



ESign API GATEWAY DOC

Table of Contents

1. INTRODUCTION	2
2. API CONNECT COMPONENTS	3
3. API AUTHENTICATION	3
4. API INTERNAL DESIGN	3
5. STEPS TO ACCESS & SUBSCRIBE IN PORTAL	8

1.1 Design Document Purpose

The purpose of this document is to provide a detailed specification of the **ESign** in sufficient depth to:

- Enable the component to be built and tested.
- Ensure that it can be enhanced, supported and maintained by other areas of the organization after initial implementation.

1.2 Design Reviews

The service design will be reviewed within Middleware team and security testing team as needed. Once finalized, design resource will provide an overview to other teams such as front end application development team and various project resources.

2. API CONNECT COMPONENTS

- API Connect is used to expose the service to front end applications.
- Within API Connect, there are multiple Products. The **ESign** API is exposed within **ESign Product**.
- Service Consumers must subscribe to the API. A unique application Identification (client-id) and a secret will be generated.
- The assigned Client-id and Client-Secret must be supplied in the header for each API.
- URLs for invoking the services can be found in API Connect Developer portal and also mentioned in the below section.

3. API AUTHENTICATION

JWT Access token to be passed in header section of Authorization field. The Token can be generated by subscribing to **TOKEN API**.

3.1 TOKEN API

- **Overview:** To retrieve access token.
- **Request Type:** POST
- **Mandatory HTTP Headers:**
 - X-IBM-Client-Id: xxxxxxxxxxxxxxxx
 - X-IBM-Client-Secret: xxxxxxxxxxxxxxxx
- **SIT URL:** <https://apiuat.ujjivansfb.in/ujjivan/development/v1/tokens>

4. API INTERNAL DESIGN

- **Overview:**

Service is designed to electronically sign the document. The API Gateway makes a backend **Finahub Esign Server**.
- **Request Type:** POST
- **API URL:**

UAT :
UploadDocument
<https://apiuat.ujjivansfb.in/ujjivan/development/v1/ESignServer/uploaddoc>

GetEsignXML
<https://apiuat.ujjivansfb.in/ujjivan/development/v1/ESignServer/getesignxml>

PostEsignXML

<https://apiuat.ujjivansfb.in/ujjivan/development/v1/ESignServer/nsdlpostxml>

DownloadDoc

<https://apiuat.ujjivansfb.in/ujjivan/development/v1/ESignServer/downloaddoc>

PROD :

UploadDocument

<https://api.ujjivansfb.in/ujjivan/development/v1/ESignServer/uploaddoc>

GetEsignXML

<https://api.ujjivansfb.in/ujjivan/development/v1/ESignServer/getesignxml>

PostEsignXML

<https://api.ujjivansfb.in/ujjivan/development/v1/ESignServer/nsdlpostxml>

DownloadDoc

<https://api.ujjivansfb.in/ujjivan/development/v1/ESignServer/downloaddoc>

➤ **Mandatory HTTP Headers:**

- X-IBM-Client-Id: xxxxxxxxxxxxxxxxx
- X-IBM-Client-Secret: xxxxxxxxxxxxxxxxx
- Authorization: <Access token to be passed, generated from TOKEN API.>

➤ **Input&Output Parameters:**

A) **UploadDocument**

➤ **Minimum Required Input Parameter:**

Element Name	M/O/C	Data Type	Description
PdfBase64	M	String	Base64 converted version of the pdf document to be signed.
securityToken	M	String	A unique security token to be used.
clientId	M	String	Registered client id from our database client table

➤ **Output Parameters:**

Element Name	M/O/C	Data Type	Description
Status	M	String	Display 'success' or 'failed'
Details	M	String	Display 'success' or 'failed'
ErrorCode	M	String	Populate value in case of error else leave empty string.

