NACH-API Specifications

Version 1.2

**Contents**

[1.1 Objectives 3](#_bookmark0)

1. [Communication Channel 4](#_bookmark1)
	1. [Channel Encryption Details 4](#_bookmark2)
	2. [Certificate Format 4](#_bookmark3)
	3. [XML Format Changes for signature 5](#_bookmark4)
	4. [Encoding and decoding of the singed XML message using Base64 6](#_bookmark5)
	5. [Encryption and Signing process 8](#_bookmark6)
2. [API Protocol 8](#_bookmark7)
	1. [Obtaining/Get PAN Number 12](#_bookmark8)
	2. [Validation of an account based on account number and IFSC 15](#_bookmark9)
	3. [Request for an account holders name 18](#_bookmark10)
	4. [Request for Aadhaar Seeding and deactivation 21](#_bookmark11)
	5. [General ACK/NACK Message format 25](#_bookmark12)
	6. [Destination Asynchronous Failure Response Message format 25](#_bookmark13)
	7. [Elements and Attributes Definition 26](#_bookmark14)
3. [Error Codes 37](#_bookmark15)
	1. [Technical Validation Error Codes 37](#_bookmark16)

[1.1 Business Validation Error Codes 40](#_bookmark17)

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.

# 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

1. **Communication Channel**

# 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**

#### Signing 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.

# Certificate Format

#### Certificate formats

* + - 509 certificates v3: (etc.npci.org.in)
			* We need fully qualified domain name certificate from authorised CA.(Main,Intermediate,root)
			* No wildcards in certificate
			* Self-signed certificate not acceptable
		- 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**

# XML Format Changes for signature

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<ach:GetPanDtlsRqst xmlns:ach=["http://npci.org/ach/schema](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="JRxmM3vOzAGVMxCnPKow/0QgRV6uk7URBwsiB9VDwUGh2iEVc4PgV80mhhJfT+7EUT1fs3tNZhg VcIq7y+cQdRudHsfl4v1iRWKna++OyX8YtTp8ZR/yIexVuO5Hv/6vnMrCjPTL3lnTm8HWu3bhagO1i0UwO UeQSFwwd9tUYupnDklKBZpuCDaOarDg6K/OrpgoA42GtJPhVmt+ON3TlFarTs9KOd9xFIDGq1LGr0hA1ZF O7AWLsMOzCf8SJu6pwcleiO9cRpoCEWfRKySribxqUsnx2YJJb6cQOx18DNwaG623/FAcmMnQV3zzq/Jvi SYID8xzGmh1sL266SMasQ==" />

</ReqData>

<NpciRefId value="" />

<Signature [xmlns="h](http://www.w3.org/2000/09/xmldsig)ttp:/[/www.w3.org/2000/09/xmldsig#](http://www.w3.org/2000/09/xmldsig)"><SignedInfo><CanonicalizationMethod Algorith[m="http://w](http://www.w3.org/TR/2001/REC-xml-c14n-20010315)ww.[w3.org/TR/2001/REC-xml-c14n-20010315](http://www.w3.org/TR/2001/REC-xml-c14n-20010315)"/><SignatureMethod Algorith[m="http://w](http://www.w3.org/2001/04/xmldsig-more#rsa-sha256)ww.[w3.org/2001/04/xmldsig-more#rsa-sha256](http://www.w3.org/2001/04/xmldsig-more#rsa-sha256)"/><Reference URI=""><Transforms><Transform Algorith[m="http://w](http://www.w3.org/2000/09/xmldsig#enveloped-)ww.[w3.org/2000/09/xmldsig#enveloped-](http://www.w3.org/2000/09/xmldsig#enveloped-) signature"/></Transforms><DigestMethod Algorith[m="http://w](http://www.w3.org/2001/04/xmlenc#sha256)ww.[w3.org/2001/04/xmlenc#sha256"/](http://www.w3.org/2001/04/xmlenc#sha256)><DigestValue>z55eEtt1DM9jPJ0j rlBS1VRzyETu9S1ek6GDvNuHVzg=</DigestValue></Reference></SignedInfo><SignatureValue

>VoIIdHrtoSe4PFIrvpuGS4yCnoF+SXus4ha8r9HSpCygYyp9nIqP9BWC+Y5kJne8VWd1/m/6TUZG NewpKYm+TVGYoxUJCZw7do3Hws0BRqeq8Cq8U4iBwHm9qcAnIGii2gFd+Kh5F4noaZbsmKpL4XMh

