

**NPCI Mandate Approval Gateway Service**

**Merchant Specification Document**

**Version 4.2**

**DEC, 2021**

**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 7](#_Toc92196618)

[1.1 Abbreviation 7](#_Toc92196619)

[2. Interface specification details for Mandate Approval 8](#_Toc92196620)

[2.1 Registration with NPCI 8](#_Toc92196621)

[2.2 Mandate Approval function flow (For Net Banking & Debit Card mode of authentication) 8](#_Toc92196622)

[2.2.1 End to End Process Flow (For Net Banking & Debit Card mode of authentication) 9](#_Toc92196623)

[2.3 Interface Layer 10](#_Toc92196624)

[3. Specification Format for Request & Response 11](#_Toc92196625)

[4. Technical Integration Specification 12](#_Toc92196626)

[4.1 Merchant Site Integration Requirements 12](#_Toc92196627)

[4.1.1 Request from Merchant to NPCI 12](#_Toc92196628)

[4.1.1.1 Encoding of Mandate Request XML 21](#_Toc92196629)

[4.1.2 Response from NPCI to Merchant 22](#_Toc92196630)

[4.1.2.1 Encoding of Response XML 23](#_Toc92196631)

[4.2 NPCI Gateway Specification 23](#_Toc92196632)

[4.2.1 Forward Flow specification from Merchant to NPCI 23](#_Toc92196633)

[4.2.2 Return Flow specification from NPCI to Merchant 24](#_Toc92196634)

[4.3 Authentication Modes 25](#_Toc92196635)

[4.3.1 Net Banking and Old Debit Card Mode of Authentication 25](#_Toc92196636)

[4.3.2 New Debit Card Flow 25](#_Toc92196637)

[4.3.3 Aadhaar Authentication Flow 28](#_Toc92196638)

[4.4 Signing and Encryption process 31](#_Toc92196639)

[4.5 Encoding Guideline 32](#_Toc92196640)

[4.6 Handling of Time out / not reachable Scenarios 32](#_Toc92196641)

[5. Miscellaneous Features 34](#_Toc92196642)

[5.1 API to get live destination banks for e-mandate 34](#_Toc92196643)

[5.2 API to get Transaction Status for Merchant 35](#_Toc92196644)

[5.3 API to Get Response posted to Merchant 37](#_Toc92196645)

[5.4 Restricting Duplicate Transactions 39](#_Toc92196646)

[6. About NPCI Gateway 41](#_Toc92196647)

[7. Appendix 41](#_Toc92196648)

[7.1 Request & Response XML Specification for Merchants 41](#_Toc92196649)

[7.2 Sample XML Formats and Schemas 41](#_Toc92196650)

[7.3 Error Codes 41](#_Toc92196651)

