NACH-API & Technology Specifications

Version 1.0

Contents

1.1 Objectives 3
2. Communication Channel 4
2.1 Channel Encryption Details 4
2.2 Certificate Format 4
2.3 XML Format Changes for signature 5
2.4 Encryption and Signing process 6
3. API Protocol 7
3.1 Aadhaar Linkage to an account number 8
3.2 Obtaining/Get PAN Number 11
3.3 Validation of an account based on the basis of account number and IFSC 14
3.4 Request for an account holders name 17
3.5 General ACK/NACK Message format 20
3.6 Destination Asynchronous Failure Response Message format 20
3.7 Elements and Attributes Definition 22
3.8 Error Codes 30

1. Introduction

This is a value added services from NACH to all member banks and their customers. In the current scenario, the customer is solely dependent either on the corporates/ user institutions for all NACH related queries. Corporates in turn depended on their sponsor banks for getting the information from NACH. This service will help in eradicate the manual follow ups and will take lesser time to resolve any queries. These services can be implemented at the bank branches where the customer walks in and solution can be arrived instantly. This will also improve the overall efficiency of the echo system.

1.1 Objectives

Objectives of introducing Application interface in NACH is to offer a set of standard APIs to facilitate banks for verifying the user details before initiating the financial transactions or initiating the mandates into NACH application.

- To reduce the transaction return percentage.
- To avoid wrong debits/credits.
- Mandate information to the users through the bank.
- To check whether Aadhaar number is linked to the bank account

2. Communication Channel

2.1 Channel Encryption Details

NACH network communication channel should be encrypted and secured to maintain the secrecy and eligibility of the data travelling through the medium.

Source and Destination banks need to exchange the RSA public keys with NPCI as demonstrated in below figure.

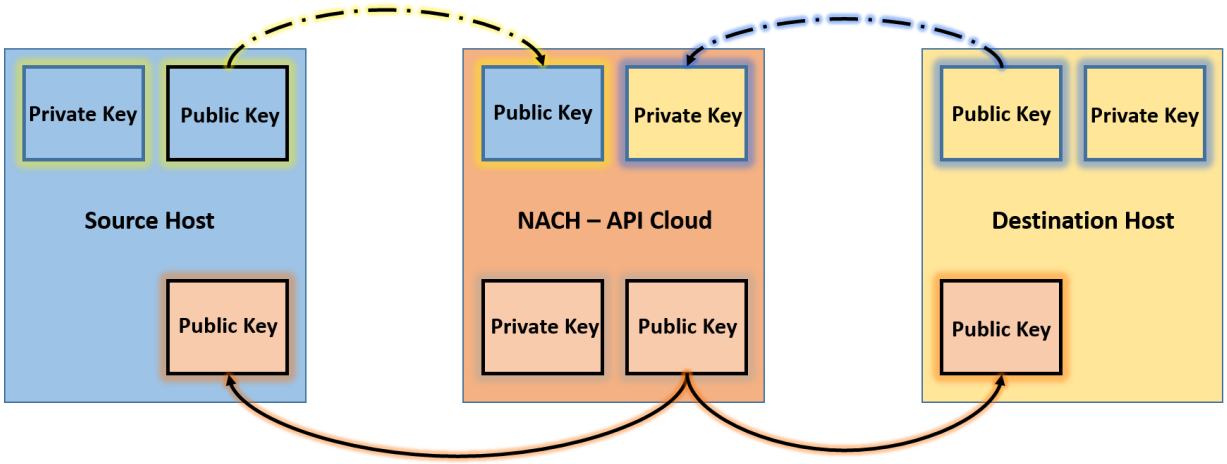


Figure 1 Key Exchange

Encryption Description:

- Source Bank signs the XML message using their private key.
- When Source message received at NACH system, it trusts the message using Source Bank Public Key.
- Handshake protocol will exchange public key certificate to authenticate server (i.e., Source Host) & client to each other.
- In case of RSA key exchange,
    - Source Host generate pre-master secret.
    - Pre-master secret is encrypted using NACH Public Key.
    - NACH can decrypt the PMK (Pre Master Key) using Private Key.
- Similarly, when Destination Host acts as server the respective public key will be used to exchange the pre-master secret key.
- Client authentication by server is mandatory.

2.2 Certificate Format

Certificate formats

- 509 certificates v3: (etc.npci.org.in)
    - We need fully qualified name
    - No wildcards in certificate
- TLS_RSA_WITH_AES_256_CBC_SHA

Chiper Suites

- Key exchange- RSA
  - Authentication- RSA 2048
  - Block Chiper AES 256
- Hash -SHA 256 (HMAC & PRF)

