EntityLocker
Document Wallet to Empower Entities

Requester – Entity Locker API Specification

Version 1.0

October 2024

A Digital India Initiative
National e-Governance Division.
Ministry of Electronics and
Information Technology.

Revision History

Version	Date	Comments
0.1	01/10/2024	Released version.

Table of Contents

Requester - Meri Pehchaan API Specification

Introduction

Entity Locker is a transformative initiative designed to streamline document management and sharing for organizations. Built on the principles of the Digital India program, Entity Locker aims to reduce reliance on physical documents by providing a secure, digital platform for storing and exchanging verified documents.

With Entity Locker, authorized organizations can directly upload and share verified digital documents, ensuring that all stakeholders have access to up-to-date information. This platform also allows for the storage of legacy documents, which can be scanned, uploaded, and digitally signed using secure electronic signature facilities. Entities can share these documents effortlessly with various departments and agencies, enhancing collaboration and improving service delivery.

This document outlines the technical specifications and integration guidelines for Entity Locker with applications from trusted partners. It is intended for readers familiar with the functionalities of Entity Locker and the broader digital document ecosystem.

Authorization APIs (For Server Side Web Applications)

You must first obtain the user's authorization to access files in an organization’s Entity Locker account from your application. Entity Locker APIs utilize the OAuth 2.0 protocol for this authorization process. The platform supports standard OAuth 2.0 scenarios applicable to web servers, mobile applications, and limited-input devices such as printers and scanners.

For enhanced security, especially for mobile application clients, Entity Locker also implements the Proof Key for Code Exchange (PKCE) protocol. This adds an extra layer of protection to the authorization flow.

For more information on OAuth 2.0 please refer to Internet Engineering Task Force’s (IETF) documentation on The OAuth 2.0 Authorization Framework (https://tools.ietf.org/html/rfc6749), Proof Key for Code Exchange by OAuth Public Clients (https://tools.ietf.org/html/rfc7636) and OAuth 2.0 for Native Apps (https://tools.ietf.org/html/rfc8252).

Get Authorization Code

Call to this API starts authorization flow using OAuth 2.0 protocol. This isn't an API call—it's a Entity Locker web page that lets the Entity sign in to Entity Locker and authorize your application to access Entity’s data. After the Entity decides whether or not to authorize your app, they will be redirected to the redirect link provided by your application.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/authorize
```

HTTP METHOD

GET

PARAMETERS

- **response_type** (*required*) Provide the grant type parameter as "code".
- **client_id** (*required*) Provide the app id/client id that was created during the application registration process.
- **redirect_uri** (*required*) The URI to redirect the Entity after authorization has completed. This must be the exact URI registered in the DigiLoker Partner Portal. A redirect URI is required for the token flow, but optional for the code flow.
- **state** (*required*) This is your application specific data that will be passed back to your application through *redirect_uri*.
- **code_challenge** (*required*) A unique random string called code verifier (*code_verifier*) is created by the client application for every authorization request. A *code_verifier* is a high-entropy cryptographic random string created using the unreserved characters [A-Z] / [a-z] / [0-9] / "-" / "." / "_" / "~", with a minimum length of 43 characters and a maximum length of 128 characters. The *code_verifier* should have enough entropy to make it impractical to guess the value.

  The code_challenge sent as this parameter is the Base64URL (with no padding) encoded SHA256 hash of the code verifier.

  ```
  code_challenge = base64_url_encode_without_padding(sha256(code_verifier))
  ```

  Here is the pseudo code to implement a base64url-encoding function without padding, based upon the standard base64-encoding function that uses padding:

  ```
  string base64_url_encode_without_padding(string arg)
  {
    string s = base64encode(arg); //Regular base64 encoder with padding
    s = s.replace(’=’,’’); //Remove any trailing ’=’
    s = s.replace(’+’, ’-’); //Replace ’+’ with ’-’
    s = s.replace(’/’, ’_’); //Replace ’/’ with ’_’
   return s;
  }
  ```

- **code_challenge_method** (*required*) Specifies what method was used to encode a *code_verifier* to generate *code_challenge* parameter above. This parameter must be used with the *code_challenge* parameter. The only supported values for this parameter is *S256*.
- **dl_flow** (*optional*) If this parameter is provided its value will always be signup. This parameter indicates that the Entity does not have a Entity Locker account and will be directed to the signup flow directly. After the account is created, the Entity will be

directed to the authorization flow. If this parameter is not sent, the Entity will be redirected to the sign in flow.

- **acr** (*optional*) If this parameter is provided its value will always be either pan, cin or udyam. This parameter indicates that the Entity have to verify their authentic content recognition from Digilocker service, if Entity successfully verified acr then client will receive the verified acr data inside "id_token" in the response of "Get Access Token" API .
- **purpose** (*optional*) If this parameter is provided its value will always be either kyc, verification, compliance, availing_services, or educational. This parameter specifies the purpose of the consent and allows the Entity to change it on the consent screen.
- **consent_valid_till** (*optional*) Provide a timestamp value in seconds using the UNIX (or POSIX) format and the IST time zone. This parameter represents the expiration date for the Entity's consent, if provided.

RETURNS

Since /oauth2/1/authorize is a website, there is no direct return value. However, once a Entity successfully authorizes your app, the Entity Locker application will forward the flow to your redirect URI. The type of response varies based on the response_type.

If the response_type parameter is passed as code then the following parameters are returned in the query string:

- **code** The authorization code, which can be used to attain a bearer token by calling the Get Access Token API.
- **state** This is application specific data, if any, originally passed to /oauth2/1/authorize

ERRORS

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the flow will result in error response.

If the resource owner denies the access request or if the request fails for reasons other than a missing or invalid redirection URI, the following parameters will be included in the redirect URI:

- **error** An error code as per the OAuth 2.0 spec.
- **error_description** A Entity-friendly description of the error that occurred.
- **state** The state content originally passed to authorization flow if any.

Get Access Token

This endpoint only applies to apps using the authorization code flow. An app calls this endpoint to acquire a bearer token once the Entity has authorized the app. Calls to /oauth2/1/token need to be authenticated using the app's key and secret. These can either be passed as application/x-www-form-urlencoded POST parameters (see parameters below) or via HTTP basic authentication. If basic authentication is used, the app key should be provided as the username, and the app secret should be provided as the password.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/token
```

HTTP METHOD POST

HTTP REQUEST HEADER

- *Content-Type: application/x-www-form-urlencoded*

PARAMETERS

- **code**(*required*) The code acquired by directing Entities to /oauth2/1/authorize?response_type=code.
- **grant_type**(*required*) The grant type, which must be authorization_code.
- **client_id** (*required*) If credentials are passed in POST parameters, this parameter should be present and should be the app key/client id.
- **client_secret** (*required*) If credentials are passed in POST parameters, this parameter should be present and should be the app's secret.
- **redirect_uri**(*required*) Only used to validate that it matches the original /oauth2/authorize, not used to redirect again.
- **code_verifier**(*required if code_challenge parameter is passed in authorization request*) The code_verifier created during authorization request. This parameter is mandatory for mobile client applications.

RETURNS

A JSON string containing following fields will be returned in response:

- **access_token** The access token that can be used to call the Entity Locker APIs.
- **expires_in** The duration in seconds for which the access token is valid.
- **token_type** The type of token which will always be Bearer.
- **scope** Scope of the token provided by the Entity and scopes are separated by space. For example : files.issueddocs(*Access to Entity's Get List of Issued Documents API*) files.uploadeddocs(*Access to Entity's Get List of uploaded Documents API*) Entitydetails(*Access to Entity's Get Account Details API*) issued/ in.gov.pan-OPNCR-98765432 (*Access to Entity's Get File From URI API for Organisation PAN Card*) issued/in.gov.mca-CPMTD-201412345678 (*Access to Entity's Get File From URI API for Company Master Details Documents*)
- **consent_valid_till** This contain the timestamp in UNIX format that carry information about the consent expire provided by the Entity.
- **refresh_token** The refresh token used to refresh the above access token when it expires. Please refer to Refresh Access Token API for more details.
- **new_account** This indicates whether the Entity's account existed earlier or the Entity signed up on Entity Locker during the authorization code flow. Possible values are Y and N.

```
Sample Response:
{
    "access_token": "bc125c212a4b03a9a188a858be5a163f379e878a",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope":"files.issueddocs partners.OPNCR partners.CPMTD",
    "consent_valid_till": 1684731048
    "refresh_token": "a47ab18c593703e4f83a274694db7422a8cfcb8f",
    "new_account": "Y",
}
```

ERRORS

The authorization server responds with an HTTP status code as follows:

Code	Description
400	Bad request.
401	If the access token is expired or has been revoked by DigiLocker user.