# 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.

# 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 |

# Interface specification details for Mandate Approval

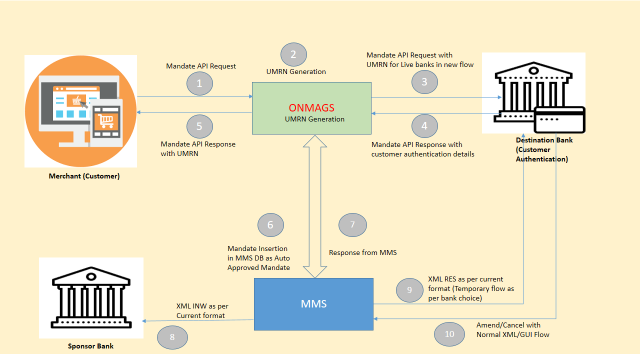
# Registration with NPCI

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

# 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.



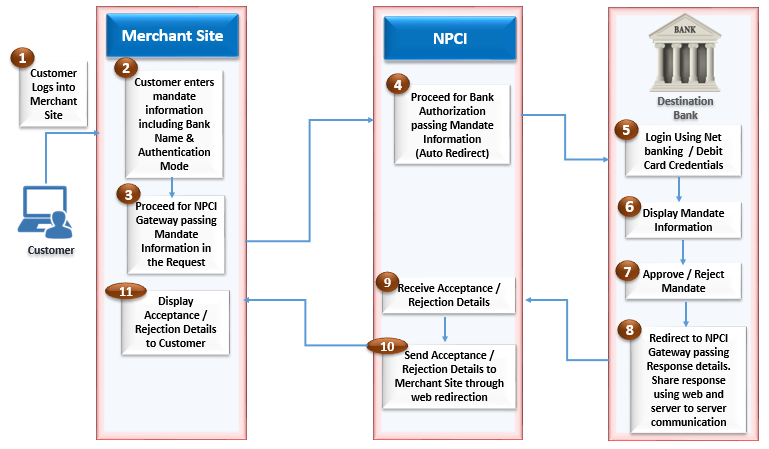
The process flow is mentioned in the next section.

**Note:**

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

# 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.



* 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.

# Interface Layer

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

* Signing Certificate

# Specification Format for Request & Response

[Appendix 9.1](#_heading=h.32hioqz)

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](#_heading=h.41mghml).

* 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.

# Technical Integration Specification

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

# Merchant Site Integration Requirements

# 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](https://apac01.safelinks.protection.outlook.com/?url=https%3A%2F%2F103.14.161.144%3A8086%2Fonmags%2FsendRequest&data=01%7C01%7C%7C921e5078abc64448e74808d5e00700f9%7C2d538e6436c741bc8b7d4d804956e957%7C1&sdata=ryrV%2B8dyOaM0k%2FnBD51dB%2FvRnq30Gp%2Fgl0cP0Exxulk%3D&reserved=0)

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:

1. Debtor Account Number
2. First Collection Date
3. Final Collection Date
4. Collection Amount
5. 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.

1. 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.

1. 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](#_heading=h.32hioqz) 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 |

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

# 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.

# 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](#_heading=h.1hmsyys).

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”.

# 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.

# 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.

# 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.

# 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:

1. Accptd
2. AccptRefNo
3. ReasonCode
4. ReasonDesc
5. 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.

1. **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.

1. **Signing of the Response XML**

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

1. **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.