Km/iZ0n7bawUbb3/cn5DT2j6b+RTlpVRb0lLuZhmr18lcDWvdL0bsVCfSovNPhpCrsEIKvzVEd8n dvYkTFbGCW1DvmqZ6VSzUseu2qUlrB3je7CoY5JVZm42p8wt3ciaWZD1HWCcl6C5zRei1M1LkhRJ teSUeHrDtTgHSrU1aOYMXEOTUG5AS5wEOZLx3A==</SignatureValue><KeyInfo><X509Data><X509S ubjectName>CN=cm.npci.org.in,ST=Tamil Nadu,OU=NETC,O=National Payment Corporation of India,C=IN</X509SubjectName><X509Certificate>MIIFmTCCBIGgAwIBAgIKZjLe4Slj1iSCGjANB gkqhkiG9w0BAQsFADCB9zELMAkGA1UEBhMCSU4x RTBDBgNVBAoTPEluc3RpdHV0ZSBmb3IgRGV2ZWxvcG1lbnQgYW5kIFJlc2VhcmNoIGluIEJhbmtp bmcgVGVjaG5vbG9neTEdMBsGA1UECxMUQ2VydGlmeWluZyBBdXRob3JpdHkxDzANBgNVBBETBjUw MDA1NzESMBAGA1UECBMJVGVsYW5nYW5hMSkwJwYDVQQJEyBSb2FkIE5vLjEsIE1hc2FiIFRhbmss IEh5ZGVyYWJhZDEVMBMGA1UEMxMMQ2FzdGxlIEhpbGxzMRswGQYDVQQDExJJRFJCVCBDQSAgU1BM IDIwMTUwHhcNMTgwMjIzMTA0MDU0WhcNMjAwMjIyMTA0MDU0WjB6MQswCQYDVQQGEwJJTjEuMCwG

A1UEChMlTmF0aW9uYWwgUGF5bWVudCBDb3Jwb3JhdGlvbiBvZiBJbmRpYTENMAsGA1UECxMETkVU QzETMBEGA1UECBMKVGFtaWwgTmFkdTEXMBUGA1UEAxMOY20ubnBjaS5vcmcuaW4wggEiMA0GCSqG SIb3DQEBAQUAA4IBDwAwggEKAoIBAQClseI1cQrEadfJhMOROQW9cdDkSceorVgZgkCGryWIfNQO EslkPg9W6B1PeHYUcQ/riHO60nPZNhDFHKKonq6ImuJRCWDGw9xQdhHaLpRdR6v4zl5CwaiWnXw3 FPht3ZKLPOKxKa1z51M+bH7+uX7GdkE67nXLC+oPhzejdiN2w/oORimzQAIHpht0d+SL1UlTvMhZ JHkkYa88Y6BPE19wrBa9tBEXsfDHKYp6z+fx6sWEAisrVeGZJhMNa+kUTp2rwxltSNMeuECBjvAJ rWRHFBC/8Zb9id4M+hnz1BpyoW9mhKuZCMrPXGaGD9a4jU4vtjhs5akvRJWFwGD0xHflAgMBAAGj ggGhMIIBnTAOBgNVHQ8BAf8EBAMCBaAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMIG1

BgNVHSAEga0wgaowgacGBmCCZGQCAzCBnDCBmQYIKwYBBQUHAgIwgYwagYlDbGFzcyAzIGxldmVs IGlzIHJlbGV2YW50IHRvIGVudmlyb25tZW50cyB3aGVyZSB0aHJlYXRzIHRvIGRhdGEgYXJlIGhp Z2ggb3IgdGhlIGNvbnNlcXVlbmNlcyBvZiB0aGUgZmFpbHVyZSBvZiBzZWN1cml0eSBzZXJ2aWNl cyBhcmUgaGlnaDBZBgNVHR8EUjBQMCSgIqAghh5odHRwOi8vMTAuMC42NS42NS9jcmxfMjdDMS5j cmwwKKAmoCSGImh0dHA6Ly9pZHJidGNhLm9yZy5pbi9jcmxfMjdDMS5jcmwwHwYDVR0jBBgwFoAU Tr5+OYyYyj5K1Xrikxgyfa017U0wGQYDVR0RBBIwEIIOY20ubnBjaS5vcmcuaW4wHQYDVR0OBBYE FMHNG+QTRB9c+XjJkU94dxUEjIaLMA0GCSqGSIb3DQEBCwUAA4IBAQB/vVGwJ0NbeeevNWBEbBjE

rn3E1IXfA7CaS1hyrtmhbpTPyblg3zKqATpTYlqfm/ZFNfBtSiukNuZ2H7qDy0i0lpJjhvPhiz1E Cn9hPggPokxdKF3NWxidxZyJZ/UANc3zomjGiq6I/159ocEGa9M0lKG2t/q5E8vntA8up0iLW3Sv d2zmRbnInVs9jGr/zv6ZYXDDvxQ47KQf0Ew3DdB4sysIOyH3BJXbScLS54l0Y7JqfTQ1oaBj1M4T 3qfCQNa3i1dIEhOrYcL8hTYAUjUzIT39C3uwyiWeiNQjOFGxawGoEissPlFhIPoKXM13d+zHTkHU BmkeXAxPGNgNGsTW</X509Certificate></X509Data></KeyInfo></Signature></ach:GetPanDtl sRqst>

# Encoding and decoding of the singed XML message using Base64

### For all asynchronous API services, NPCI API Service system will send the ACK message upon receiving the request and followed by actual response will be sent to the corresponding originator once the processing is over.

### The Signed XML content of input JSON message should be in encoded format using **Base64** encoding. The output/actual response message will sent from NPCI also will be having encoded XML value for the message.

