&gt;
                         &lt;Spn_Bnk_Nm&gt;&lt;/Spn_Bnk_Nm&gt;
                   &lt;/Info&gt;
              &lt;/ReqInitPty&gt;
        &lt;/GrpHdr&gt;
        &lt;Mndt&gt;
              &lt;MndtReqId&gt;&lt;/MndtReqId&gt;
              &lt;MndtId&gt;UMRN&lt;/MndtId&gt;
              &lt;Mndt_Type&gt;&lt;/Mndt_Type&gt;
              &lt;Schm_Nm&gt;&lt;Schm_Nm&gt;
              &lt;Ocrncs&gt;
                   &lt;SeqTp&gt;&lt;/SeqTp&gt;
                   &lt;Frqcy&gt;&lt;/Frqcy&gt;
                   &lt;FrstColltnDt&gt;&lt;/FrstColltnDt&gt;
                   &lt;FnlColltnDt&gt;&lt;/FnlColltnDt&gt;
              &lt;/Ocrncs&gt;
              &lt;ColltnAmt
Ccy=&quot;INR&quot;&gt;&lt;/ColltnAmt&gt;
              &lt;MaxAmt Ccy=&quot;INR&quot;&gt;&lt;/MaxAmt&gt;
              &lt;Dbtr&gt;
                   &lt;Nm&gt;&lt;/Nm&gt;
                   &lt;AccNo&gt;&lt;/AccNo&gt;
                   &lt;Acct_Type&gt;&lt;/Acct_Type&gt;
                   &lt;Cons_Ref_No&gt;&lt;/Cons_Ref_No&gt;
                   &lt;Phone&gt;&lt;/Phone&gt;
                   &lt;Mobile&gt;&lt;/Mobile&gt;
                   &lt;Email&gt;&lt;/Email&gt;
                   &lt;Pan&gt;&lt;/Pan&gt;
              &lt;/Dbtr&gt;
              &lt;CrAccDtl&gt;
                   &lt;Nm&gt;&lt;/Nm&gt;
                   &lt;AccNo&gt;&lt;/AccNo&gt;
                   &lt;MmbId&gt;&lt;/MmbId&gt;
              &lt;/CrAccDtl&gt;
        &lt;/Mndt&gt;
   &lt;/MndtAuthReq&gt;
&lt;/Document&gt;
```

4.2 Bank Site Integration Requirements

4.2.1 Net Banking Flow

In case of Netbanking or if the Bank has opted for the old debit card flow, then NPCI ONMAGS would redirect to Bank Page. The URL for redirection for Net banking should be made available to NPCI by the banks. NPCI will pass the XML content mentioned in the sheet (“NPCI Mandate Request to Bank”) & CheckSumVal as part of the request.

The request body will contain the following key-value pair.

Key	Value
MandateReqDoc	Encrypted and Signed XML
CheckSumVal	Encrypted Checksum Hash value

Specifics on Signing, Encryption and Checksum are mentioned in the section 4.2.1

Bank site should unsign the XML using the public key of NPCI and then decrypt the key fields using the private key of the Bank. Checksum should be decrypted using the private key of the Bank. In case of any errors during unsigning, decryption or checksum validation, Bank needs to construct the Error response in the format "ErrorXML Resp from Bank to NPCI".

Below are the validations done at Bank layer for the request received from NPCI. For more details refer to sheet "NPCI Mandate Request to Bank" in the excel "NPCI Mandate Authorization Specification for Banks.xlsx" available in the Appendix Section.

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 7 digit OLD ICS or 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	35	UMRN
Mndt_Type	Mandate Type	Alpha Numeric	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
FrstColltnDt	Date of First Collection. Mandatory Field. This field is in ISODate Format	Alpha Numeric	16	
FnlColltnDt	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.
ColltnAmt	Either of ColltnAmt 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 40 digit	Alpha Numeric	40	
Debtor AccNo	Customer Account Number should be maximum of 35 digit.	Alpha Numeric	35	
Acct_Type	Debtor Account Type	Alpha Numeric	35	Should be either of SAVINGS or CURRENT
Cons_Ref_No	Consumer Reference Number	Alpha Numeric	35	
Phone	Phone Number of the Customer	Alpha Numeric	16	Should be given in the format +91-xxx-xxxxxxxx. +91- is mandatory.
Mobile	Mobile Number of the Customer	Alpha Numeric	14	Should be given in the format +91-xxxxxxxxxx. +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	10	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

End user would enter his/her net banking credentials information in the authentication page of the bank. An SMS OTP validation also has to be done as second level authentication.

Upon making a successful login bank should first validate whether the bank account number passed in the request XML matches the bank account number of the authenticated end user. If the bank account number does not match the customer would not be allowed to proceed further. Appropriate error message needs to be displayed to the customer and a link provided to return back to the merchant site.

If the customer is not able to make a successful login after predetermined login attempts the Bank has to redirect back to the NPCI ONMAGS layer. The reject reason will be "Invalid Login Credentials".

If the account number matches, then the customer needs to be shown a form which displays specific details of the mandate and a "Terms & Policy" section displaying terms and policies of the bank. A confirmation check box needs to be provided for end user for agreeing to the displayed information.

The below information needs to be mandatorily displayed to the User at the Bank end:

- Mandate request Initiate Party's Category Description ("CatDesc")
- Name of Initiator
- Collection Amount
- Max Amount
- Recurring Frequency
- First Collection Date
- Final Collection Date

User has to be provided links for either accepting the mandate or Rejecting the mandate. On selection of either of the option the user would be redirected to the NPCI ONMAGS interface. The response should contain the XML mentioned in the sheet "Response from Bank to NPCI". The element <AccptncRslt> will contain the result of the approval status of the mandate. The URL for redirection to NPCI ONMAGS interface would be shared by NPCI.

The response body will contain the following key-value pair. Bank will send the below data as MIME content to NPCI with type as “application/x-www-form-urlencoded”.

Key	Value
BankID	Participant ID of the Bank in NACH
MandateRespDoc	Encrypted and Signed XML
CheckSumVal	Encrypted Checksum Hash value
RespType	Will be either of ErrorXML / RespXML

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