# Authentication Modes

# 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.

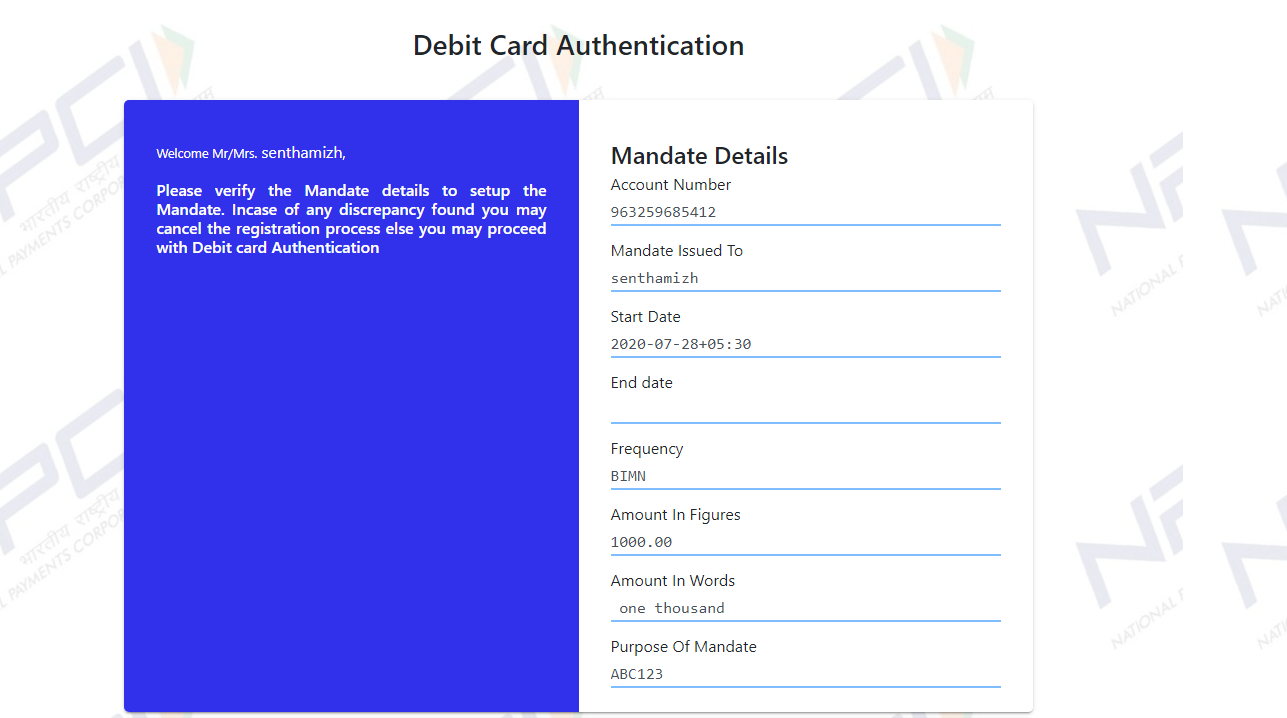
# New Debit Card Flow

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.

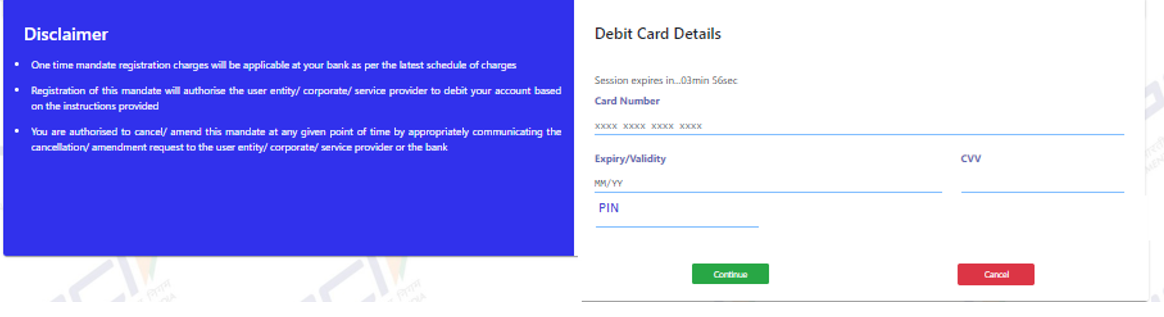
Debit Card Information will be accepted in ONMAGS page itself and validated with Bank through server to server call. The steps in this flow is described below:

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.



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.



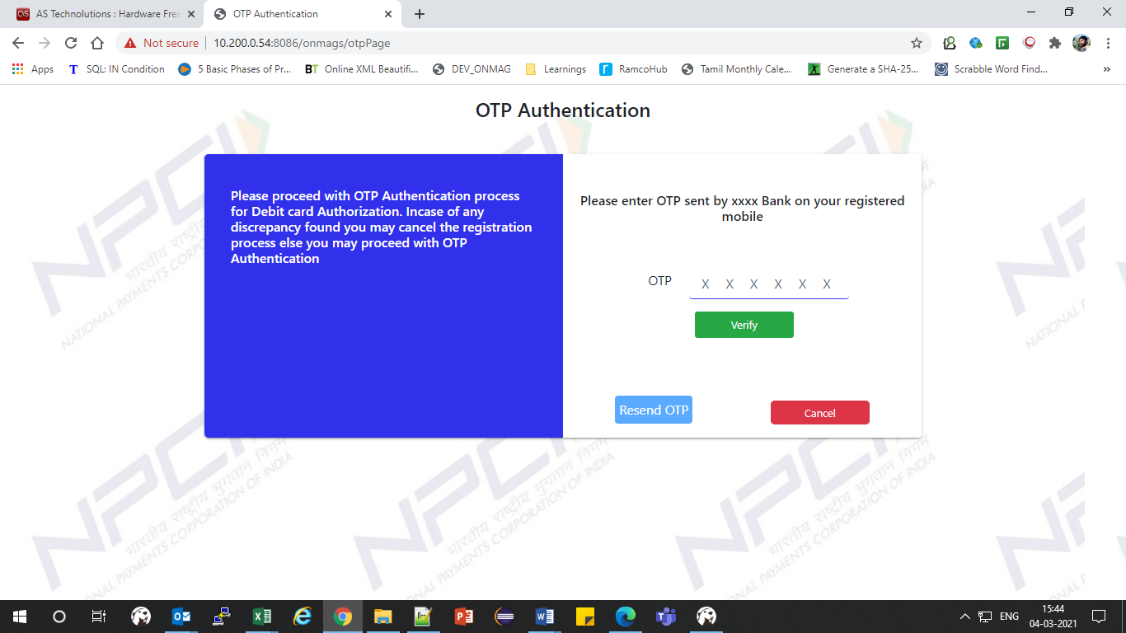
Below are the validation done related to the entered Debit Card Details