### Note: ACK/NACK will not sent in the encoded format, it will be in the form of type “***application/xml***”

## Encoded format of XML request:

**{“Source”: “U0JJTg==”,**

**“Service”: “QWFkaGFhckxpbmthZ2U=”,**

**“Type”: “UmVxdWVzdA==”,**

**“Message”: “PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+DQo8YWNoO kdldFBhbkR0bHNScXN0IHhtbG5zOmFjaD0iaHR0cDovL25wY2kub3JnL2FjaC9zY2hlbWEvIiA+DQo8SGV hZCB2ZXI9IjEuMCIgdHM9Ijw8SVNPIFRpbWVzdGFtcD4+IiAvPgkNCjxTb3VyY2UgdHlwZT0iQ09ERSIgd mFsdWU9IiIgbmFtZT0iIiAvPg0KPERlc3RpbmF0aW9uIHR5cGU9IkNPREUiIHZhbHVlPSIiIG5hbWU9IiI gLz4NCjxSZXF1ZXN0IGlkPSIiIHR5cGU9IlN0YXR1cyBFbnF8RGV0YWlscyBFbnEiIHJlZlVybD0iIi8+D Qo8UmVxRGF0YT4NCjxEZXRhaWwgYWNjTm89IkpSeG1NM3ZPekFHVk14Q25QS293LzBRZ1JWNnVrN1VSQnd**

**zaUI5VkR3VUdoMmlFVmM0UGdWODBtaGhKZlQrN0VVVDFmczN0TlpoZ1ZjSXE3eStjUWRSdWRIc2ZsNHYxa VJXS25hKytPeVg4WXRUcDhaUi95SWV4VnVPNUh2LzZ2bk1yQ2pQVEwzbG5UbThIV3UzYmhhZ08xaTBVd09 VZVFTRnd3ZDl0VVl1cG5Ea2xLQlpwdUNEYU9hckRnNksvT3JwZ29BNDJHdEpQaFZtdCtPTjNUbEZhclRzO UtPZDl4RklER3ExTEdyMGhBMVpGTzdBV0xzTU96Q2Y4U0p1NnB3Y2xlaU85Y1Jwb0NFV2ZSS3lTcmlieHF Vc254MllKSmI2Y1FPeDE4RE53YUc2MjMvRkFjbU1uUVYzenpxL0p2aVNZSUQ4eHpHbWgxc0wyNjZTTWFzU T09IiAvPg0KPC9SZXFEYXRhPgkNCjxOcGNpUmVmSWQgdmFsdWU9IiIgLz4NCjxTaWduYXR1cmUgeG1sbnM 9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiPjxTaWduZWRJbmZvPjxDYW5vbmljYWxpe mF0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvVFIvMjAwMS9SRUMteG1sLWMxNG4 tMjAwMTAzMTUiLz48U2lnbmF0dXJlTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwM S8wNC94bWxkc2lnLW1vcmUjcnNhLXNoYTI1NiIvPjxSZWZlcmVuY2UgVVJJPSIiPjxUcmFuc2Zvcm1zPjx UcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjZW52ZWxvc GVkLXNpZ25hdHVyZSIvPjwvVHJhbnNmb3Jtcz48RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d 3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PERpZ2VzdFZhbHVlPno1NWVFdHQxRE05alBKM GpybEJTMVZSenlFVHU5UzFlazZHRHZOdUhWemc9PC9EaWdlc3RWYWx1ZT48L1JlZmVyZW5jZT48L1NpZ25 lZEluZm8+PFNpZ25hdHVyZVZhbHVlPlZvSUlkSHJ0b1NlNFBGSXJ2cHVHUzR5Q25vRitTWHVzNGhhOHI5S FNwQ3lnWXlwOW5JcVA5QldDK1k1a0puZThWV2QxL20vNlRVWkcNCk5ld3BLWW0rVFZHWW94VUpDWnc3ZG8 zSHdzMEJScWVxOENxOFU0aUJ3SG05cWNBbklHaWkyZ0ZkK0toNUY0bm9hWmJzbUtwTDRYTWgNCkttL2laM G43YmF3VWJiMy9jbjVEVDJqNmIrUlRscFZSYjBsTHVaaG1yMThsY0RXdmRMMGJzVkNmU292TlBocENyc0V JS3Z6VkVkOG4NCmR2WWtURmJHQ1cxRHZtcVo2VlN6VXNldTJxVWxyQjNqZTdDb1k1SlZabTQycDh3dDNja WFXWkQxSFdDY2w2QzV6UmVpMU0xTGtoUkoNCnRlU1VlSHJEdFRnSFNyVTFhT1lNWEVPVFVHNUFTNXdFT1p MeDNBPT08L1NpZ25hdHVyZVZhbHVlPjxLZXlJbmZvPjxYNTA5RGF0YT48WDUwOVN1YmplY3ROYW1lPkNOP**

