

**Unique Identification Authority of India**  
Planning Commission, Govt. of India (GoI),  
3rd Floor, Tower II,  
Jeevan Bharati Building,  
Connaught Circus,  
New Delhi 110001



## **AADHAAR OTP REQUEST API**

### **API SPECIFICATION - VERSION 1.0 (DRAFT)**

**DECEMBER 2011**

## Table of Contents

<b>1. INTRODUCTION</b> .....	<b>3</b>
<b>1.1 AUTHENTICATION FACTORS AND ONE TIME PIN (OTP)</b> .....	<b>3</b>
<b>1.2 TARGET AUDIENCE AND PRE-REQUISITES</b> .....	<b>4</b>
<b>1.3 OBJECTIVE OF THIS DOCUMENT</b> .....	<b>4</b>
<b>2. APPLICATION INITIATED OTP REQUEST API</b> .....	<b>5</b>
<b>2.1 OTP REQUEST PROCESS</b> .....	<b>5</b>
<b>2.2 API PROTOCOL</b> .....	<b>6</b>
<b>2.2.1 Element Details</b> .....	<b>6</b>
<b>2.3 OTP REQUEST API: INPUT DATA FORMAT</b> .....	<b>7</b>
<b>2.3.1 Element Details</b> .....	<b>7</b>
<b>2.4 OTP REQUEST API: RESPONSE DATA FORMAT</b> .....	<b>8</b>
<b>2.4.1 Element Details</b> .....	<b>9</b>
<b>3. APPENDIX</b> .....	<b>10</b>
<b>3.1 RELATED PUBLICATIONS</b> .....	<b>10</b>

# 1. Introduction

---

The Unique Identification Authority of India (UIDAI) has been created, with the mandate of providing a Unique Identity (Aadhaar) to all Indian residents. The UIDAI proposes to provide online authentication using demographic and biometric data.

Aadhaar “*authentication*” means the process wherein Aadhaar Number, along with other attributes, including biometrics, are submitted to the Central Identities Data Repository (CIDR) for its verification on the basis of information or data or documents available with it. UIDAI will provide an online service to support this process. Aadhaar authentication service only responds with a “yes/no” and no personal identity information is returned as part of the response.

## 1.1 Authentication Factors and One Time Pin (OTP)

Authentication focuses on matching a person’s identity based on the reliability of a credential offered. Various agencies have different requirements for the degree of assurance required while authenticating beneficiaries/customers. When authenticating a resident, multiple factors may be used to strengthen the authenticity of the request.

In general, following are the 3 categories of factors:

1. ***What you have***: Something the user uniquely has (e.g., a card, security token, mobile phone, tablet/laptop computer accessing email, etc.)
2. ***What you know***: Something the user knows that is not public (e.g., a password, PIN, secret question, etc.). Some of the demographic details such as date of birth may also be classified in this although they are considered a weak factor.
3. ***Who you are***: Something the user individually is or does (e.g., fingerprint, iris pattern, signature, handwriting, etc.).

Aadhaar authentication provides multi-factor authentication based on the following factors:

1. *What you have* – Mobile phone / Email
2. *What you know* – 6 digit PIN, currently offered only for direct UIDAI transactions
3. *Who you are* – Fingerprints and Iris

When user agencies adopt Aadhaar authentication, they can choose option 1 or 3 or both. Resident authentication can be strengthened by verifying the possession of the mobile by resident. One Time Pin (OTP) is a mechanism to do achieve this. Aadhaar authentication supports OTP and can be used by user agencies.

In a nutshell, OTP request can be initiated by the resident by calling IVR or sending SMS or the request can be initiated by the application on behalf of the resident. This document covers the API details for “*application initiated*” OTP request. Notice that OTP

is always delivered on the resident's mobile and application is expected to capture that during authentication so that OTP can also be validated along with authentication.

## 1.2 Target Audience and Pre-Requisites

This is a technical document and is targeted at software professionals working in technology domain and interested in incorporating Aadhaar authentication into their applications. Readers must be fully familiar with Aadhaar Authentication API 1.5 [http://uidai.gov.in/images/FrontPageUpdates/aadhaar\\_authentication\\_api\\_1\\_5\\_rev1\\_1.pdf](http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_5_rev1_1.pdf) before reading this document.

## 1.3 Objective of this document

This document provides Aadhaar OTP Request API (Application Programming Interface) specification for applications to request OTP on behalf of the resident. It contains details including API data format, protocol, and security specifications.