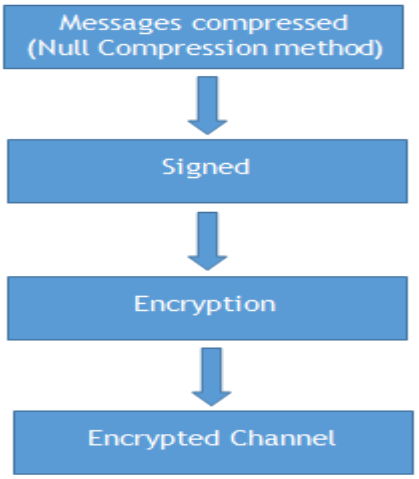


Figure 2 Flow Chart

2.3 XML Format Changes for signature


```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ach:AadhaarLinkageRqst xmlns:ach="http://npci.org/ach/schema/">
<Head ts="ISO Timestamp" ver="1.0"/>
<Source name="" type="CODE" value=""/>
<Destination name="" type="CODE" value=""/>
<Request id="" refUrl="" type="Status ENQ|Details ENQ"/>
<ReqData>
       <Detail aadhaar="" accNo=""/>
</ReqData>
<NpciRefId value=""/>
<Signature
xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/><SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/><Reference
URI=""><Transforms><Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/><DigestValue>Y6WMVFx5jmkVUTbf+
ls853ZpqLlMJpXIRjcI/d2k0Mo=</DigestValue></Reference></SignedInfo><SignatureValue>C
hiWo4jtdiV7q5T9t3F7aSpAy0Vndue80U2ByU6pVQnvotCxwFpRnC3g3gynnlDwl/7xVJsYucOm
8xY3YzD8kfcs79oO2f32gEOTrbOrLX0WyPI5aTpowuwWdB7oTIaFhtPUF6gvF/CR5mGyAgDb82uE
gbXktdT10U7DSo1IobZWrP4Q7XCu+XLTbi9WL2MgxsRymGEKOYiaZgWU/5BiQ+tjabKoYXDbALLR
6GxJ5X1girj6PM0nQ7EIxKPyltWyW6ycaGTwUMHykAc3OBvQfxHxlNMsvbvxBiNmzOmkFr9WPfrh
jjTq82BmdzS/fygmwAV1r8K2wzuKR4gwV/DFNA==</SignatureValue><KeyInfo><X509Data><X509Su
bjectName>CN=cm.npci.org.in,ST=TN,OU=Centralized
Mapper,O=NPCI,C=IN</X509SubjectName><X509Certificate>MIIFfTCCBGWgAwIBAgIKECMkJ8Fo6z
Es8zANBgkqhkiG9w0BAQsFADCB9jELMAkGA1UEBhMCSU4x
RTBDBgNVBAoTPEluc3RpdHV0ZSBmb3IgRGV2ZWxvcG1lbnQgYW5kIFJlc2VhcmNoIGluIEJhbmtp
bmcgVGVjaG5vbG9neTEdMBsGA1UECxMUQ2VydGlmeWluZyBBdXRob3JpdHkxDzANBgNVBBETBjUw
MDA1NzESMBAGA1UECBMJVGVsYW5nYW5hMSkwJwYDVQQJEyBSb2FkIE5vLjEsIE1hc2FiIFRhbmss
```

IEh5ZGVyYWJhZDEVMBMGA1UEMxMMQ2FzdGxlIEhpbGxzMRowGAYDVQQDExFJRFJCVCBDQSBTUEwg
MjAxNjAeFw0xNjAyMjYxMDAyMzZaFw0xODAyMjYxMDAyMzZaMF8xCzAJBgNVBAYTAklOMQ0wCwYD
VQQKEwROUENJMRswGQYDVQQLExJDZW50cmFsaXplZCBNYXBwZXIxCzAJBgNVBAgTAlROMRcwFQYD
VQQDEw5jbS5ucGNpLm9yZy5pbjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALnFiqPU
62LW9nrWS7D97ToLv72u51pGixLdazIg30hh6/ZjkNSlqtIGSBC+TT8My01a4lWRkWuj4khy83RJ
8TMIrEM8AHghESzXp5aHbN65XXuWefTVy3HdACIOMgK/guC6cbH1fFsfv/FpO4RIZjoEJDHYUDrr
Mxmua6pXi3buHQN/DhNWk4jZ9wEibIkpK/d0J/xOyVUyGPdskQH6ejOHBMnp3VoYGQXTCuLOkLJP
g6LMNXry1+PpezSyuq4ns+IVbEWmJoQC6VefrZvbASFqzFiI6P5HkrFTev/mNMWlsFTwUVqxZRqz
WfEYbKKIjC0fwdCU6mOqnJ+6Tm2a/ikCAwEAAaOCAaEwggGdMA4GA1UdDwEB/wQEAwIFoDAdBgNV
HSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwgbUGA1UdIASBrTCBqjCBpwYGYIJkZAIDMIGcMIGZ
BggrBgEFBQcCAjCBjBqBiUNsYXNzIDMgbGV2ZWwgaXMgcmVsZXZhbnQgdG8gZW52aXJvbm1lbnRz
IHdoZXJlIHRocmVhdHMgdG8gZGF0YSBhcmUgaGlnaCBvciB0aGUgY29uc2VxdWVuY2VzIG9mIHRo
ZSBmYWlsdXJlIG9mIHNlY3VyaXR5IHNlcnZpY2VzIGFyZSBoaWdoMFkGA1UdHwRSMFAwJKAioCCG
Hmh0dHA6Ly8xMC4wLjY1LjY1L2NybF8yN0JBLmNybDAooCagJIYiaHR0cDovL2lkcmJ0Y2Eub3Jn
LmluL2NybF8yN0JBLmNybDAfBgNVHSMEGDAWgBRTPr+bw5KofjGVW9KXh1YFP/TqEzAZBgNVHREE
EjAQgg5jbS5ucGNpLm9yZy5pbjAdBgNVHQ4EFgQUsxGtCEaHH+sxEkkb9ukhi7Y98vIwDQYJKoZI
hvcNAQELBQADggEBAMZbQ2FSpCTKDQK/LSKeWOsDaWfhE7lp7EKskCgWKfCdmcV5SVXz0+29vQ6I
wY93kXndGdNSg9RuO17OOeX8Ow6uJXtnrmyBCT7LFD0BrTtnqOIu+l7Y4KUjJGbyprJErVLIluqI
NkeHcO1qmyFsd6yut21b0GT8p7q9/5gIciTkQvMhsBBSsVPf4TLuHEzBFoBe7v2DD19DncWGiPYm
bwM7EaTy2SCcR1vEZjAkLYgMLT2OsSnOBPVf1zoc6Pt+g6pSbvmPA5maLgm7AAAcYDwERYXHRdsE
JvylE/w5lIg1KNwIcMdBMIn7ZbHyeMKE7wngYr8ZcR022kqN9DEFkPU=</X509Certificate></X509Dat
a></KeyInfo></Signature>
</ach:AadhaarLinkageRqst>

2.4 Encryption and Signing process

Below is the process for encryption during the various flows.