1. Generating checksum for the secure information in the XML

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

   a) Accptd
   b) AccptRefNo
   c) ReasonCode
   d) ReasonDesc
   e) RejectBy

   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 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.

2. Replace the secure information in the XML with the encrypted text.

   The attributes mentioned above needs to be encrypted individually and placed in the respective XML tags. Encryption should be done using the public key of the certificate which NPCI shares.

   We will use the below methodology for encryption of secure information.

   Encryption Methodology – Asymmetric

Hashing Algorithm – SHA256

Cryptography – RSA/ECB/OAEPWithSHA-256AndMGF1Padding 2048 bits

Encryption needs to be done using the Public Key of the certificate shared by NPCI.

3. Signing of the Response XML

The response XML got from Step-2 has to be signed using the Private Key certificate of the Bank.

The below are the validation done at NPCI ONMAGS layer for the response received from Bank. For more details refer to sheet "Response from Bank to NPCI" in the excel "NPCI Mandate Authorization Specification for Banks" available in the Appendix Section.

Element Name	Validation	Length	Remarks
Xmlns	Namespace tag. This is mandatory tag. Value cannot be empty. Namespace value should be "http://npci.org/ONMAGS/schema"		
MsgId	This is a reference generated by the bank to identify the response message. Should be unique for the day for a Bank	35	
GrpHdr - CreDtTm	Should be in ISO Date time format. E.g.2017-02-09T15:11:39	25	
ReqInitPty	Request Initiating Party ID. This will refer to the Bank Short Code	18	
MndtReqId	Mandate Request ID should be same as the MndtReqId send in the original request to Bank	35	
NPCI_RefMsgId	Message ID for NPCI Reference in the original request. Should be same as the NPCI_RefMsgId send in the original request to Bank	35	
OrgnlMsgInf - CreDtTm	Creation Date Time send in the original request to Bank	18	
MsgNmId	Both the tag & value are optional		
Accptd	Mandatory. Allowed values are true / false	5	Indicates whether the mandate request was accepted or rejected.
AccptRefNo	Will be non-empty if accptd is true. Should be unique for the Bank. If accptd is false empty value can be provided.	34	Accepted Reference Number.
ReasonCode	Mandatory if <Accptd> value is false. Reason code should be as per master provided by NPCI.	5	If acceptance is false, reason code of rejection is entered here. If

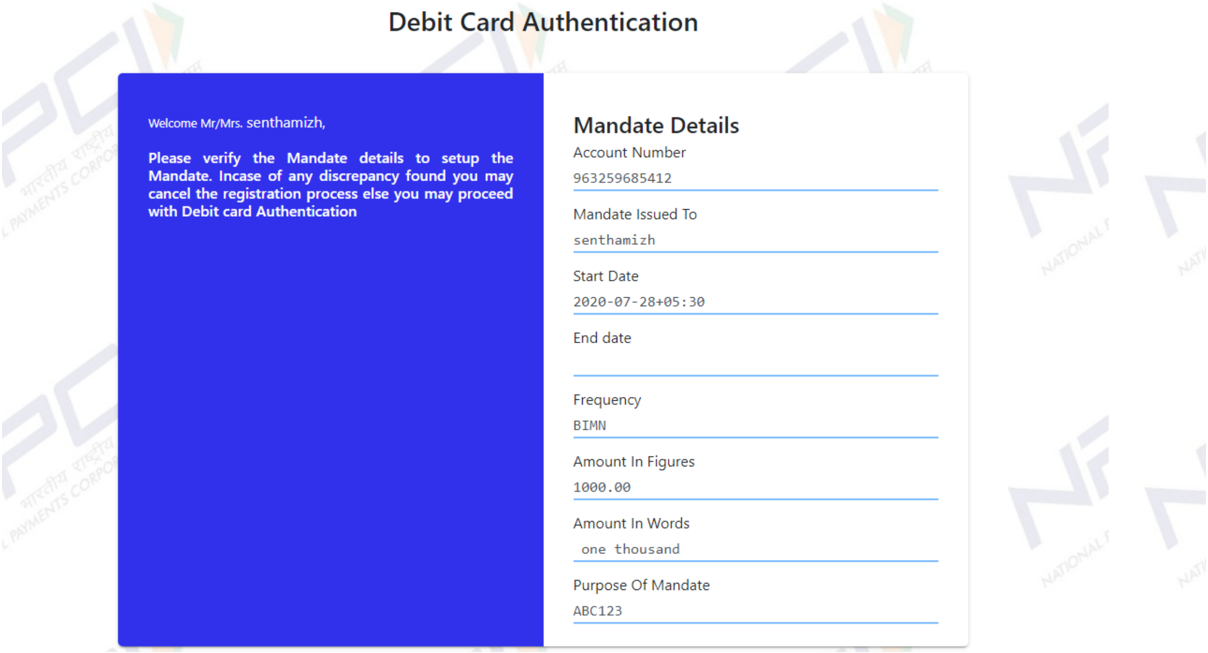
			acceptance is true then this value would be “N/A”.
ReasonDesc	Mandatory. Reason Description should match the Reason Code specified by NPCI.	50	If acceptance is false, reason description of rejection is entered here. If acceptance is true then this value would be “N/A”.
RejectBy	Mandatory. Should be either of “BANK” or “USER” or “N/A”	10	If acceptance is true then this value would be “N/A”.
IFSC	Mandatory if <Accptd> tag value is true, IFSC of the destination bank	11	

4.2.2 New Debit Card Flow

In case Bank has opted for the new Debit Card Flow, then from the merchant site, the user will be landing on the NPCI’s ONMAGS Debit Card authentication page.

Debit Card Information will be accepted in ONMAGS page itself and validated with Bank through server to server call. The steps in this flow is described below:

The mandate information passed by the merchant will be displayed in the top portion of the page.



Debit Card Authentication

Welcome Mr/Mrs. senthamizh,

Please verify the Mandate details to setup the Mandate. Incase of any discrepancy found you may cancel the registration process else you may proceed with Debit card Authentication

Mandate Details

Account Number
963259685412

Mandate Issued To
senthamizh

Start Date
2020-07-28+05:30

End date

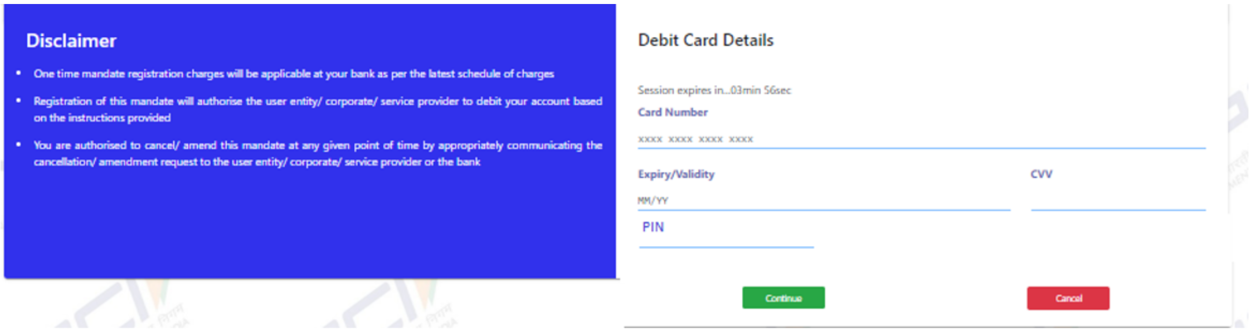
Frequency
BIMN

Amount In Figures
1000.00

Amount In Words
one thousand

Purpose Of Mandate
ABC123

User needs to verify the mandate information displayed in the Mandate Details section. Once mandate information are verified by the user he/she can proceed with entering the Debit Card information in the lower section of the page.



Disclaimer

- One time mandate registration charges will be applicable at your bank as per the latest schedule of charges
- Registration of this mandate will authorise the user entity/ corporate/ service provider to debit your account based on the instructions provided
- You are authorised to cancel/ amend this mandate at any given point of time by appropriately communicating the cancellation/ amendment request to the user entity/ corporate/ service provider or the bank

Debit Card Details

Session expires in...03min 56sec

Card Number: xxxx xxxx xxxx xxxx

Expiry/Validity: MM/YY

CVV

PIN

Below are the validation done related to the entered Debit Card Details

- Card Number should be 16 digit Numerical.
- Expiry Year and Month should be current month or future year month.
- Expiry period cannot be greater than 10 years.
- CVV should be 3 digit Numeric.
- PIN number is optional field and its bank's specific
- CVV/PIN is mandatory (Banks can opt for CVV+PIN (or) either CVV/PIN

On entering the Debit Card Information user can click on Continue, to proceed with Debit Card Verification.

In case the user does not want to proceed further with authentication then he/she can click on Cancel. On clicking on Cancel, the transaction will be cancelled, merchant response gets generated redirects to the Merchant response page.

4.2.2.1 Request Information to Bank

Once the User clicks on Continue, ONMAGS will construct the below JSON request and post to the Bank. Both the Mandate Details and the Card Information will be passed in the request. The request will be made as an API call to the bank and will happen as a server to server call. The response to the API call has to be provided in a synchronous manner by the bank.

JSON Request with Mandate and Card information