**WNtLm5wY2kub3JnLmluLFNUPVRhbWlsIE5hZHUsT1U9TkVUQyxPPU5hdGlvbmFsIFBheW1lbnQgQ29ycG9 yYXRpb24gb2YgSW5kaWEsQz1JTjwvWDUwOVN1YmplY3ROYW1lPjxYNTA5Q2VydGlmaWNhdGU+TUlJRm1UQ 0NCSUdnQXdJQkFnSUtaakxlNFNsajFpU0NHakFOQmdrcWhraUc5dzBCQVFzRkFEQ0I5ekVMTUFrR0ExVUV CaE1DU1U0eA0KUlRCREJnTlZCQW9UUEVsdWMzUnBkSFYwWlNCbWIzSWdSR1YyWld4dmNHMWxiblFnWVc1a 0lGSmxjMlZoY21Ob0lHbHVJRUpoYm10cA0KYm1jZ1ZHVmphRzV2Ykc5bmVURWRNQnNHQTFVRUN4TVVRMlZ 5ZEdsbWVXbHVaeUJCZFhSb2IzSnBkSGt4RHpBTkJnTlZCQkVUQmpVdw0KTURBMU56RVNNQkFHQTFVRUNCT**

**UpWR1ZzWVc1bllXNWhNU2t3SndZRFZRUUpFeUJTYjJGa0lFNXZMakVzSUUxaGMyRmlJRlJoYm1zcw0KSUV oNVpHVnlZV0poWkRFVk1CTUdBMVVFTXhNTVEyRnpkR3hsSUVocGJHeHpNUnN3R1FZRFZRUURFeEpKUkZKQ 1ZDQkRRU0FnVTFCTQ0KSURJd01UVXdIaGNOTVRnd01qSXpNVEEwTURVMFdoY05NakF3TWpJeU1UQTBNRFU wV2pCNk1Rc3dDUVlEVlFRR0V3SkpUakV1TUN3Rw0KQTFVRUNoTWxUbUYwYVc5dVlXd2dVR0Y1YldWdWRDQ kRiM0p3YjNKaGRHbHZiaUJ2WmlCSmJtUnBZVEVOTUFzR0ExVUVDeE1FVGtWVQ0KUXpFVE1CRUdBMVVFQ0J NS1ZHRnRhV3dnVG1Ga2RURVhNQlVHQTFVRUF4TU9ZMjB1Ym5CamFTNXZjbWN1YVc0d2dnRWlNQTBHQ1NxR w0KU0liM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQkFRQ2xzZUkxY1FyRWFkZkpoTU9ST1FXOWNkRGtTY2V**

**vclZnWmdrQ0dyeVdJZk5RTw0KRXNsa1BnOVc2QjFQZUhZVWNRL3JpSE82MG5QWk5oREZIS0tvbnE2SW11S lJDV0RHdzl4UWRoSGFMcFJkUjZ2NHpsNUN3YWlXblh3Mw0KRlBodDNaS0xQT0t4S2ExejUxTStiSDcrdVg 3R2RrRTY3blhMQytvUGh6ZWpkaU4ydy9vT1JpbXpRQUlIcGh0MGQrU0wxVWxUdk1oWg0KSkhra1lhODhZN kJQRTE5d3JCYTl0QkVYc2ZESEtZcDZ6K2Z4NnNXRUFpc3JWZUdaSmhNTmEra1VUcDJyd3hsdFNOTWV1RUN**

**CanZBSg0KcldSSEZCQy84WmI5aWQ0TStobnoxQnB5b1c5bWhLdVpDTXJQWEdhR0Q5YTRqVTR2dGpoczVha 3ZSSldGd0dEMHhIZmxBZ01CQUFHag0KZ2dHaE1JSUJuVEFPQmdOVkhROEJBZjhFQkFNQ0JhQXdIUVlEVlI wbEJCWXdGQVlJS3dZQkJRVUhBd0VHQ0NzR0FRVUZCd01DTUlHMQ0KQmdOVkhTQUVnYTB3Z2Fvd2dhY0dCb**

**UNDWkdRQ0F6Q0JuRENCbVFZSUt3WUJCUVVIQWdJd2dZd2FnWWxEYkdGemN5QXpJR3hsZG1Wcw0KSUdsekl ISmxiR1YyWVc1MElIUnZJR1Z1ZG1seWIyNXRaVzUwY3lCM2FHVnlaU0IwYUhKbFlYUnpJSFJ2SUdSaGRHR WdZWEpsSUdocA0KWjJnZ2IzSWdkR2hsSUdOdmJuTmxjWFZsYm1ObGN5QnZaaUIwYUdVZ1ptRnBiSFZ5WlN CdlppQnpaV04xY21sMGVTQnpaWEoyYVdObA0KY3lCaGNtVWdhR2xuYURCWkJnTlZIUjhFVWpCUU1DU2dJc UFnaGg1b2RIUndPaTh2TVRBdU1DNDJOUzQyTlM5amNteGZNamRETVM1ag0KY213d0tLQW1vQ1NHSW1oMGR**

**IQTZMeTlwWkhKaWRHTmhMbTl5Wnk1cGJpOWpjbXhmTWpkRE1TNWpjbXd3SHdZRFZSMGpCQmd3Rm9BVQ0KV**

**HI1K09ZeVl5ajVLMVhyaWt4Z3lmYTAxN1Uwd0dRWURWUjBSQkJJd0VJSU9ZMjB1Ym5CamFTNXZjbWN1YVc**

**0d0hRWURWUjBPQkJZRQ0KRk1ITkcrUVRSQjljK1hqSmtVOTRkeFVFaklhTE1BMEdDU3FHU0liM0RRRUJDd 1VBQTRJQkFRQi92Vkd3SjBOYmVlZXZOV0JFYkJqRQ0Kcm4zRTFJWGZBN0NhUzFoeXJ0bWhicFRQeWJsZzN**