* 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

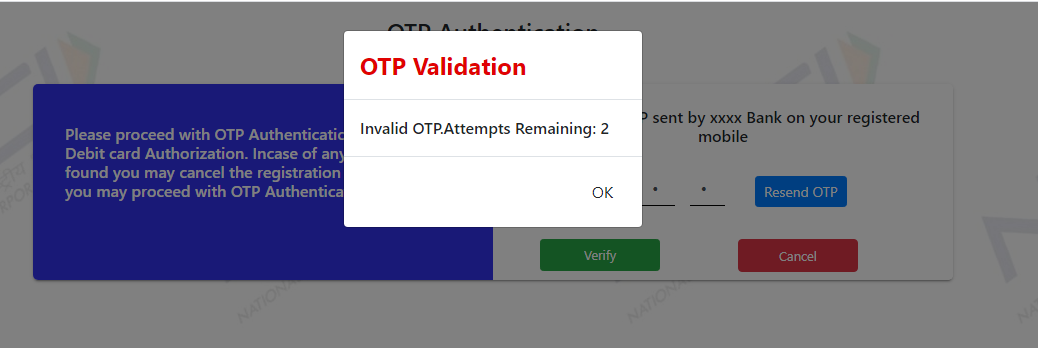
If Debit card validation fails, corresponding error message will be shown to the user. If the debit card validation gets passed at Bank site, then ONMAGS will redirect to the OTP verification page.



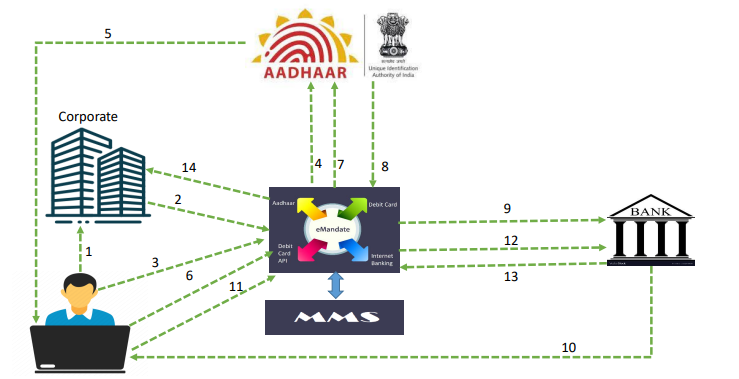
* OTP will be a 6 digit numeric number
* In case User did not receive OTP, there is an option to Resend OTP which the user can retry maximum of 3 times.
* In case of retry as well the request will be posted to bank in the above mentioned format only.

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.