```
{
   "mandateAuthDtls": {
      "transactionID": "<Transaction ID>",
      "mandateRequestDtl": {
         "MandateReqDoc": "<Encrypted and Signed response XML>",
         "CheckSumVal": "<Check sum value of secure attributes>"
      },
      "cardInfo": {
         "cardNo": "<Encrypted Card Number>",
          "expiry": "<Encrypted expiry Date>",
          "cvv": "<Encrypted CVV>",
           "pin": "<Encrypted pin>"
       }
    }
    }
```

Note:

"**pin**" Key value is optional field, bank needs to decrypt if the pin value is present ,else no need to validate.

- MandateReqDoc XML will be encoded and sent.
- For encryption the existing logic and keys will be used (i.e, NPCI will do the encryption using the Public Key provided by the bank and Bank will do the decryption using their private key.

Bank needs to first verify the mandate request details and then the card details and provide the response in any of the below formats.

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 card details if sending failure response (because of request XML validation failure at bank end).

```
{⊟
   "mandateVerifyDtls": {⊟
      "transactionID": "<Transaction ID>",
      "mandateValidation": "failure",
      "cardValidation": "none",
      "mandateRejectDtl": {⊟
         "ErrorCode": "<Error Code>",
         "ErrorDesc": "<Error Description>"
      }
   }

