INTEGRATION DOCUMENTATION

Of

E-SIGN GATEWAY

Powered by,

Nebula Cloud Infra Services Limited.

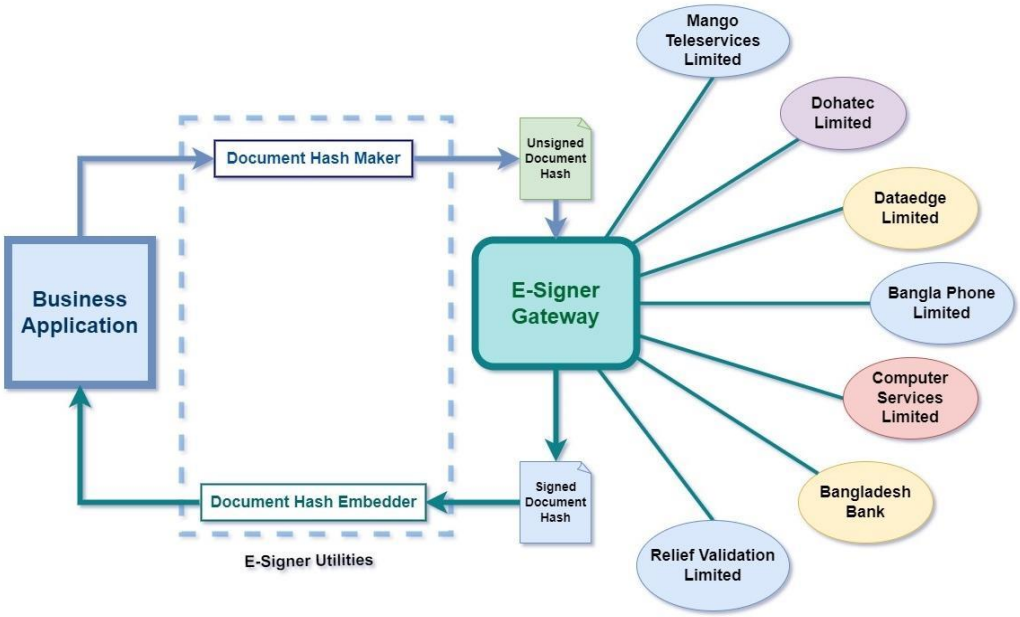
Table of Contents

Overview

E-Signer Gateway is a bridge to Certifying Authority that simplifies document E-Signing by enabling direct communication with their E-Signer Platform, making it more efficient and secure.

1. How E-Sign Gateway Works

1. From the Business application, a user has to choose the **document** to sign.
2. This document has to be **hashed** using a **self-hosted File Hasher software** utility in the Business Application end.
3. Send the **document hash** to the E-sign Gateway.
4. Gateway will facilitate users to **choose a Certifying Authority**.
5. E-Sign Gateway works on the process of sending the unsigned hash to the chosen Certifying Authority's (CA) E-Signer Platform.
6. CA E-Signer Platform sends back the signed hash to the Business application through E-sign Gateway.
7. Business application sends the **signed hash** to the same self-hosted File Hasher software, to embed the signed hash in the original document.
8. The document will be available for download as an electronically and legally signed document in the Business Application End.



2. How to Integrate?

2.1 Choose Plan

Explore our pricing plans and discover the features and benefits that is best for your business application. Click on Get Started button on the chosen plan to Register for a plan or you can start a free trial.

2.2 Register

When you click on the Free Trial button or Get Started button you will be requested to register. Input details about your business application and register for the selected plan. You will receive a secret token according to the chosen plan which you will have to use while integrating E-Sign Gateway.

2.3 Setup API

It is assumed that your business application has a file to sign. Users from business applications have to have at least one digital certificate to sign the file. So, at first business applications need to fetch digital certificates for logged in users.

>>Fetch User Certificate [Ref. 2.3.1]

>>Hashing the Document to be signed [Ref. 2.3.2]

>>Request for Signing the Hash [Ref. 2.3.3]

>>Embed the Signed Hahed into the original Document [Ref. 2.3.4]

2.3.1 User Certificate Fetching API

Business application will fetch corresponding certificate for logged in user by accessing the following API

API: https://gateway.e-signer.xyz/certificate-authority?data=<base64encoded(JSON_encode(parameter))>

API TYPE: GET

Parameters:

Parameter	Value	Description
mobile	01XXXXXXXXX	11 Digit Mobile Number of User
return_url	Returning URL	The URL where system will return after fetching the certificate.

Response:

The Response of this API will be json encoded and base64 encoded.