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

Aadhaar number, Account number and PAN number attributes should be encrypted in all the request and response messages

3. API Protocol

All APIs are exposed as stateless service over HTTPS. Usage of open data format in XML and widely used protocol such as HTTP allows easy adoption by the members.
API input data should be sent to the following URL as JSON content using Content-Type as "text/plain". All APIs are processed in asynchronous manner.

Following is the URL format for all API's: **https://<host>/<ach>/<api>**

host – API server address (Actual production server address will be provided to members at the time of rollout and all API clients should ensure that actual URL is configurable).
ach – static value denoting the root of all API URL paths under the National Automated Clearing System
api – name of the API URL endpoint.

Following is the JSON format of request and response messages.

Source :<< Source Bank Short Name>>

Service : <<Service Name>>

Type : <<Request | Response >>

Message : << Signed XML message of actual Request or Response >>

Example:

Source :<< Source Bank Short Name >>

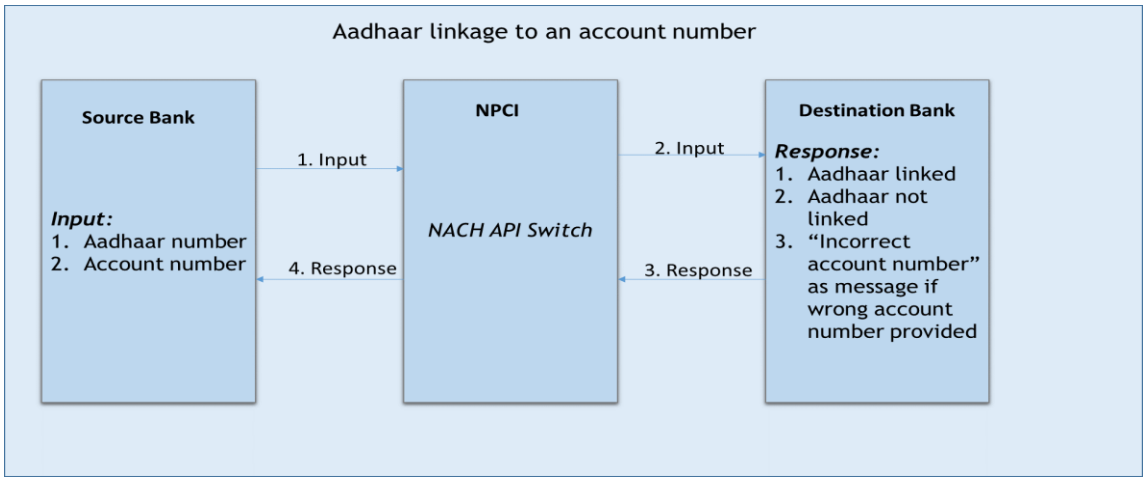
Service: AadhaarLinkage

Type : Request

Message :
```
<ach:AadhaarLinkageRqst xmlns:ach="http://npci.org/ach/schema/" >
<Head ver="1.0" ts="<<ISO Timestamp>>" />
<Source type="CODE" value="" name="" />
 <Destination type="CODE" value="" name="" />
<Request id="" type="Status ENQ|Details ENQ" refUrl=""/>
<ReqData>
    <Detail aadhaar="" accNo="" />
 </ReqData>
 <NpciRefId value=""/>
</ach:AadhaarLinkageRqst>
```

3.1 Aadhaar Linkage to an account number

This API is initiated by Source Banks to know the Aadhaar linkage of specific Aadhaar number and account number. Upon receivable of the message, NACH –API Cloud system acknowledges the request to source bank by sending either ACCEPTED or ERROR. After successful validation of NACH – API cloud system passes the message to destination Banks based on the registered URL of destination Bank.



Input data: Aadhaar number, account number (All fields are mandatory)
Responding entity: Bank
Response to be provided:

1. Aadhaar linked
2. Aadhaar not linked
3. "Incorrect account number" as message if wrong account number provided

Request Message Format

```
<ach:AadhaarLinkageRqst xmlns:ach="http://npci.org/ach/schema/" >
<Head ver="1.0" ts="2017-10-16T10:02:00" />
<Source type="CODE" value="SBIN" name="State Bank of India" />
<Destination type="CODE" value="AXIS" name="" />
<Request id="123456789" type="STATUS_ENQ" refUrl=""/>
<ReqData>
      <Detail aadhaar="234567890123" accNo="56475648900909" />
</ReqData>
<NpciRefId value=""/>
</ach:AadhaarLinkageRqst>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1

Index	Message Item	<XML Tag>	Occurrence
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message (Transmission time)	ts	1..1
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	Input data related to the request	<ReqData>	1..1
6.2	Details of the Input parameters of the request	<Details>	1..1
6.2.1	First parameter of the request - Aadhaar number	aadhaar	1..1
6.2.2	Second parameter of the request – Account Number	accNo	1..1
7.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	0..1
7.1.1	Actual unique value generated by NPCI	Value	0..1

Response Message Format

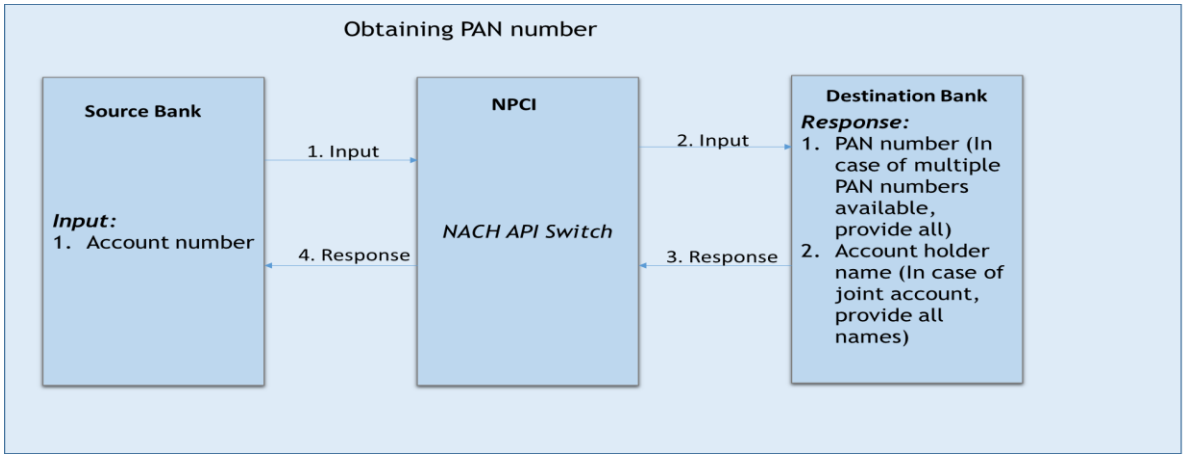
```
<ach:AadhaarLinkageResp xmlns:ach="http://npci.org/ach/schema/">
<Head ver="1.0" ts="2017-10-16T10:02:15" >
<Source type="CODE" value="SBIN" name="State Bank of India" />
<Destination type="CODE" value="AXIS" name="" />
<Request id="123456789" type="STATUS_ENQ" refUrl=""/>
<Resp ts="2017-10-16T10:02:15" result="SUCCESS" errCode="" rejectedBy="" />
<RespData>
      <Aadhaar linkStatus="YES" />
</RespData>
<NpciRefId value="6afd4578-f021-4321-a908-04b355a758fa" />
</ach:AadhaarLinkageResp>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message (Transmission time)	ts	1..1
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	Unique Identifier assigned by NPCI for the request	<NpciRefId>	1..1
6.1.1	Actual unique value generated by NPCI	value	1..1
7.1	Response of the Message	<Resp>	1..1
7.1.1	Time of response from the sender of the message	ts	1..1
7.1.2	Result of the request	result	1..1
7.1.3	Error reason codes for the failure message	errCode	0..1
7.1.4	Actual rejecter of the message. NPCI or Destination who rejected the message	rejectedBy	0..1
8.1	Response Data	<RespData>	0..1
8.2	Aadhaar Linkage Status	<Aadhaar>	0..1
8.2.1	Actual value of linkage Status	linkStatus	0..1

3.2 Obtaining/Get PAN Number

This API is initiated by Source Banks to know the PAN number of the given account number.

Upon receivable of the message from source, NACH –API Cloud system acknowledges the request to source bank by sending either ACCEPTED or ERROR. After successful validation of NACH – API cloud system passes the message to destination Banks based on the registered URL of destination Bank.



Input data: Account number
Responding entity: Bank
Response to be provided:

1. PAN number (In case of multiple PAN numbers available, provide all)
2. Account holder name (In case of joint account, provide all names

Request Message Format

```
<ach:GetPanDtlsRqst xmlns:ach="http://npci.org/ach/schema/" >
<Head ver="1.0" ts="<<ISO Timestamp>>" />
<Source type="CODE" value="" name="" />
<Destination type="CODE" value="" name="" />
<Request id="" type="Status Enq|Details Enq" refUrl=""/>
<ReqData>
      <Detail accNo="" />
</ReqData>
<NpciRefId value="" />
</ach:GetPanDtlsRqst>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message	ts	1..1

Index	Message Item	<XML Tag>	Occurrence
	(Transmission time)		
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	input data related to the request	<ReqData>	1..1
6.2	Details of the Input parameters of the request	<Details>	1..1
6.2.1	Parameter of the request – Account Number	accNo	1..1
7.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	0..1
7.1.1	Actual unique value generated by NPCI	Value	0..1

Response Message Format

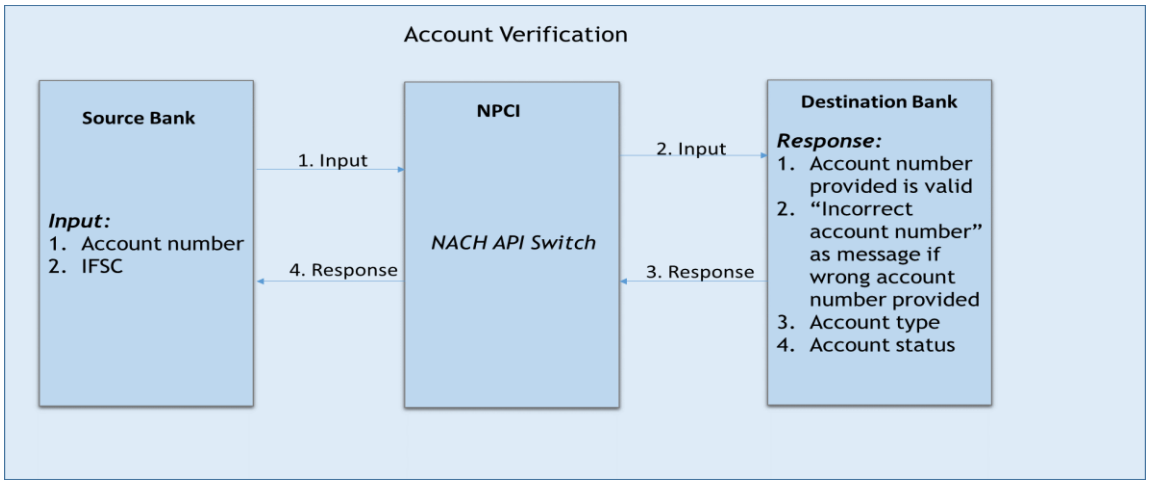
```
<ach:GetPanDtlsResp xmlns:ach="http://npci.org/ach/schema/">
<Head ver="1.0" ts="<<ISO Timestamp>>" />
<Source type="CODE" value="" name="" />
<Destination type="CODE" value="" name="" />
<Request id="" type="Status Enq|Details Enq" refUrl=""/>
<NpciRefId value="gkhdfkghfk13274973" />
<Resp ts="<<ISO Timestamp>>" result="SUCCESS" errCode="" rejectedBy=""/>
<RespData>
      <AccHolderList>
            <AccHolder pan="" name="" />
            <AccHolder pan="" name="" />
      </AccHolderList>
</RespData>
</ach:GetPanDtlsResp>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message (Transmission time)	ts	1..1
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	1..1
6.1.1	Actual unique value generated by NPCI	value	1..1
7.1	Response of the Message	<Resp>	1..1
7.1.1	Time of response from the sender of the message	ts	1..1
7.1.2	Result of the request	result	1..1
7.1.3	Error reason codes for the failure message	errCode	0..1
7.1.4	Actual rejecter of the message. NPCI or Destination who rejected the message	rejectedBy	0..1
8.1	Response Data	<RespData>	0..1
8.2	List of account holder details	<AccHolderList>	0..1
8.3	Each Account Holder details	<AccHolder>	0..1
8.3.1	Pan number of account holder	pan	0..1
8.3.2	Name of the account holder	name	0..1