}
```

Note:-

Attribute values mandateValidation, cardValidation, ErrorCode & 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. Card details need not be validated in such a scenario.

```
{
   "mandateVerifyDtls": {
      "transactionID": "<Transaction ID>",
      "mandateValidation": "failure",
      "cardValidation": "none",
      "mandateRejectDtl": {
         "ReasonCode": "<Error Code>",
         "ReasonDesc": "<Error Description>"
      }
   }
}
```

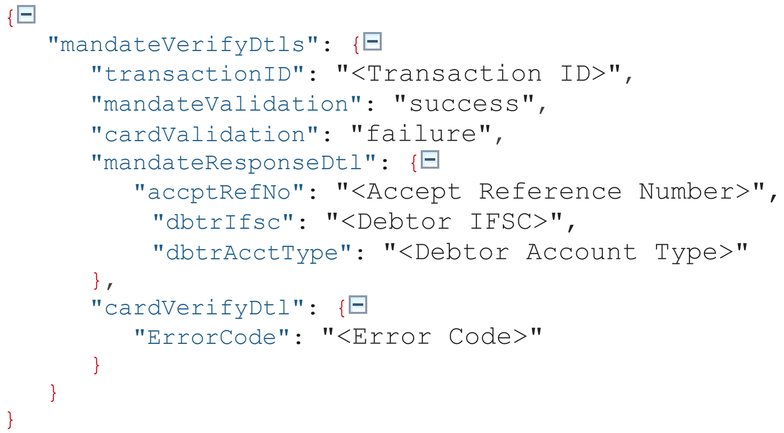
Note:-

Attribute values mandateValidation, cardValidation, ReasonCode & ReasonDesc needs to be encrypted

C. If the destination bank is able to successfully parse the mandate request XML and business validation passes, then bank needs to validate the card details. Bank needs to verify the below details:

   I. Verify Debit Card Number
   II. Verify Expiry / validity
   III. Verify CVV number
   IV. Verify PIN number if its applicable
   V. Account number of debit card matches with the "Debtor AccNo" provided in the mandate Request XML

If any of the above validation fails, then the bank needs to provide the response as below: -