# 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.

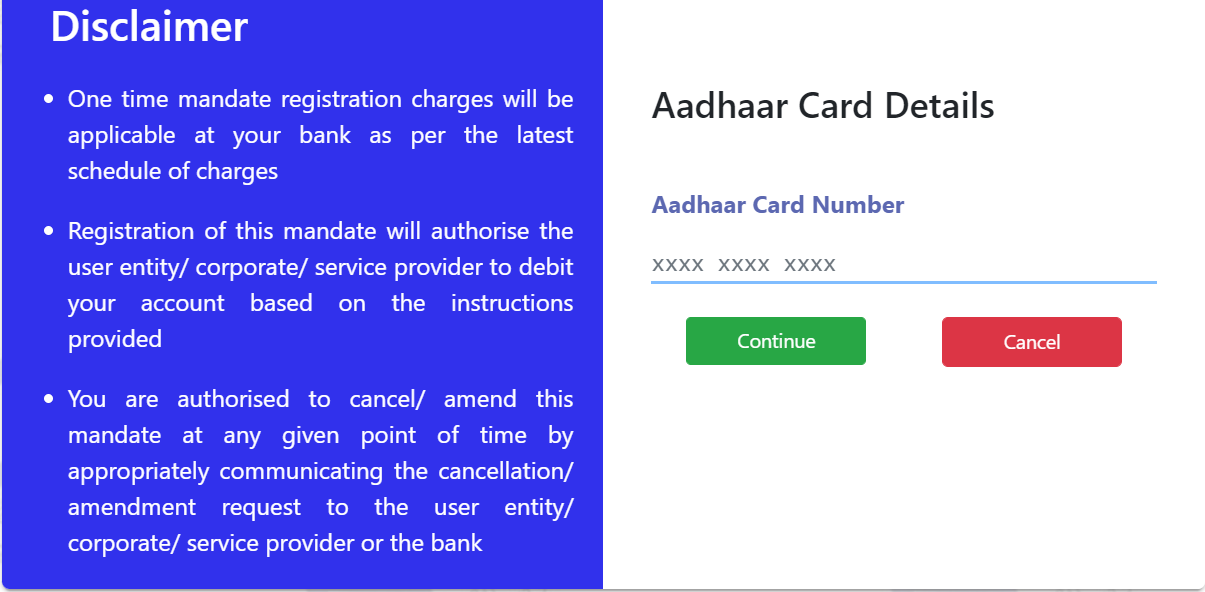
**Step11:** Customer will enter the Bank OTP in ONMAGS platform for Authentication.

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

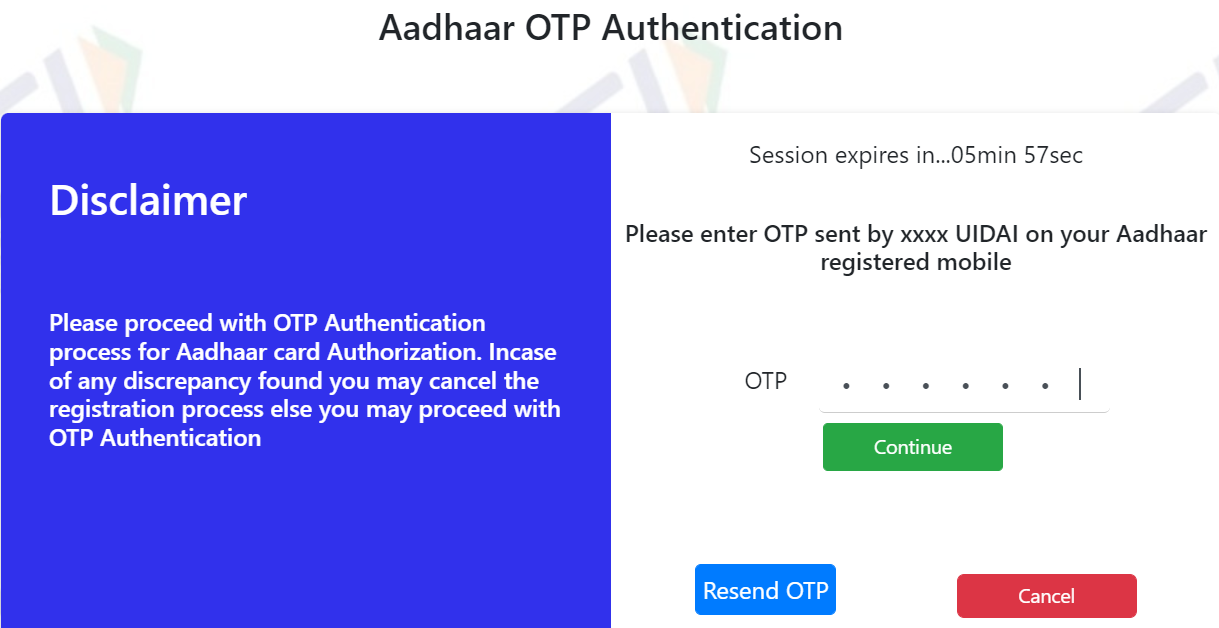
**Step13:** If OTP verification is successful only Bank needs to mark the mandate as accepted at their end. Until OTP validation is passed the mandate would be in non-accepted state at the Bank end.