To get the actual response of the API, first, you will have to json decode the response and then base64 decode the result.
Then you will get the response data as follows:

Variable	Value	Description
ca_type	Integer value	An integer value that implies for which Certifying Authority the request was. 1= Mango CA, 2= BCC CA
status	success error	API response status that implies if the request was successful or not.
certificate (array)	[certificate=> <value>, base64_certificate=> <value>]	An Array that comes with two types of certificate value. Certificate can be both base64 encoded and not base64 encoded. Normally received certificate can be found in 'certificate' named element of the array and base64 encoded certificate will be found in 'base64_certificate' named element of the array.

Note: It is recommended that user's digital certificates are saved in the business application database if the user has more than one certificate. Now the next step is File Hashing before sending the file to E-Sign Gateway. File Hashing is a part of E-Signer Utility.

2.3.2 E-Signer Utility API

Before sending the file to E-Sign Gateway, the file needs to be hashed. Hence, we offer file hasher in E-Signer Utilities. If a user has more than one active certificate. For each certificate, the File Hashing Utility API will be called. You will have to download the File Hashing E-Signer Utilities from this link …….

https://e-signer.xyz/download-file-hashing-utility

After downloading the source code, you will have to host the source code of File Hashing E-Signer Utility in your preferred hosting environment.

File Hasher is a part of E-Signer Utility that generates a unique identifier, known as a hash value or hash code, from a given file or document.

Each Certifying authority accepts different methods of File hashing. To achieve the common file hashing method, we have arranged more than one API that have to be called sequentially. Let's have a look to File Hashing E-Signer Utility APIs

Common File Hasher API: <url_of_esigner_utility>/sign-embed/upload-doc

Common File Hasher API Type: POST

Common File Hasher API Form Data:

Parameter	Value	Description
APIKEY	997dc568-5d89-4147-a48f-186185b66bc7	Unique key for communication between two distinguished systems, which is constant.
pdf_doc	URL File object	An Object created from file location.
APISECKEY	887dc568-5d89-4147-a48f-186185b66bc7-997dc568-5d89-4147-a48f-186185b66bc9	This key is a secret key for ensuring secure API access which should be changed in a regular interval.
doc_name	File Original Name	The actual file name which is supposed to be hashed.

Common File Hasher API Response:

Calling this API returns a JSON response and the following keys will be found in the response:

Variable	Value	Description
name_hash		Hash of document's original name
doc_uuid		Universally Unique Identifier of the document.
uuid_hash		The hash of the Universally Unique Identifier of the document.

downloadUrl		Download URL for downloading the Document

Additional File Hasher API: < url_of_esigner_utility >/bcc-ca/file-hash

Additional File Hasher API Type: POST

Additional File Hasher API Form Data:

Data	Value	Description
docUuid	doc_uuid	doc_uuid from first response
signerId	01xxxxxxxxx	Logged In user's phone number
reason	Any Text	Reason for calling this API
location	Any Location	Specify a Location
certBase64	<base64 Code>	Base64 encoded certificate that was fetched with the API that is documented in section 2.3.1.

Additional File Hasher API Response:

Calling this API returns a JSON response and the following key will be found in the response:

Variable	Value	Description
fileHash		The hash value of the file

2.3.3 E-Sign Gateway API:

As the fileHash value achieved from the Additional File Hasher API, Business App will now call the E-Signer Gateway API.

E-Sign Gateway API:
https://gateway.e-signer.xyz/signing?data=<base64_encoded_parameter>
E-Sign Gateway API Type: GET
E-Sign Gateway API Parameters:

Note: The parameters must be a multidimensional array and the array must be base64 encoded. For each CA an array has to be set up and at the end, all the arrays should be pushed in a data array.

Array For BCC CA: ('bcc')

Array Elements	Value	Description
caId	bcc	The unique ID of BCC to identify BCC was chosen in E-Sign Gateway
document_hash	fileHash	For BCC: The response 'fileHash' from Additional API in Section 2.3.2
document_name	fileOriginalName	Original file name from business application
office_token	Hexadecimal value	User's Office Token value that came with User's Digital Certificate
certificate_token	Hexadecimal Value	User's Digital Certificate provided from BCC CA
signer_id	Any text	Unique Signer ID provided from CA
mobile	01XXXXXXXXX	User's mobile number that is associated with BCC CA account
return_url	/dashboard/sign-status	API of Signed File Hash Embedder Tool that will be called after E-Signer Gateway returns the Signed File from BCC CA.