**6S3FBVHBUWWxxZm0vWkZOZkJ0U2l1a051WjJIN3FEeTBpMGxwSmpodlBoaXoxRQ0KQ245aFBnZ1Bva3hkS 0YzTld4aWR4WnlKWi9VQU5jM3pvbWpHaXE2SS8xNTlvY0VHYTlNMGxLRzJ0L3E1RTh2bnRBOHVwMGlMVzN Tdg0KZDJ6bVJibkluVnM5akdyL3p2NlpZWEREdnhRNDdLUWYwRXczRGRCNHN5c0lPeUgzQkpYYlNjTFM1N GwwWTdKcWZUUTFvYUJqMU00VA0KM3FmQ1FOYTNpMWRJRWhPclljTDhoVFlBVWpVeklUMzlDM3V3eWlXZWl OUWpPRkd4YXdHb0Vpc3NQbEZoSVBvS1hNMTNkK3pIVGtIVQ0KQm1rZVhBeFBHTmdOR3NUVzwvWDUwOUNlc nRpZmljYXRlPjwvWDUwOURhdGE+PC9LZXlJbmZvPjwvU2lnbmF0dXJlPC9hY2g6R2V0UGFuRHRsc1Jxc3Q**

**+”}**

# 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

# 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.

#### URL format: https://<host>/endpointcontextpath

**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). **endpointcontextpath -the end point context path for the API**

#### Common URL format for all the APIs

URL format for all API’s: **https://<host>/apiGatewayListener**

#### Separate URL format for individual APIs:

GetPanDtls - **https://<host>/**getPanDtls GetAccStatus - **https://<host>/getAccStatus** GetAccHolder - **https://<host>/getAccHolder** AadhaarSeeding **- https://<host>/aadhaarSeeding**

##### Banks can choose either common URL format or separate URL format

### Following is the JSON format of request and response messages. **Source** :<< Source Bank Short Name>> // Base64 encoded **Service** : <<Service Name>> // Base64 encoded

### **Type** : <<Request | Response >> // Base64 encoded

### **Message** : << Signed XML message of actual Request or Response >>// Base64 encoded

* 1. **Flow Diagram:**



### Sender bank initiates the JSON request to NPCI system, NPCI performs initial validations and send the XML ACK with http response 200 and the connection will get close with bank later then NPCI proceeds for technical/business validations.

### If technical/business validations are passed then NPCI establishes the connection with receiver bank for forwarding the request to them, when the request has been accepted for processing, receiver bank server **MUST** return a http *202 Accepted* status code to NPCI, then the connection will get close with receiver bank.

### For sending the response to NPCI, receiver bank should establish a new connection. After receiving NPCI will perform the validations for the response if the response passed all the validations receiver bank will receive http 200 status code else XML error notification will be sent.

### NPCI establish the new connection with source bank and send the response, when the response has been accepted by source then bank server **MUST** return a http *202 Accepted* status code to NPCI.

### If receiver bank not provided with the response with in the defined SLA time, the NPCI will construct the failure response and establishes a new connection with source bank and send the failure response, when the response has been accepted by source then bank server **MUST** return a http *202 Accepted* status code to NPCI.

## Note : Receiver bank should provide the response with in the SLA time else the request will be deemed declined by NPCI.

**Example:**

### Source :<< Source Bank Short Name >> Service: GetPanDtls

### Type : Request

Message : <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>

# 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 (Mandatory)

#### 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](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 |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 beidentified | 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 enquiryor 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](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="6afd4578-f021-4321-a908-04b355a758fa" />

<Resp ts="<<ISO Timestamp>>" result="SUCCESS" errCode="S610" 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 codeof 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 ordetails 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 whorejected 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 |

# Validation of an account based on 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 status
		2. Account type

#### Request Message Format

<ach:GetAccStatusRqst xmlns:ac[h="http://n](http://npci.org/ach/schema/)pci[.org/ach/schema/"](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 shortcode 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 shortcode 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 enquiryor 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:ac[h="http://n](http://npci.org/ach/schema/)pci[.org/ach/schema/">](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="6afd4578-f021-4321-k908-04b355a758fa " />

<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 codeof 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 codeof 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 ordetails 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 |

# 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)

#### Request Message Format

<ach:GetAccHolderRqst xmlns:ac[h="http://n](http://npci.org/ach/schema/)pci[.org/ach/schema/"](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 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Message Item** | **<XML Tag>** | **Occurrence** |
| 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 shortcode 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 shortcode 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="S601" rejectedBy=""/>

<RespData>

<AccHolderList>

<AccHolder name="ABCD" />

<AccHolder name="EFGH" />

</AccHolderList>

</RespData>

<NpciRefId value="6afd4578-d021-4321-a908-04b355a758fa " />

</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 codeof 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 codeof 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 ordetails 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 whorejected 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 |

# Request for Aadhaar Seeding and deactivation

Aadhaar Seeding API will be exposed as stateless REST service over HTTPS similar to the other existing API services. I.e. The Aadhaar Seeding API input data should be sent to the existing “apiGateway” URL as JSON content using Content-Type as “text/plain”.