DocRepold	M	String	Document Id will be returned which need to pass in the next section esigned xml.
------------------	---	--------	--

B) Get esigned XML:

➤ **Minimum Required Input Parameter:**

Element Name	M/O/C	Data Type	Description
reason	M	String	Reason text String for the Signature. Eg: Official
securityToken	M	String	A unique security token to be used.
clientId	M	String	Registered client id from our database client table
redirectUrl	M	String	Url to which nsdl should redirect after completing Esign Process. Leave empty in case request is from mobile app.
coordinates	M	String	Location of visible signature on page Eg: "36, 748, 144, 780
dateString	M	String	Current datetime in format ("yyyy-MM-dd'T'HH:mm:ss")
version	M	String	Pass value "2.0" for now
signerName	M	String	Signer name text
bioType	M	String	Pass value "IRIS" for Iris based ESign and pass value "FINGERPRINT" for finger print based ESign and pass value "OTP" for otp based ESign
docRepold	M	String	Document id returned by the system when the original document is uploaded.
pageNo	M	String	Page number of Pdf where the visible signature needs to be placed. Value "firstPage" will add the signature to the first page. And the value "lastPage" will add signature to the last page of the pdf document. If you specify "all", the real signature will be placed on the first page and all other pages will have signature look like representation.
userConsent	M	String	Pass "Y" to indicate , user has given consent for ESign. If else pass "N"
standardSignaturePosition	M	String	Instead of exact co-ordinates api also supports standard positions. Possible values of signature positions are "topleft","topright","bottomleft","bottomright". Coordinates have preference over this.
location	M	String	Location text String of the Signature Eg: Mumbai
signatureText	M	String	Signature custom text to be shown instead of name. Leave this field empty to have a Signature text auto created using name.
signOutputType	M	String	Pass value "normal" for regular pdf signature. For getting raw Signature data for mandate signing , use value "rawSignature"

➤ **Output Parameter:**

Element Name	M/O/C	Data Type	Description
sigName	M	String	Signature Name
transactionCode	M	String	Represents transaction code.
Xml	M	String	XML data with all details
DocFilePath	M	String	Contains the document file path.

C) PostEsignXML:

➤ **Minimum Required Input Parameter:**

Element Name	M/O/C	Data Type	Description
file	M	String	Pass Base64 encoded absolute path of pdf file with empty Signature. In case of raw signature mode leave empty.You have got as response from earlier API call
securityToken	M	String	A unique security token to be used.
signOutputType	M	String	Pass value "normal" for regular pdf signature. For getting raw Signature data for mandate signing , use value "rawSignature"
xml	M	String	Pass Base64 encoded original response xml you got from ESP
signFieldName	M	String	The Signature field name, you got as response from earlier API call
transactionCode	M	String	The Transaction Id of the ESign transaction, which is available as response to previous API call
ssoUserId	M	String	Only used in case SSO, By default leave empty.

➤ **Output Parameter:**

Element Name	M/O/C	Data Type	Description
Status	M	String	Display "success" or "failed"
country	M	String	Display country name
gender	M	String	Gender type
pkcs7Content	M	String	Content

lastFourDigitOfAadhaar	M	String	Display last four digit of aadhar number.
nameInCertificate	M	String	Name as in certificate.
photoHash	M	String	Hashed photo content
DocRepold	M	String	Doc rEpo id
TransactionId	M	String	Transaction id
Details	M	String	
Xml	M	String	
userX509Certificate	M	String	
state	M	String	Display state name
ErrorCode	M	String	Display a code in case of error else empty string.
yearOfBirth	M	String	Year of birth.

D) Document Download:

➤ Minimum Required Input Parameter:

Element Name	M/O/C	Data Type	Description
docRepold	M	String	Document id of the document to be downloaded
securityToken	M	String	A unique security token to be used.
clientId	M	String	Registered client id from our database client table
downloadType	M	String	Valid values normal and base64 Passing value normal will initiate a browser file download. Passing base64 will return the files base64 value as part of return JSON response.

➤ Output Parameter:

Element Name	M/O/C	Data Type	Description
Status	M	String	Display "success" or "failed"
docRepold	M	String	Document id of the document downloaded
base64Doc	M	String	Downloaded document in encrypted format.

➤ **Error Handling**

HTTP Status Code	HTTP Message	More Information
401	Unauthorized	Access token missing or validation
400	Bad Request	The parameters are invalid or missing.
503	Service Unavailable	The parameters were valid but the request failed.
200	Success	

5. STEPS TO ACCESS & SUBSCRIBE IN PORTAL

Refer Subscription User manual shared during initial on board.

-----**END OF DOCUMENT**-----