```
{⊟
   "mandateVerifyDtls": {⊟
      "transactionID": "<Transaction ID>",
      "mandateValidation": "success",
      "cardValidation": "failure",
      "mandateResponseDtl": {⊟
         "accptRefNo": "<Accept Reference Number>",
          "dbtrIfsc": "<Debtor IFSC>",
          "dbtrAcctType": "<Debtor Account Type>"
      },
      "cardVerifyDtl": {⊟
         "ErrorCode": "<Error Code>"
      }
   }
}
```

Note:-

Attribute values mandateValidation, cardValidation, AccptRefNo & ErrorCode needs to be encrypted.

The below table provides the error codes details that are newly introduced for Direct Debit Card Flow

error_code_id	error_desc	applicable_leg
601	Invalid Debit Card Number	BTN
602	Invalid Expiry / Validity	BTN
603	Invalid CVV	BTN
604	Account Details Does not Match	BTN
605	Otp Verification Failure	BTN
606	Duplicate Request	MTN
607	Previous Request in Progress	MTN
608	Bank Restricts Duplicate request	MTN
609	Invalid PIN	BTN

D. If all the above validation passes then the bank needs to provide the response as below: -

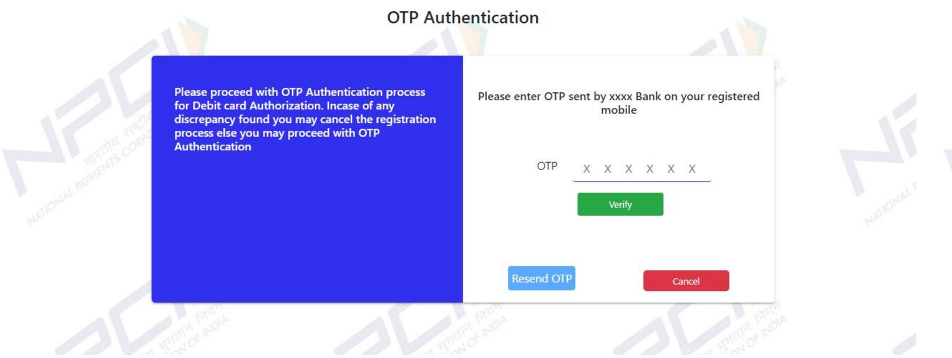
```
{
   "mandateVerifyDtls": {
      "transactionID": "<Transaction ID>",
      "mandateValidation": "success",
      "cardValidation": "success",
      "mandateResponseDtl": {
         "accptRefNo": "<Accept Reference Number>",
          "dbtrIfsc": "<Debtor IFSC>",
          "dbtrAcctType": "<Debtor Account Type>"
      },
      "cardVerifyDtl": {
         "successCode": "<Success Code>"
      }
   }
}
```

Note:-

- Attribute values mandateValidation, cardValidation, AccptRefNo & successCode needs to be encrypted
- IFSC code is optional field, if banks give invalid IFSC code in the Response ONMAGS system will update the IFSC code as per the Bank Masters
- Bank needs to store the mandate details received along with the transaction ID for the subsequent OTP validation.

For scenarios (a), (b) & (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) mandate status will be “In Process” for the bank until the OTP verification is completed.

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



OTP Authentication

Please proceed with OTP Authentication process for Debit card Authorization. Incase of any discrepancy found you may cancel the registration process else you may proceed with OTP Authentication

Please enter OTP sent by xxxx Bank on your registered mobile

OTP X X X X X X

- OTP will be a 6 digit numeric number.
- In case User did not receive OTP, there is an option to Resend OTP which the user can retry maximum of 3 times.

- Once user clicks on the verify button the entered OTP is encrypted and sent to the server. From the server end VerifyOTP API call will be made to the bank server.

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

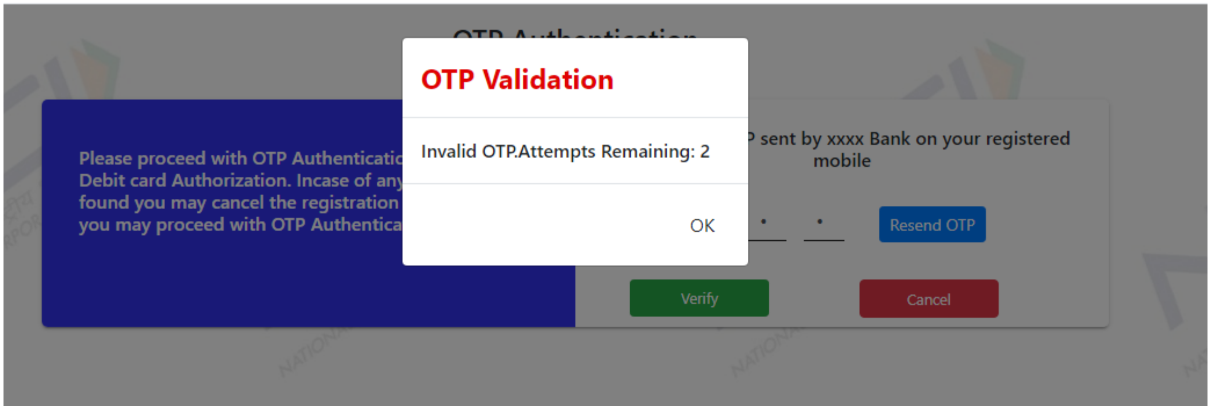
- In case of retry as well the request will be posted to bank in the above mentioned format only.

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.