3.3 Validation of an account based on the basis of account number and IFSC

This API is initiated by Source Banks to know the status the given account number and IFSC number.

Upon receivable of the message from source, NACH –API Cloud system acknowledges the request to source bank by sending either ACCEPTED or ERROR. After successful validation of NACH – API cloud system passes the message to destination Banks based on the registered URL of destination Bank.



Input data: Account number and IFSC (Both mandatory)
Responding entity: Bank
Response to be provided:

1. Account number provided is valid
2. Account status
3. Account type

<u>Request Message Format</u>

```
<ach:GetAccStatusRqst xmlns:ach="http://npci.org/ach/schema/" >
<Head ver="1.0" ts="<<ISO Timestamp>>"/>
<Source type="CODE" value="" name="" />
<Destination type="CODE" value="" name="" />
<Request id="" type="Status Enq|Details Enq" refUrl=""/>
<ReqData>
      <Detail accNo="" ifsc="" />
</ReqData>
<NpciRefId value="" />
</ach:GetAccStatusRqst>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message (Transmission time)	ts	1..1
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	input data related to the request	<ReqData>	1..1
6.2	Details of the Input parameters of the request	<Details>	1..1
6.2.1	First parameter of the request – Account Number	accNo	1..1
6.2.2	Second parameter of the request – IFSC number	Ifsc	1..1
7.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	0..1
7.1.1	Actual unique value generated by NPCI	Value	0..1

Response Message Format

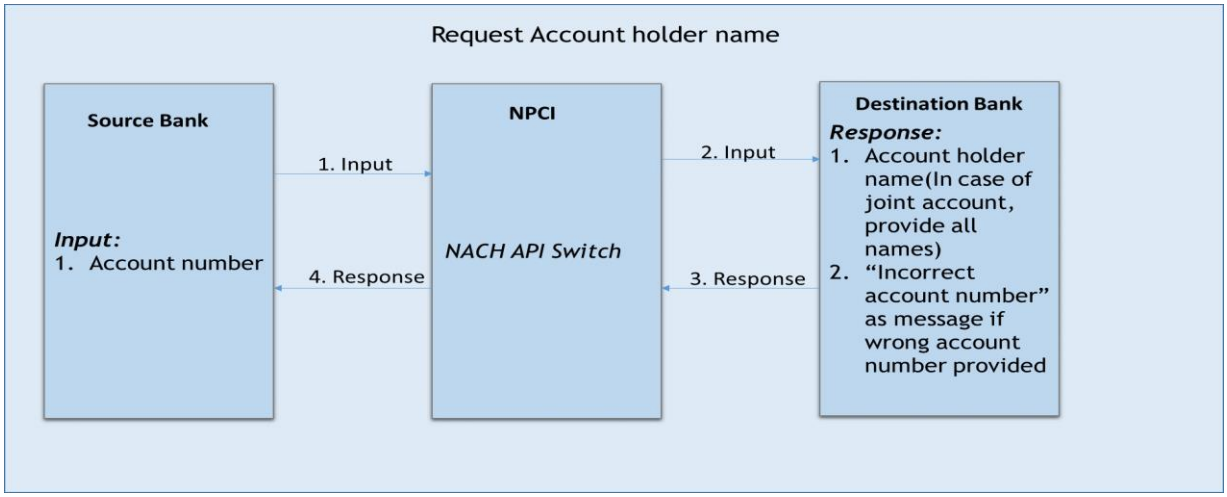
```
<ach:GetAccStatusResp xmlns:ach="http://npci.org/ach/schema/">
<Head ver="1.0" ts="<<ISO Timestamp>>" />
<Source type="CODE" value="" name="" />
<Destination type="CODE" value="" name="" />
<Request id="" type="Status Enq|Details Enq" refUrl=""/>
<NpciRefId value="12345678990" />
<Resp ts="<<ISO Timestamp>>" result="SUCCESS" errCode="" rejectedBy=""/>
<RespData>
      <Account type="" status="" />
</RespData>
</ach:GetAccStatusResp>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message (Transmission time)	ts	1..1
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	1..1
6.1.1	Actual unique value generated by NPCI	value	1..1
7.1	Response of the Message	<Resp>	1..1
7.1.1	Time of response from the sender of the message	ts	1..1
7.1.2	Result of the request	result	1..1
7.1.3	Error reason codes for the failure message	errCode	0..1
7.1.4	Actual rejecter of the message. NPCI or Destination who rejected the message	rejectedBy	0..1
8.1	Response Data	<RespData>	0..1
8.2	Account details	<Account>	0..1
8.2.1	Type of the account	type	0..1
8.2.2	Status of the account	status	0..1

3.4 Request for an account holders name

This API is initiated by Source Banks to know the account holder details such as name of the given account number.