This API is used by Source bank to seed or modify the Aadhaar number to the NPCI mapper database.

Upon receivable of the message from source, NACH –API Cloud system acknowledges the request to source bank by sending either ACCEPTED or ERROR as a result of message authentication by signature verification. After successful validation, NACH – API cloud system passes the message to NACH Aadhaar mapper system to process the request.

Once the request is processed by NACH Aadhaar mapper system, the response message will be sent to NACH API cloud system and which in turn sends the response to the corresponding originator bank.



Responding Entity - **NPCI**

#### Request Message Format

<ach:AadhaarSeedingRqst xmlns:ach="[http://npci.org/ach/schema/"](http://npci.org/ach/schema/) >

<Head ver="1.0" ts="2017-10-16T10:02:00" />

<Source type="CODE" value="SBIN" name="" />

<Request id="123456789" type="AadhaarSeeding" refUrl=""/>

<ReqData>

<Detail recRefNo="123" aadhaar="234567890123" mapStatus="A" mdFlag="Y" mdCustDate="2017-12-28" odFlag="N" odDate="" previousIIN=""/>

</ReqData>

</ach:AadhaarSeedingRqst>

|  |  |  |  |
| --- | --- | --- | --- |
| **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 shortcode 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 | Request Message property | <Request> | 1..1 |
| 4.1.1 | Id of the Request generated by the originator | Id | 1..1 |
| 4.1.2 | Type of the request. Ie indication of status enquiryor details enquiry | Type | 1..1 |
| 4.1.3 | Reference URL for the transaction | refUrl | 0..1 |
| 5.1 | input data related to the request | <ReqData> | 1..1 |
| 5.2 | Details of the Input parameters of the request | <Detail> | 1..1 |
| 5.2.1 | Record Reference number – Unique numberassigned to the request by source system. No significance at NPCI level | recRefNo | 1..1 |
| 5.2.2 | Aadhaar number which needs to seeded ormodified | aadhaar | 1..1 |
| 5.2.3 | Activity of the operation like activate or deactivation of the Aadhaar number | mapStatus | 1..1 |
| 5.2.4 | Mandate flag – indication of mandates submitted ornot at the bank for linking the Aadhaar number with the account number | mdFlag | 1..1 |
| 5.2.5 | Mandate submission date | mdCustDate | 1..1 |
| 5.2.6 | Indication of OD(Over Draft) facility availed or notIs mandatory when OD Date is provided | odFlag | 1..1 |
| 5.2.7 | OD facility availed date | odDate | 0..1 |
| 5.2.8 | IIN of Previous Bank – Earlier Aadhaar linked Bank | previousIIN | 0..1 |

#### Response Message Format ( Success )

<ach:AadhaarSeedingResp xmlns:ach="[http://npci.org/ach/schema/"](http://npci.org/ach/schema/) >

<Head ver="1.0" ts="2017-10-16T10:02:00" />

<Source type="CODE" value="SBIN" name="" />

<Request id="123456789" type="AadhaarSeeding" refUrl=""/>

<NpciRefId value="6afd4578-f021-4321-a908-04b355a758fa"/>

<Resp ts="" result="SUCCESS" errCode="" rejectedBy="" totReqCnt="3" sucessReqCnt="3"/>

<RespData>

<Detail recRefNo="123" aadhaar="517487254455” mapStatus="A” mdFlag="Y"

mdCustDate="2017-12-28" odFlag="N" odDate="" accepted="T" reasonCode="">

</RespData>

</ach:AadhaarSeedingResp>

#### Response Message Format ( Failure )

<ach:AadhaarSeedingResp xmlns:ach="[http://npci.org/ach/schema/"](http://npci.org/ach/schema/) >

<Head ver="1.0" ts="2017-10-16T10:02:00" />

<Source type="CODE" value="SBIN" name="" />

<Request id="123456789" type="AadhaarSeeding" refUrl=""/>

<NpciRefId value="6afd4578-f021-4321-a908-04b355a758fa"/>

<Resp ts="" result="FAILURE" errCode="" rejectedBy="Processor" totReqCnt="5" sucessReqCnt="0"/>

<RespData>

<Detail recRefNo="123" aadhaar="717171570033" mapStatus="" mdFlag="Y" mdCustDate="2017-12-28" odFlag="N" odDate="" accepted="F" reasonCode="32">