```
   {⊟
      "otpVerifyInfo": ⊟
         {⊟
            "transactionID": "<Transaction ID>",
            "optVerifyStatus": "<Encrypted OTP verification status. It will be either
   success / failure>"
         }
}
```

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.



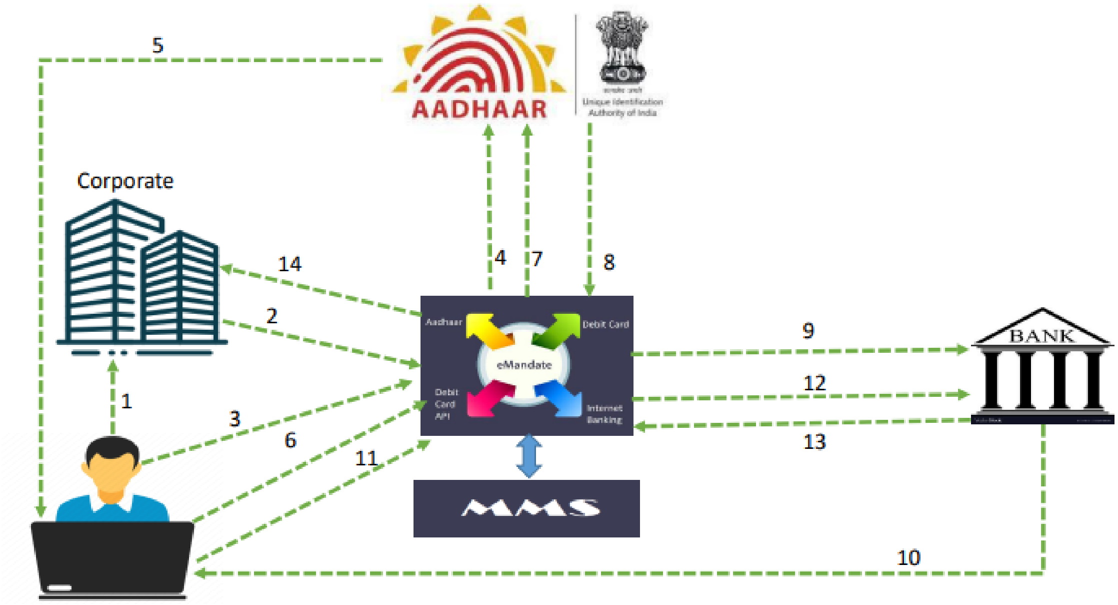
OTP Validation

Invalid OTP.Attempts Remaining: 2

Please proceed with OTP Authenticatic
Debit card Authorization. Incase of any
found you may cancel the registration
you may proceed with OTP Authentica

P sent by xxxx Bank on your registered mobile

4.2.3 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.

Step3: Customer enters Aadhaar Number along with required details.

Step4: ONMAGS Platform will forward that request to UIDAI for Customer Authentication via OTP generation

Step5: UIDAI will generate the OTP and send it to Customer's registered mobile number for Authentication

Step6: Customer will enter the OTP in the ONMAGS OTP page

Step7: ONMAGS Platform will forward that OTP to UIDAI for Verification

Step8: UIDAI sends response for OTP Verification. If the request is not authenticated by UIDAI then the flow ends here by showing the error message in Merchant Portal.

Step9: Once the customer is successfully authenticated, then the ONMAGS platform will send the mandate request to destination bank. If customer bank doesn't opt for additional OTP authentication then skip Step 10, Step 11 and Step 12.

Step10: After customer successfully authenticated by UIDAI, he/she will be landed on ONMAGS OTP page. ONMAGS will send an API request to customer's Bank to verify the customer details will generate the Bank OTP and send it to customer for Authentication.

Step11: Customer will enter the Bank OTP in ONMAGS platform for Authentication.

Step12: ONMAGS platform will forward that OTP to destination bank for Verification.

Step13: 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.

Step 14: ONMAGS Platform in turn redirects the response to Merchant Web Page where customer can view the response.

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 contains 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 contains 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	If Authmode is null, then user will get cardInfo JSON Object and consider as authomode as Debit Card else authMode value will be Aadhaar and user will get aadhaarInfo JSON Object in request.
aadhaarInfo	JSON Object	This will contains the aadhaar details and flag indicating that the customer authentication has been successful though UIDAI.
aadhaarNumber	String	Last four digit of aadhaar number
uidaiAuthenticated	Char	Always Y to be sent

Mandate Request to Bank:

```
{⊟
   "mandateAuthDtls": {⊟
      "transactionID": "<Transaction ID>",
      "mandateRequestDtl": {⊟
         "MandateReqDoc": "<Encrypted and Signed request XML>",
         "CheckSumVal": "<Check sum value of secure attributes>"
      },
      "authMode":"Aadhaar",
      "aadhaarInfo": {⊟
         "aadhaarNumber": "<Encrypted Aadhaar Number>",
         “uidaiAuthenticated” : “Y”
      }
   }
}
```

<u>Unencrypted and Unsigned request XML for MandateReqDoc Key:</u>

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
FrstColltnDt	Date of First Collection. Mandatory Field. This field is in ISODate Format	Alpha Numeric	16	
FnlColltnDt	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.
ColltnAmt	Either of ColltnAmt 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-xxxxxxxx. +91- is mandatory.
Mobile	Mobile Number of the Customer	Alpha Numeric	34	Should be given in the format +91-xxxxxxxxxx. +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

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.
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>",
      "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>",
      "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 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 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>",
      "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>",
      "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 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

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

A. Response
B. TimeStamp

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

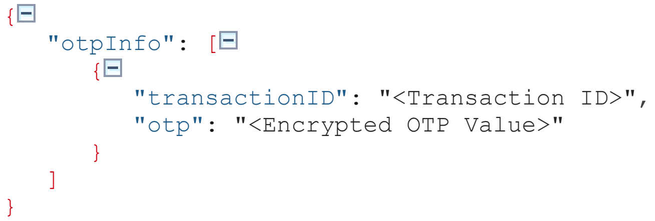
Generate checksum on the concatenated values. We will use SHA-2 as the hash function.

2. Signing of the Response JSON

- 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.

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 as well 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