Upon receivable of the message from source, NACH –API Cloud system acknowledges the request to source bank by sending either ACCEPTED or ERROR. After successful validation of NACH – API cloud system passes the message to destination Banks based on the registered URL of destination Bank.



Input data: Account number
Responding entity: Bank
Response to be provided:

1. Account holder name(In case of joint account, provide all names)
2. "Incorrect account number" as message if wrong account number provided

<u>Request Message Format</u>

```
<ach:GetAccHolderRqst xmlns:ach="http://npci.org/ach/schema/" >
<Head ver="1.0" ts="<<ISO Timestamp>>"/>
<Source type="CODE" value="" name="" />
<Destination type="CODE" value="" name="" />
<Request id="" type="Status Enq|Details Enq" refUrl=""/>
<ReqData>
      <Detail accNo="" />
</ReqData>
<NpciRefId value="" />
</ach:GetAccHolderRqst>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1

Index	Message Item	<XML Tag>	Occurrence
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message (Transmission time)	ts	1..1
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	input data related to the request	<ReqData>	1..1
6.2	Details of the Input parameters of the request	<Details>	1..1
6.2.1	First parameter of the request – Account Number	accNo	1..1
7.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	0..1
7.1.1	Actual unique value generated by NPCI	Value	0..1

Response Message Format

```
<ach:GetAccHolderResp xmlns:ach="http://npci.org/ach/schema/">
<Head ver="1.0" ts="<<ISO Timestamp>>" />
<Source type="CODE" value="" name="" />
<Destination type="CODE" value="" name="" />
<Request id="" type="Status Enq|Details Enq" refUrl=""/>
<Resp ts="<<ISO Timestamp>>" result="SUCCESS" errCode="" rejectedBy=""/>
<RespData>
      <AccHolderList>
            <AccHolder name="ABCD" />
            <AccHolder name="EFGH" />
      </AccHolderList>
</RespData>
<NpciRefId value="12345678990" />
</ach:GetAccHolderResp>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Header for the message	<Head>	1..1
2.1.1	Version of the API	ver	1..1
2.1.2	Time of request from the creator of the message (Transmission time)	ts	1..1
3.1	Source of the message	<Source>	1..1
3.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
3.1.2	Actual value of the routing type	value	1..1
3.1.3	Name of the Source Bank	name	0..1
4.1	Destination of the message	<Destination>	1..1
4.1.1	Routing type of the Source banks – based on short code of the bank the corresponding URL will be identified	type	1..1
4.1.2	Actual value of the routing type	value	1..1
4.1.3	Name of the Destination Bank	name	0..1
5.1	Request Message property	<Request>	1..1
5.1.1	Id of the Request generated by the originator	Id	1..1
5.1.2	Type of the request. Ie indication of status enquiry or details enquiry	Type	1..1
5.1.3	Reference URL for the transaction	refUrl	0..1
6.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	1..1
6.1.1	Actual unique value generated by NPCI	value	1..1
7.1	Response of the Message	<Resp>	1..1
7.1.1	Time of response from the sender of the message	ts	1..1
7.1.2	Result of the request	result	1..1
7.1.3	Error reason codes for the failure message	errCode	0..1
7.1.4	Actual rejecter of the message. NPCI or Destination who rejected the message	rejectedBy	0..1
8.1	Response Data	<RespData>	0..1
8.2	List of account holder details	<AccHolderList>	0..1
8.2.1	Each Account Holder details	<AccHolder>	0..1
8.2.2	Name of the account holder	name	0..1

3.5 General ACK/NACK Message format

ACK/NACK will be a HTTP Response with below XML body.

```
<ach:GatewayAck xmlns:ach="http://npci.org/ach/schema/" >
 <NpciRefId value="6afd4578-f021-4321-a908-04b355a758fa"/>
 <Resp ts="2017-10-16T10:02:00" result="ACCEPTED" errCode="" rejectedBy="" />
</ach:GatewayAck>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	1..1
2.1.1	Actual unique value generated by NPCI	value	1..1
3.1	Response of the Message	<Resp>	1..1
3.1.1	Time of response from the sender of the message	ts	1..1
3.1.2	Result of the request	result	1..1
3.1.3	Error reason codes for the failure message	errCode	0..1
3.1.4	Actual rejecter of the message. NPCI or Destination who rejected the message	rejectedBy	0..1

Note:

- ✓ This Acknowledgement will follow single root element of type GatewayAck for all type of request and response
- ✓ In case of Request Message, new NpciRefID will be generated and shared along with Resp
- ✓ Irrespective of success or failure scenarios at Gateway level, the NpciRefId will be generated for Request and shared
- ✓ For Response Type, the NpciRefID will be empty for both success and failure scenarios

3.6 Destination Asynchronous Failure Response Message format