</RespData>

</ach:AadhaarSeedingResp>

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Request Message property | <Request> | 1..1 |
| 4.1.1 | Id of the Request generated by the originator | Id | 1..1 |
| 4.1.2 | Type of the request. Ie indication of status enquiry ordetails enquiry | Type | 1..1 |
| 4.1.3 | Reference URL for the transaction | refUrl | 0..1 |
| 5.1 | Unique Identifier assigned by NPCI for the request | <NpciRefId> | 1..1 |
| 5.1.1 | Actual unique value generated by NPCI | value | 1..1 |
| 6.1 | Response of the Message | <Resp> | 1..1 |
| 6.1.1 | Time of response from the sender of the message | ts | 1..1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Message Item** | **<XML Tag>** | **Occurrence** |
| 6.1.2 | Result of the request | result | 1..1 |
| 6.1.3 | Error reason codes for the failure message | errCode | 0..1 |
| 6.1.4 | Actual rejecter of the message. NPCI | rejectedBy | 0..1 |
| 6.1.5 | Total number detail records submitted as part of thisrequest | totReqCnt | 0..1 |
| 6.1.6 | Count of successfully processed detail records | successReqCnt | 0..1 |
| 7.1 | Response Data | <RespData> | 0..1 |
| 7.2 | List of detail response of each record | <Detail> | 0..1 |
| 7.2.1 | ***Input*** Record Reference number – Unique numberassigned to the request by source system. No significance at NPCI level | recRefNo | 1..1 |
| 7.2.2 | ***Input*** Aadhaar number which needs to seeded ormodified | aadhaar | 1..1 |
| 7.2.3 | ***Input*** Activity of the operation like activate or deactivation of the Aadhaar number | mapStatus | 1..1 |
| 7.2.4 | ***Input*** Mandate flag – indication of mandates submittedor not at the bank for linking the Aadhaar number with the account number | mdFlag | 1..1 |
| 7.2.5 | ***Input*** Mandate submission date | mdCustDate | 0..1 |
| 7.2.6 | Indication of OD(Over Draft) facility availed or not –***Input*** | odFlag | 1..1 |
| 7.2.7 | OD facility availed date - ***Input*** | odDate | 0..1 |
| 7.2.8 | Processing status of the request – accepted or not | accepted | 1..1 |
| 7.2.9 | Error reason code in case of failure. Multiple errors willbe separated by comma | reasonCode | 0..1 |

# 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="2017-10-16 10:02:00" 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 whorejected 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

# Destination Asynchronous Failure Response Message format

“Source” : “NPCI” “Service” : “GetPanDtls” “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

# Elements and Attributes Definition

#### Element: Root

**Definition:** XML root element representing each API (GetPanDtls, GetAadhaarLinkage,, 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:** Float

**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;

#### Element: <Request>

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)

**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 field along with source element’s value attribute 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 given below

**Format:** Min Length: NA Max Length: NA

|  |  |
| --- | --- |
| **Purpose code** | **Purpose** |
| CO01 | Customer on-boarding – loan |
| CO02 | Customer on-boarding – Demat |
| CO03 | Customer on-boarding - Brokerage/ SIP/ MF |
| CO04 | Customer on-boarding - Subscriptions |
| CO05 | Customer on-boarding - Others services |
| VO01 | Vendor on-boarding |
| MR01 | Mandate registration - Physical mandate |
| MR02 | Mandate registration - eSign mandate |
| MR03 | Mandate registration - API mandate |
| PP01 | Payment pre-validation – Credit |
| PP02 | Payment pre-validation – Debit |
| PP03 | Payment pre-validation - Refund credit |
| DB01 | DBT beneficiary on-boarding |
| DB02 | DBT payment pre-validation |
| STATUS\_ENQ | - |
| DETAILS\_ENQ | - |

#### 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

#### Attribute: name

**Definition:** This attribute carries the name of the Source.

**Data Type:** Alphabets with special characters like dot, space, hyphen & single quote

**Format:** Min Length: 0 Max Length: 100

#### 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: name

**Definition:** This attribute carries the name of the destination.

**Data Type:** Alphabets with special characters like dot, space, hyphen & single quote

**Format:** Min Length: 0 Max Length: 100

#### 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:** Numeric

**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:** Numeric

**Format:** Min Length: 12 Max Length: 12

**Note:** Apart from length and numeric 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 output parameter and it carries the name of the account holder.

**Data Type:** Alphabets with special characters like dot, space, hyphen & single quote

**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

|  |  |
| --- | --- |
| **Code** | **Value** |
| S601 | Account is in open and active state |
| S602 | Account under litigation |
| S603 | A/c inactive (No Transactions for last 3 Months) |
| S604 | Dormant A/c (No Transactions for last 6 Months) |
| S605 | Account holder expired |
| S606 | A/c blocked or frozen |

|  |  |
| --- | --- |
| S607 | Customer insolvent / insane |
| S608 | Account Closed |
| S609 | No such Account |
| S610 | KYC Documents Pending |
| S611 | Invalid IFSC code |
| S612 | Network failure (CBS) |
| S613 | A/c in Zero balance/No transactions have happened |

#### Attribute: type

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

**Data Type:** Code

**Format:** Min Length: NA Max Length: NA

|  |  |
| --- | --- |
| **Code** | **Value** |
| T651 | Savings account |
| T652 | Current account |
| T653 | Cash credit account |
| T654 | Overdraft account |
| T699 | Others |
| T656 | FD |
| T657 | RD |
| T658 | Loan account |
| T659 | PMJDY account |
| T660 | NRE/NRO account |
| T661 | HUF - Hindu Undivided family |
| T662 | PF & PPF |
| T663 | Basic Savings Bank Deposit (BSBD) |