```
    {⊟
       "otpVerifyInfo": [⊟
          {⊟
             "transactionID": "<Transaction ID>",
             "optVerifyStatus": "<Encrypted OTP verification status. It will be either
    success / failure>"
          }
       ]
}
```

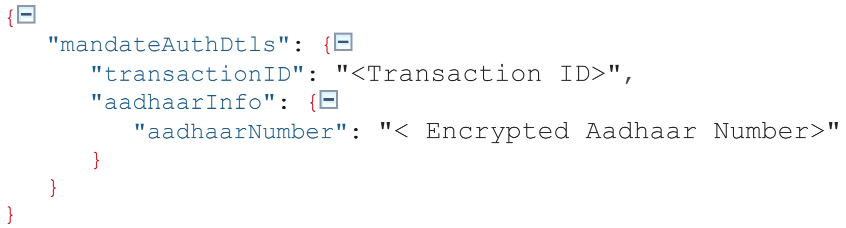
If OTP verification is successful only Bank needs to mark the mandate as accepted at their end. Until OTP validation is passed the mandate would be in non-accepted state at the Bank end.

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
aadhaarNumber	String	Encrypted aadhaar number only last four digit.

JSON Request:



```
{
   "mandateAuthDtls": {
      "transactionID": "<Transaction ID>",
      "aadhaarInfo": {
         "aadhaarNumber": "< 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.

If the bank chooses not to do the OTP authentication again, ONMAGS will send an API request to bank with the customer details, Aadhaar number and other mandate details. Bank can verify the details at their end, check whether Aadhaar linked to Account number, accept/reject the mandate and send the response back to ONMAGS.

Bank will not generate any OTP, skip the OTP verification step and needs 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

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

4.3 Signing and Encryption process

Below is the process for encryption & signing during the various flows.

- ➢ **NPCI to Bank**

  - ❖ Encryption will be done using the Public Key of the certificate shared by Bank.

  - ❖ Signing Using Private key certificate of NPCI

- **Bank to NPCI**
    - Encryption will be done using the Public Key of the certificate shared by NPCI.
    - Signing Using Private key certificate of Bank

4.4 Encoding Guidelines

The request XML & response XML posted to NPCI and received from NPCI should in encoded format. As part of encoding specific characters would be replaced by escaped character of those.

Symbol	Spelled	Escaped Character
‘	Single Quotes	&apos;
“	Double Quotes	&quot;
&	Ampersand	&amp;
<	Less Than	&lt;
>	Greater Than	&gt;

5. Response through Offline Server to Server Communication

To account for online failures, the response from Bank to NPCI needs to be sent using server to server communication as well.

NPCI will expose an API for accepting server to server communication from Bank. Bank needs to invoke this URL for posting response through server to server communication.

Note:

- Since communication is received both by browser redirection & server to server call, NPCI would mark the status of the transaction based on the first response received. The second communication received would be ignored.
- Error Code & Error Description list will be shared by NPCI.

5.1 Handling of Time out / not reachable Scenarios

During the entire flow time out can happen at various stages. The following timeouts needs to be maintained at individual levels across the participating entities.

Flow	Auth Mode/Request	Timeout	Remarks
NPCI to Bank	**Old Net Banking/Debit Card**	30Min	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI. (merchant can use Status API to know the status of the request)
	New Debit Card – Submit Card Details	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	New Debit Card – OTP Verification	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Aadhaar verification by UIDAI	90 Sec	No response from UIDAI for original request. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Aadhaar OTP Authentication	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Account verification by Bank	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at

			NPCI and user will be redirected to merchant site with appropriate error code
	Aadhaar Authentication – Bank OTP Authentication	90 Sec	No response from Bank for original request & for 3 subsequent sync requests. Request will be marked as timed out at NPCI and user will be redirected to merchant site with appropriate error code

Explained below are the actions taken at NPCI ONMAGS layer for timeouts happening at various levels.

➢ **NPCI to Bank**

Scenario-1: Destination Bank not reachable

Action: The request will be auto closed as Failed at NPCI end after the specified duration. Merchant will be shown respective error code.

➢ **Bank to NPCI**

Scenario-1: Bank has not responded to NPCI within the timeout period.

Action: NPCI will send list of transactions for which communication is not received from Bank at periodic interval. Once the pre-defined cut off time for the transaction is reached the transaction would be marked as "No Response from Bank" auto closed.