```
Source  : NPCI
Service : GetMandateDtls
Type    : ErrorNote
Message :<ach:DestErrorNotification xmlns:ach="http://npci.org/ach/schema/">
         <NpciRefId value="6afd4578-f021-4321-a908-04b355a758fa" />
         <Resp ts="2017-10-16T10:02:26" result="ERROR" errCode="245"
          rejectedBy="NPCI" />
        </ach:DestErrorNotification>
```

Index	Message Item	<XML Tag>	Occurrence
1.1	API Name	<ach>	1..1
1.1.1	API Schema namespace	xmlns	1..1
2.1	Unique Identified assigned by NPCI for the request	<NpciRefId>	1..1
2.1.1	Actual unique value generated by NPCI	value	1..1
3.1	Response of the Message	<Resp>	1..1
3.1.1	Time of response from the sender of the message	ts	1..1
3.1.2	Result of the request	result	1..1
3.1.3	Error reason codes for the failure message	errCode	0..1
3.1.4	Actual rejecter of the message. NPCI or Destination who rejected the message	rejectedBy	0..1

Note:

- ✓ Any failure found in technical or business validation of Response message will trigger this notification
- ✓ Only for failure scenarios, the Destination bank will receive this notification

3.7 Elements and Attributes Definition

Element: Root

Definition: XML root element representing each API (GetMandateDtls, GetAadhaarLinkagr, etc...)

Attribute: xmlns

Definition: API Schema Namespace.

Data Type: Alphanumeric

Format: Min Length: 1

Max Length: 255

Element: <Head>

Definition: **Header of the Message**

Attribute: ver

Definition: Version of the API

This is the API version. NPCI may host multiple versions for supporting gradual migration. As of this specification, default production version is "1.0".

Data Type: Alphanumeric

Format: Min Length: 1 (*length is not checked as version should be "1.0")*

Max Length: 6

Attribute: ts

Definition: Time of request from the creator of the message.

API request time stamp. Since timestamp plays a critical role, it is highly recommended that devices are time synchronized with a time server.

Data Type: ISODateTime

Format: Min Length: 19

Max Length: 19

YYYY-MM-DDThh:mm:ss

(eg 1997-07-16T19:20:30)

where;

YYYY = Four-digit year

MM = Two-digit month (01=January, etc.)

DD = Two-digit day of month (01 through 31)

hh = Two digits of hour (00 through 23) (am/pm NOT allowed)

mm = Ttwo digits of minute (00 through 59)

ss = Two digits of second (00 through 59)

Element: <Request>

Definition: This element contains the Request details and is visible to all parties involved in the transaction processing. This element is populated by the originator of the request and the same must be passed across all the entities.

Attribute: id

Definition: Unique Identifier for the request across all entities.

This will be created by the originator. This will be used to identify each request uniquely across all the entities.

Data Type: Alphanumeric

Format: Min Length: 1

Max Length: 22

Attribute: refUrl

Definition: URL for the transaction

Data Type: Alphanumeric with special characters

Format: Min Length: 1

Max Length: 35

Attribute: type

Definition: This attribute describes the type of the Request

Data Type: Enumeration or Code. Length check is not there as it should be in the list of prescribed types. The allowed values are “STATUS ENQ” or “DETAILS ENQ”

Format: Min Length: NA

Max Length: NA

Element: <Source>

Definition: This element contains the details of the originator of the request and the same must be passed across all the entities.

Attribute: type

Definition: This indicates the routing type to be used. Currently allowed routing type is only Bank short code and it should be always 'CODE'. Length check will not be done as it should be always CODE

Data Type: Alpha

Format: Min Length: NA

Max Length: NA

Attribute: value

Definition: This attribute contains the actual value of the routing type and this value will be used to identify the endpoint URL of the participant which is used to initiate any communication from NPCI

Data Type: Alpha

Format: Min Length: 4

Max Length: 4

Element: <Destination>

Definition: This element contains the details of the originator of the request and the same must be passed across all the entities.

Attribute: type

Definition: This indicates the routing type to be used. Currently allowed routing type is only Bank short code and it should be always 'CODE'. Length check will not be done as it should be always CODE

Data Type: Alpha

Format: Min Length: NA

Max Length: NA

Attribute: value

Definition: This attribute contains the actual value of the routing type and this value will be used to identify the endpoint URL of the participant which is used to initiate any communication from NPCI

Data Type: Alpha

Format: Min Length: 4

Max Length: 4

Attribute: value

Definition: This attribute contains the actual value of the routing type and this value will be used to identify the endpoint URL of the participant which is used to initiate any communication from NPCI

Data Type: Alpha

Format: Min Length: 4

Max Length: 4

Element: <Detail>

Definition: This element contains the parameters of the actual request and the same must be passed to Destination for processing.

Attribute: accNo

Definition: Few of the request is based on this parameter value. And this attribute carries the account number.

Data Type: Alpha

Format: Min Length: 1

Max Length: 35

Attribute: aadhaar

Definition: Few of the request is based on this parameter value. And this attribute carries the Aadhaar number value.

Data Type: Alphanumeric;

Format: Min Length: 12

Max Length: 12

Note: Apart from length and alphanumeric pattern, it will be validated to ensure that it is not begin with “0” or “1” and it is as per verhoeff algorithm.

Attribute: ifsc

Definition: This attribute is also part of the input parameter of request and it carries the bank ifsc code.

Data Type: Alphanumeric; And will be validated against the IFSC pattern

Format: Min Length: 11

Max Length: 11

Element: <AccHolder>

Definition: This element contains the details of the various attributes such as PAN & name of the account holder’s depends upon the requested details.

Attribute: pan

Definition: This attribute is part of the output parameter and will be part of the response and it carries the PAN (Permanent Account Number) value.

Data Type: Alphanumeric; and will be validated against the PAN number pattern

Format: Min Length: 10

Max Length: 10

Attribute: name

Definition: This attribute is also part of the input parameter of request and it carries the PAN (Permanent Account Number) value.

Data Type: Alphanumeric; and will be validated against the PAN number pattern

Format: Min Length: 0

Max Length: 100

Element: <Account>