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

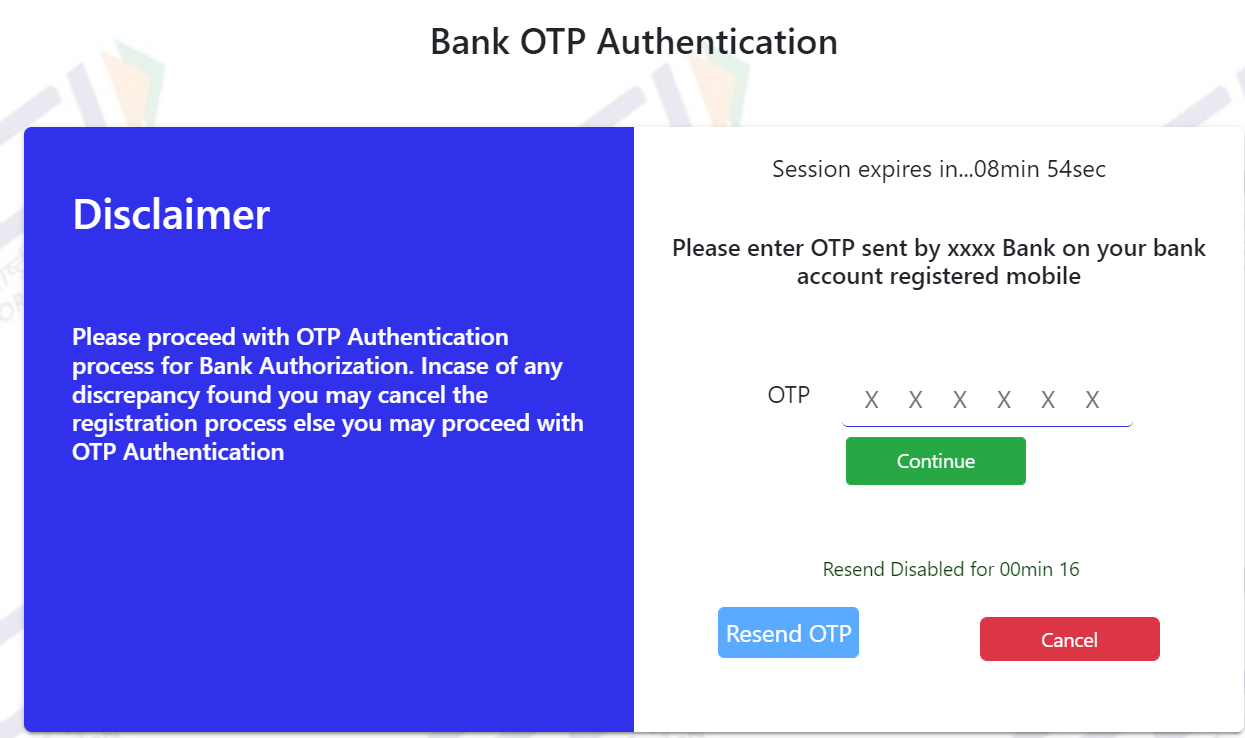
**ONMAGS page where customer enters his/her Aadhar Number**



**Aadhaar OTP Authentication:**



**Bank OTP Authentication:**



# 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

# 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; |

# 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.

# Miscellaneous Features

# 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>

# 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>

# 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>

# 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.

# 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.

# Appendix

# Request & Response XML Specification for Merchants

[Validation\_Sheet Merchant](https://drive.google.com/file/d/19mX7Jhh2qQWXSf40tJDeL42aeR_wD29I/view?usp=drive_web)



# Sample XML Formats and Schemas



# Reason & Error Codes

[merchant error codes](https://drive.google.com/file/d/1jRnsxoNEgw6HU_9i_5xK5WIxLZVNj8bn/view?usp=drive_web)