## 2. Application Initiated OTP Request API

---

This chapter describes the AUA application initiated OTP request API in detail including the flow, communication protocol, and data formats.

### 2.1 OTP Request Process

Application initiated OTP request works as follows:

1. Application (an application on an assisted device or self-service kiosks, or applications on the Internet), wanting to use Aadhaar OTP as a factor within Aadhaar authentication, initiates the transaction flow
2. Application captures Aadhaar number along with other attributes (name, address, biometric, etc.) as needed by the application
3. Application, through AUA server, invokes the OTP API by forming digitally signed API Input XML (format explained later in this document)
4. UIDAI server processes the input, validates it, generates OTP, and sends it to resident's registered mobile and email (based on Aadhaar data in UIDAI server)
5. UIDAI server then responds with an XML with success or indicating any error (see OTP response XML explained later in this document)
6. Resident receives the OTP from Aadhaar server on his/her email and/or mobile.
7. Application then requests resident to enter OTP that was into the application screen so that application can send that data for Aadhaar authentication

Rest of the flow is same as authentication flow explained in authentication document. If OTP has not expired, and if other factors are matched, authentication responds with a "yes/no" with appropriate return code. Notice that OTP expires with a UIDAI stipulated time. OTP message sent to resident provides time at which OTP was generated and when it is going to expire.

Since OTP is always sent to both mobile and email (based on what resident has provided to UIDAI), resident may use any of the mechanism to receive OTP and use it while authentication.



AUA Application should not initiate OTP request without the resident having to explicitly request the service offered by AUA and also should make sure that resident is made aware of OTP usage so that no OTP is sent to resident without his/her knowledge.

## 2.2 API Protocol

Aadhaar OTP Request service (or simply OTP service) is exposed as stateless service over HTTPS. Usage of open data format in XML and widely used protocol such as HTTP allows easy adoption and deployment of these services.

Following is the URL format for Aadhaar OTP service:

```
https://<host>/otp/<ver>/<ac>/<uid[0]>/<uid[1]>/
```

API input data should be sent to this URL as XML document using Content-Type “application/xml” or “text/xml”.



For all SSL communication the agencies should automatically validate the Aadhaar certificates. Also on every SSL handshake agencies should validate the revocation list online.

### 2.2.1 Element Details

**host** – Aadhaar OTP server address. Actual production server address will be provided to ASAs and AUAs. Note that production servers can only be accessed through secure leased lines. For development and testing purposes, public URL “*auth.uidai.gov.in*” can be used. Application at AUA/ASA server should ensure that actual URL is configurable.

**Next part of the URL “otp” indicates that this is a OTP Request API call instead of regular authentication API call. Ensure that this is provided.**

**ver** – OTP API version (optional). If not provided, URL points to current version. UIDAI may host multiple versions for supporting gradual migration. As of this specification, only valid version for production use is “1.0”.

**ac** – A unique code for the AUA which is assigned by UIDAI. This is an alpha-numeric string having maximum length 10. (A default value “public” is available for testing.)

**uid[0]** and **uid[1]** – First 2 digits of Aadhaar Number. Used for load-balancing.

For all valid responses, HTTP response code 200 is used. All application error codes are encapsulated in response XML element. In the case of connection and other server errors, standard HTTP error response codes are used (4xx codes such as 403, 404, etc.). HTTP automatic redirects also should be handled by ASA/AUA server.



ASA server must set REMOTE\_ADDR header of HTTP with ASA server's static IP address. Authentication sub-system system allows only white listed IP addresses coming through a secure private network.

## 2.3 OTP Request API: Input Data Format

Aadhaar OTP service uses XML as the data format for input and output. To avoid sending unnecessary data, do not pass any optional attribute or element unless its value is different from default value. Any bad data or extra data will be rejected.