Definition: This element is used to communicate to the originator about the various attributes of account.

Attribute: status

Definition: This attribute provides the any one of predefined account status.

Data Type: Code

Format: Min Length: NA

Max Length: NA

Sl.No	Value	Code
1	Account is in open and active state	S601
2	Account under litigation	S602
3	A/c inactive (No Transactions for last 3 Months)	S603
4	Dormant A/c (No Transactions for last 6 Months)	S604
5	Account holder expired	S605
6	A/c blocked or frozen	S606
7	Customer insolvent / insane	S607
8	Others	S650

Attribute: type

Definition: This attribute provides the any one of predefined account type.

Data Type: Code

Format: Min Length: NA

Max Length: NA

Sl.No	Value	Code
1	Savings account	T651
2	Current account	T652
3	Cash credit account	T653
4	Overdraft account	T654
5	Others	T699

Element: <Aadhaar>

Definition: This element is used to communicate to the status of the Aadhaar linkage with the given account number.

Attribute: linkStatus

Definition: This attribute provides the any one of predefined values as 'YES' or 'NO'.

Data Type: Code

Format: Min Length: NA

Max Length: NA

Element: <Resp>

Definition: This element contains the information about the Response.

Attribute: ts

Definition: Time of request from the creator of the message.

API request time stamp. Since timestamp plays a critical role, it is highly recommended that devices are time synchronized with a time server.

Data Type: ISODateTime

Format: Min Length: 19

Max Length: 19

YYYY-MM-DDThh:mm:ss

(eg 1997-07-16T19:20:30)

where;

YYYY = Four-digit year

MM = Two-digit month (01=January, etc.)

DD = Two-digit day of month (01 through 31)

hh = Two digits of hour (00 through 23) (am/pm NOT allowed)

mm = Ttwo digits of minute (00 through 59)

ss = Two digits of second (00 through 59)

Attribute: result

Definition: This attribute is used to indicate the end result of the requested message. And it should have the any one of the value from the pre-defined list.

Data Type: Code

Format: Min Length: NA

Max Length: NA

Code	Value
SUCCESS	Request message is Successfully processed
FAILURE	Request has been rejected either by NPCI or DEST

Attribute: rejectedBy

Definition: This attribute is used to indicate the source of rejection in case of failure. And it should have the any one of the value from the pre-defined list.

Data Type: Code

Format: Min Length: NA

Max Length: NA

Code	Value
NPCI	NPCI rejected the message
DEST	Destination system rejected the message

Attribute: errCode

Definition: This attribute is used to indicate the reasons for rejection in case of failure. And it should have the one or many values from the pre-defined list. Multiple error codes will be separated by comma.

Data Type: Code

Format: Min Length: NA

Max Length: NA

3.8 Error Codes

Technical Validation Error Codes

Error Code	Error Description
101	Incorrect Request
102	Source Is Missing
103	Service Is Missing
104	Service Type is Missing
105	Message is Missing
106	Error in Verification - No Signature Tag Found
107	Error in Verification - Incorrect Signature Method Algorithm Used
108	Error in Verification - Incorrect Digest Method
109	Error in Verification - No Matching Certificate Available
110	Error in Verification - Signature is Invalid
111	Type should be either Request or Response
112	Dependency Failed; Unable to publish the Message in Queue
113	Error in DB connectivity
114	Message is not in correct format
115	Head Tag is Mandatory
116	Source Tag is Mandatory
117	Destination Tag is Mandatory
118	Request Tag is Mandatory
119	ReqData Tag is Mandatory
120	Detail Tag is Mandatory
121	NpciRefId Tag is Mandatory
122	Resp Tag is Mandatory
123	RespData Tag is Mandatory
124	Aadhaar Tag is Mandatory
125	AccountHolderList Tag is Mandatory
126	MandateList Tag is Mandatory
127	Mandate Tag is Mandatory
128	AccHolder Tag is Mandatory
129	Account Tag is Mandatory
130	One Or More Attribute is Missing for Head Tag
131	One Or More Attribute is Missing for Source Tag
132	One Or More Attribute is Missing for Destination Tag
133	One Or More Attribute is Missing for Request Tag
134	One Or More Attribute is Missing for Detail Tag
135	One Or More Attribute is Missing for NpciRefID Tag
136	One Or More Attribute is Missing for Resp Tag
137	One Or More Attribute is Missing for Aadhaar Tag
138	One Or More Attribute is Missing for Mandate Tag
139	One Or More Attribute is Missing for AccHolder Tag
140	One Or More Attribute is Missing for Account Tag
141	Attribute Timestamp is Invalid

Error Code	Error Description
142	Attribute Version is Invalid
143	Attribute Code is Invalid
144	Attribute Value is Invalid
145	Attribute Name is Invalid
146	Attribute ID is Invalid
147	Attribute Type is Invalid
148	Attribute RefUrl is Invalid
149	Attribute Result is Invalid
150	Attribute ErrCode is Invalid
151	Attribute RejectedBy is Invalid
152	Attribute UMRN is Invalid
153	Attribute Status is Invalid
154	Attribute EndDate is Invalid
155	Attribute Aadhaar is Invalid
156	Attribute accNo is Invalid
157	Attribute IFSC is Invalid
158	Attribute PAN is Invalid
159	Attribute Type is Invalid
160	Attribute Status is Invalid
161	Attribute linkStatus is Invalid

Business Validation Error Codes

Error Code	Error Description
201	Service Name is Invalid or It Is Not Active
202	Source is NOT valid participant. I.e. Not Available in DB
203	Source is NOT active
204	Source is NOT having privilege for the service
205	Header Timestamp should not be future Timestamp
206	Header Timestamp should not be older than 24 Hours
207	Response Timestamp should not be future time
208	Response Timestamp should not be older than 24 Hours
209	Time difference between Response Received Time and Request Publish Time should not be more than SLA
210	NpciRefID should be valid and should have valid matching record - Invalid NpciRefId
211	Late Resposne; Request is NOT in pending state
212	Source Value Is Not Match with Request Message
213	Destination Value is NOT matching with Request Message
214	Error Code is not part of Defined Destination Reject Reason List
215	Destination is NOT reachable
216	Destination sent Invalid/Incomplete Response
217	Destination Did not send Response with in SLA
218	Message Source is Not matching with the value of Source Tag

Error Code	Error Description
219	Message Source is Not matching with the value of Destination Tag
220	Invalid Destination. I.e. Destination is NOT available In DB
221	Destination is NOT active

API services rejection codes at destination bank

Reason code	Reason description
501	Account closed
502	Invalid account number
503	PAN not updated in CBS
504	Invalid IFSC