#### 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;

#### Attribute: result

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)

**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. Or the status code in case of success 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

Error code for Account Holder Name and Get Pan API.

|  |  |  |
| --- | --- | --- |
| **errCode** | **Description** | **Data****Mandatory** |
| S601 | Account is in open and active state | Y |
| S602 | Account under litigation | Y |
| S603 | A/c inactive | Y |
| S604 | Dormant A/c | Y |
| S605 | Account holder expired | Y |
| S606 | A/c blocked or frozen | Y |
| S607 | Customer insolvent / insane | Y |
| S608 | Account Closed | Y |
| S609 | No such Account | N |
| S610 | KYC Documents Pending | N |
| S612 | Network failure (CBS) | N |
| S613 | A/c in Zero balance/No transactions havehappened | Y |
| 900 | PAN Number not linked to Account | N |

Where ever it was given as Y against the each code the detailed data values should be provided in response. For N it is optional

#### Attribute: NpcirefId

**Definition:** This attribute is used to indicate the uniqueness for the each request which will be generated by the NPCI only. In the request this attribute value should be kept blank by source banks and in the response destination bank should provide the exact value which they have received in the request file

**Data Type:** Code

**Format:** Min Length: 1 Max Length: 36

#### Attribute: recRefNo

**Definition:** Few of the request is based on this parameter value. And this attribute carries unique number assigned by Bank for Aadhaar seeding request.

**Data Type:** ALPHANUMERIC

**Format:** Min Length: 1 Max Length: 15

#### Attribute: mapStatus

**Definition:** Few of the request is based on this parameter value. And this attribute carries unique number assigned by Bank for Aadhaar seeding request.

**Data Type:** CODE

**Format:** Min Length: NA Max Length: NA

|  |  |
| --- | --- |
| **Code** | **Value** |
| A | Activate Aadhaar number for an input Bank |
| I | Inactivate the Aadhaar number from the input Bank |
| D | Deseeding of Aadhaar number from the input bank |

**Definition:** This attribute is used to indicate the mandates submitted or not to the Bank by the end customer

**Data Type:** CODE

**Format:** Min Length: NA Max Length: NA

|  |  |
| --- | --- |
| **Code** | **Value** |
| Y | Mandate is submitted |
| N | Mandates are not submitted |

#### Attribute: mdCustDate

**Definition:** This attribute is used to indicate the mandate submission date by End customer for an Aadhaar linking with account number.

**Data Type:** ISODATE – “YYYY-MM-DD”

**Format:** Min Length: 10 Max Length: 10

#### Attribute: odFlag

**Definition:** This attribute is used to indicate whether the end customer is availed the Overdraft facility or not at the requested bank.

**Data Type:** CODE

**Format:** Min Length: NA Max Length: NA

|  |  |
| --- | --- |
| **Code** | **Value** |
| Y | OD is availed |
| N | OD is not availed |

#### Attribute: odDate

**Definition:** This attribute is used to indicate the date at which the end customer availed the overdraft facility at the requested bank.

**Data Type:** ISODATE – “YYYY-MM-DD”

**Format:** Min Length: 10 Max Length: 10

#### Attribute: previousIIN

**Definition:** This attribute is used to indicate earlier Aadhaar linked bank when the new bank tries to re-seed the Aadhaar number for that bank

**Data Type:** Numeric

**Format:** Min Length: 6 Max Length: 6

# 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 |

|  |  |
| --- | --- |
| **Error Code** | **Error Description** |
| 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 |
| 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 |
| 162 | Attribute Value is Invalid |
| 163 | Decryption Failed for Aadhaar field |
| 164 | Decryption Failed for AccountNo field |
| 165 | Decryption Failed for PAN field |
| 166 | Attribute Type of Destination Tag is Invalid |
| 167 | Attribute Value of Destination Tag is Invalid |
| 168 | Attribute Name of Destination Tag is Invalid |
| 169 | Attribute Timestamp of Resp Tag is Invalid |
| 170 | Attribute Name of AccHolder Tag is Invalid |
| 171 | Attribute RecRefNo is Invalid |
| 172 | Attribute mapStatus is invalid |
| 173 | Attribute mandate Flag is invalid |
| 174 | Attribute mandate Cust Date is invalid |
| 175 | Attribute Od Flag is invalid |

|  |  |
| --- | --- |
| **Error Code** | **Error Description** |
| 176 | Attribute Od Date is invalid |
| 177 | Attribute Previous IIN is invalid |
| 178 | Incorrect Message Type. Only Request is Allowed for Internal category |
| 179 | Attribute Aadhaar should be masked |

# 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 |
| 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 |
| 222 | The request is Duplicate. The fields "Source Code Value" and "Request Id" should beunique for each service. Ie The combination "**Source Code Value + Request Id + Service**" is unique |
| 223 | Request Tag of Response Message is not Matching with original request |
| 224 | Total Details Record Count is exceeding the Max Size Defined |
| 225 | Earlier Response |