Following is the XML data format for OTP API:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Otp uid="" tid="" ac="" sa="" ver="" txn="" lk="">
  <Opts ch=""/>
  <Signature>Digital signature of AUA</Signature>
</Otp>
```

### 2.3.1 Element Details

**Element: Otp** (mandatory)

- Root element of the input XML for OTP service.

**Attributes:**

- **uid** – (mandatory) Aadhaar Number of the resident
- **tid** – (mandatory) For Registered devices, send its unique Terminal ID. For Public devices, value should be passed as “public”.
- **ac** – (mandatory) A unique code for the AUA which is assigned by UIDAI during AUA registration process. This is an alpha-numeric string having maximum length 10. (A Default value “public” is available only for testing.)
- **sa** – (mandatory) A unique “Sub-AUA” code. AUAs are expected to manage these codes within their system and ensure uniqueness within their system. This allows auditing and business intelligence to be provided at SA level. If AUA and SA are same agency, use value of “ac” for this attribute. This is an alpha-numeric string having maximum length 10.
- **ver** – (mandatory) version of the API. Currently only valid value is “1.0”.
- **txn** – (optional) AUA specific transaction identifier. AUA can choose to pass this as part of input. This is returned as part of response as is. This is very useful for linking transactions full round trip across systems. This is an alpha-numeric string of maximum length 50. Only supported characters are A-Z, a-z, 0-9, period, comma, hyphen, backward & forward slash, left & right parenthesis, and colon. No other characters are supported. It is highly recommended that AUAs use this attribute for correlating requests with responses for auditing and verification.
- **lk** – (mandatory) A valid “License Key” assigned to the AUA. Administration portal of UIDAI will provide a mechanism for AUA administrator to generate

license keys and use it within the authentication. These license keys have expiry built into them and AUA administrator need to ensure that they generate new license keys before current ones expires through self-service portal. This is an alpha-numeric string of length up to 64 characters.

**Note:** You can use any valid authentication license key that has OTP feature enabled for this purpose.

**Element: Opts** (Optional)

- Various options are provided under this element.

**Attributes:**

- **ch** – (Optional) Channel through which OTP should be sent. Possible values are as follows:
  - “00” – send OTP via both SMS and Email (this is the default)
  - “01” – send OTP via SMS only
  - “02” – send OTP via Email only

**Element: Signature** (mandatory)

- In order to ensure that OTP XML was not tampered during its transit between AUA servers and UIDAI Auth Servers, it is required that OTP XML be digitally signed by AUA. X.509 certificate used by the AUA for the purposes of digital signature should have been issued by a well-know Certification Authority, and should be included in <Signature> of Auth request.

OTP API server will validate each request by verifying the digital signature and the Certification Authority of the AUA's X.509 present in the digital signature. In addition, server will validate that one of the “O” elements in the X.509 certificate's subject contains the AUA's organization name as stored in UIDAI database.

## 2.4 OTP Request API: Response Data Format

Response XML is as follows:

```
<OtpRes code="" txn="" err="" ts="">
<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/2000/09/xmldsig#rsa-sha256"/>
    <Reference URI="">
      <Transforms>
        <Transform
          Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha256"/>
      <DigestValue></DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue></SignatureValue>
  <KeyInfo></KeyInfo>
</Signature>
</OtpRes>
```

```
</Reference>
</SignedInfo>
<SignatureValue></SignatureValue>
</Signature>
</OtpRes>
```

### 2.4.1 Element Details

#### *Element: OtpRes*

##### *Attributes:*

- **code** – unique alphanumeric “OTP response” code having maximum length 40.
- **txn** – AUA specific transaction identifier. This is exactly the same value that is sent within the request.
- **ts** – Timestamp when the response is generated. This is of type XSD dateTime.
- **err** – Failure error code. If OTP request fails, this attribute provides any of the following codes:
  - “110” – Aadhaar number does not have verified mobile/email
  - “510” – Invalid “Otp” XML format
  - “520” – Invalid device
  - “530” – Invalid AUA code
  - “540” – Invalid version
  - “565” – License key has expired
  - “566” – Invalid license key
  - “569” – Digital signature verification failed
  - “570” – Invalid key info in digital signature (this means that certificate used for signing the OTP request is not valid – it is either expired, or does not belong to the AUA or is not created by a CA)
  - “940” – Unauthorized ASA channel
  - “941” – Unspecified ASA channel
  - “950” – Could not generate and send OTP
  - “951” – SMS Server not available
  - “952” – Email Server not available
  - “999” – Unknown error

#### *Element: Signature*

- This is the root element of digital signature. This is same as what is used for Authentication API.
- For more details, refer: <http://www.w3.org/TR/xmlsig-core/>

## 3. Appendix

---

### 3.1 Related Publications

The following standards are applicable and related to the information in this document.

Aadhaar Authentication API 1.5 (Rev 1)	<a href="http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_5_rev1_1.pdf">http://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_1_5_rev1_1.pdf</a>
XML Signature	<a href="http://www.w3.org/TR/xmlsig-core/">http://www.w3.org/TR/xmlsig-core/</a>