Scenario-2: Bank sends response to NPCI after the timeout period

Action: Any response after the time out period would be ignored by NPCI ONMAGS. The transaction would be treated as no response from Bank and the action for Scenario-1 would be followed.

Scenario-3: Bank sends invalid response to NPCI within the timeout period

Action: NPCI ONMAGS will mark the transaction as "Invalid Response from Bank" and corresponding Response XML with applicable error code will be send to Merchant.

Scenario-4: Bank is unable to reach NPCI.

Action: Bank needs to communicate the response to NPCI using server to server call. NPCI will update the transaction status at our end. (applicable only for net banking & Old debit card Flow)

5.1.1 JSON Response Formats

Given below are the JSON Response formats for Server to Server Communication.

Note:

Error Response XML would be shared in case the original request is not readable.

5.1.1.1 Bank to NPCI (Success & Business Rejections)

```
{
   "bankResponseDtl":[
      {
         "BANKID":"<Participant ID of the Bank in NACH>",
         "MandateRespDoc":"<Encrypted and Signed response XML>",
         "CheckSumVal":"<Check sum value of secure attributes>",
         "RespType":"RespXML"
      }
   ]
}
```

5.1.1.2 Bank to NPCI Error Response (Technical Rejections)

```
{
   "bankResponseDtl":[
      {
         "BANKID":"<Participant ID of the Bank in NACH>",
         "MandateRespDoc":"<ErrorResponse XML>",
         "RespType":"ErrorXML"
      }
   ]
}
```

6. API services

6.1API 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":"xxxxxxxxxxxxxxxxxxxx",
      "ReasonCode":"343",
      "ReasonDesc":"Invalid Account",
      "RejectBy":"Bank",
      "ErrorCode":"000",
      "ErrorDesc":"NA"
    },
    {
      "npcirefmsgID":"000f0f29dc27f00000101b09c5227457E23",
      "Accptd":"true",
      "AccptRefNo":"tranid352254221",
      "MndtId":"xxxxxxxxxxxxxxxxxxxx",
      "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

6.2 API for posting list of Open Transactions to Bank

i.

ii. NPCI will post the open transaction (transaction for which response has not been received from Bank end) to Bank at predefined interval. Bank should expose a listener for accepting the request from NPCI and send response in the same request (**Synchronous**). In the request NPCI provide either MndtId or NpciRefMsgID or Both as a input, bank ready to accept the input and provide details as mentioned in the below format.

➔ **Request:**

```
{
   "openMandateTrans":[

      {
         "MndtId":"xxxxxxxxxxxxxxxxxxxxx",
         "NpciRefMsgID":"000f0f29dc27f00000101b09c522743SK65"
      }
   ]
}
```

For the open transaction bank needs to provide the response in the same request mentioned in the below

Note:

Bank needs to provide the API URL for accepting this request which should accept the above JSON format.

Given below are the JSON Response formats.

→ Success Response

```
{
   "bankResponseDtl":[
      {
         "BANKID":"<Participant ID of the Bank in NACH>",
         "MandateRespDoc":"<Encrypted and Signed response XML>",
         "CheckSumVal":"<Check sum value of secure attributes>",
         "RespType":"RespXML"
      }
   ]
}
```

→ Error Response

Error Response XML would be shared in case the original request is not readable or in case they didn't receive any request for the given npcirefmsgid.

```
{
   "bankResponseDtl":[
      {
         "BANKID":"<Participant ID of the Bank in NACH>",
         "MandateRespDoc":"<ErrorResponse XML>",
         "RespType":"ErrorXML"
      }
   ]
}
```

6.3 HEART BEAT API

ONMAGS system will check the LIVE status of the Banks for particular interval. ONMAGS will send the HTTPS request to banks and in the same request banks give the response **(Synchronous).**

The request and response structure below.

For Banks

Banks can provide "Live" as response to NPCI only if banks can process the API E-Mandate successfully at their end. Assume if banks requires more than one service to successfully register a mandate, banks should check the availability of all services and provide "Live". Even if one service is not working, the response should be provided as "Not live".

6.3.1 Request:

The request will be in the Json format to their respective shared URL's

```
{

"action":"HEART BEAT REQUEST",

"data":

{

"server_status":"ALIVE",
"current_time":"2019-11-04T09:09:09"
}
}
```

6..3.2 Response:

```
{
"action": " HEART BEAT RESPONSE",
"data": {
"status": "ALIVE",
"current_time":"2019-11-04T09:09:09"
}
}
```

7. Appendix

7.1 Request & Response XML Specification for Banks

Validation_Sheet_Bank

Validation Sheet.xlsx

7.2 Sample XML Formats and Schemas

Request Response
Files_V7.zip

7.3 Error Codes

ErrorCode - Bank

Error_codes.xls

7.4 Bank Reject Reason codes

Bank_Reject_reason
_codes.xls

7.5 Guidelines and design for Netbanking page, Debit Card and corporate page

Bank page - API
E-Mandate - Interne

Bank page - API
E-Mandate - Debit c

Implementation
guidelines for Corp