Array For Mango CA: ('mango_ca')

Array Elements	Value	Description
caId	mango_ca	The unique ID of Mango CA to identify that Mango CA was chosen in E-Sign Gateway
document_hash	uuid_hash	For Mango CA: The response 'uuid_hash' from the Common API in Section 2.3.2
document_name	fileOriginalName	Original file name from business application

certificate_token	Hexadecimal Value	User's Digital Certificate provided from Mango CA
signer_id	Any text	Unique Signer ID provided from Mango CA
return_url	/dashboard/sign-status	API of Signed File Hash Embedder Tool that will be called after E-Signer Gateway returns the Signed File from Mango CA.

For your convenience, please have a look at this example of code:

```
$data = [
     'bcc' => [
           'caId' => 'bcc',
           'document_hash' => $file->file_hash,
           'document_name' => $file->file_name_original,
           'office_token' => 'e797d696-e614-4db9-ae79-2abd2d14c504',
           'certificate_token' => '',
           'signer_id' => $bcc_signer_id,
           'mobile' => auth()->user()->phone_number,
           'return_url' => url('dashboard/sign-status'),
      ],
      'mango_ca' => [
           'caId' => 'mango_ca',
           'document_hash' => $file->uuid_hash,
           'document_name' => $file->file_name_original,
           'certificate_token' => $mango_app_token ?? '',
           'signer_id' => $mango_signer_id,
           'mobile' => auth()->user()->phone_number,
           'return_url' => url('dashboard/sign-status'),
       ],
    ];
$encodedData=base64_encode(json_encode($data));
```

2.3.4 Signed File Hash Embedder API:

When the file is signed and the newly signed file hash returns from the E-Signer gateway, another API will be called from the e-signer gateway that will work on converting the signed file hash to its original form. **Signed File Hash Embedder is a** software tool that involves incorporating the hash value of a document directly into the document itself.

Assuming the API of this tool has been already set up as you have downloaded and hosted the source code of File Hashing E-Signer Utilities. Signed File Hash Embedder is a part of the utilities. You have already sent the URL of this Signed File Hash Embedder

tool as you have sent the URL of this tool as parameter data when you called the E-Signer Gateway API in section 2.3.3.

Now the signed file hash needs to return to its original form. To achieve the form-data of Signed File Hash Embedder Utility's API you need to form unique data sets for different Certifying Authority.

The E-Signer gateway returns some data as base64 encoded and the base64 data will be JSON encoded. After base64 and JSON decoding the returned data you will get the following data:

Data	Value	Description
signature	Hash value	The Hash value of signature returning from E-Signer Gateway
ca_type	Integer Value	1= Mango 2= BCC
signed_object	Hash value	The Hash value of file with sign returning from E-Signer Gateway
nid	13 digit NID	NID of User.
app_token	Base64 encoded token	The token that was given during registration in E-Sign Gatway.

Your business application already has the following data in application's database as mentioned earlier in document:

Data	Value	Description
doc_uuid	Hash value	Saved value when Common File Hashing API in Section 2.3.2 was called
file_hash	Hash value	Saved value when Additional API in Section 2.3.2 was called
certBase64	Base64 encoded digital certificate	User's Digital Certificate of corresponding CA

Now you need to set up form-data for Signed File Hash Embedder API. Each CA receives the signed file hash differently. So, you have to set up the API differently for each CA.

Signed File Hash Embedder API of BCC CA: < url_of_esigner_utility >/bcc-ca/apply-signed-hash

Signed File Hash Embedder API of Mango CA: < url_of_esigner_utility >/sign-embed/apply-signed-hash

API Type: POST

API Form Data for BCC CA:

Data	Value	Description
signerId	User's Signer ID	In general, it is user's mobile number
docUuid	docUuid	This value was previously saved in business application database while sending the value to E-Signer Gateway
signedHash	Hash value	The Sign Hash that is returning from E-Signer Gateway This Value must be used as form-data if the CA is Mango or BCC.
certBase64	User's Digital Certificate	User's Digital Certificate for Corresponding CA. Only required if the CA is BCC.
fileHash	Hash value	This value was previously saved in the business application database while sending the value to the E-Sign Gateway. fileHash must be used as form-data If the CA is BCC

API Form Data for Mango CA:

Data	Value	Description
signerId	User's Signer ID	In general, it is user's mobile number
docUuid	docUuid	This value was previously saved in business application database while sending the value to E-Signer Gateway

signedHash	Hash value	The Sign Hash that is returning from E-Signer Gateway This Value must be used as form-data if the CA is Mango.

Calling This API will return a JSON response. If the response returns with a success then you will find the document as a signed in your business application.