Refresh Access Token

Access tokens have limited life and expire periodically. Client applications can refresh an access token without requiring the Entity to provide frequent authorizations by logging in to Entity Locker again and again. The client application uses the refresh token obtained in the Get Access Token API response. If the call is successful, new access and refresh tokens are returned.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/token
```

HTTP METHOD POST

HTTP REQUEST HEADER

- *Authorization: Basic <client credentials>* Use HTTP basic authentication of the client using the client_id and client_secret issued to your application.
- *Content-Type: application/x-www-form-urlencoded*

PARAMETERS

- **refresh_token** The refresh token obtained in the response of Get Access token API.
- **grant_type** The grant type, which must be *refresh_token*.

RETURNS

A JSON string containing following fields will be returned in response:

- **access_token** The new access token that can be used to call the Entity Locker APIs.
- **expires_in** The duration in seconds for which the access token is valid.
- **token_type** The type of token which will always be Bearer.
- **scope** Scope of the token provided by the Entity and scopes are separated by space. For example : files.issueddocs(*Access to Entity's Get List of Issued Documents API*) files.uploadeddocs(*Access to Entity's Get List of uploaded Documents API*) Entitydetails(*Access to Entity's Get Account Details API*) issued/ in.gov.pan-OPNCR-98765432 (*Access to Entity's Get File From URI API for Organisation PAN Card*) issued/in.gov.mca-CPMTD-201412345678 (*Access to Entity's Get File From URI API for Company Master Details Documents*)

- **consent_valid_till** This contain the timestamp in UNIX format that carry information
- **refresh_token** The refresh token used to refresh the above access token when it
- **new_account** This indicates whether the Entity's account existed earlier or the Entity signed up on Entity Locker during the authorization code flow. Possible values are Y and N.

```
Sample Response:
{
    "access_token": "11d539dafa5e6b11fe39a5ec266f32c902895485",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope":"entitydetails files.issueddocs partners.OPNCR",
    "consent_valid_till": 1684731048
    "refresh_token": "780506388c6425e551520316bfee16139c200103",
    "digilockerid": "123e4567-e89b-12d3-a456-426655440000",
    "reference_key":"2a33349e7e606a8ad2e30e3c84521f9377450cf09083e162e0a
9b1480ce0f972"
}
```

ERRORS

If the request fails due to missing, invalid, or mismatching parameters, the flow will result in error response. The following parameters will be included in the redirect URI:

- **error** An error code.
- **error_description** A Entity-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
   "error":"invalid_grant_type",
   "error_description":"The grant_type parameter is invalid"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_client	The client credentials are invalid	400
invalid_grant	The refresh token is invalid	400
invalid_grant_type	The grant_type parameter is invalid	400
unexpected_error	Internal server error	500

Token/Session Revocation API

Revoke Token

Client applications can revoke a previously obtained refresh or access token when it is no longer needed. This is done by making a request to the token revocation endpoint. Entity Locker will invalidate the specified token and, if applicable, other tokens based on the same authorisation grant. This API may be used to sign out a Entity from Entity Locker. This API will work for server based web applications, mobile application and limited input devices.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/revoke
```

HHTTP METHOD

POST

HTTP REQUEST HEADER

- *Authorization: Basic <client credentials>* Use HTTP basic authentication of the client using the client_id and client_secret issued to your application.
- *Content-Type: application/x-www-form-urlencoded*

PARAMETERS

- **token**(*required*) The token that needs to be revoked.
- **token_type_hint**(*optional*) The type of the above token. The value will be one of access_token or refresh_token. If this parameter is not sent, Entity Locker will look for this token in both access and refresh tokens and then revoke it.

RETURNS

Since /oauth2/1/authorize is a website, there is no direct return value. However, once a Entity successfully authorizes your app, the Entity Locker application will forward the flow to your redirect URI. The type of response varies based on the response_type.

Revoke Session

This isn't an API call—it's a Entity Locker logout URL that gets the Entity session logged out. After calling this URL in the same web browser, Partner will be redirected to the redirect link as provided in the client ID *settings* section.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/signin/logout/Y
```

RETURNS

Since /signin/logout/Y is a website, there is no direct return value. However, once a Entity successfully authorizes your app, the Entity Locker application will forward the flow to your redirect URI with error=Entity_loggedout&error_description=Entity

Account Detail API

Get Entity Details

Client applications can call this API to get the Entity details. An access token is required to call this API. The API will return the entity details of the account with which the access token is linked. It is strongly recommended that the API can be called by client application only once after acquiring the access token.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/entity
```

HTTP METHOD **GET**

HTTP REQUEST HEADER

- *Authorization: Bearer <access token>*

PARAMETERS

There are no parameters for this API.

RETURNS

Returns following user details in JSON format:

- **entitylockerid** A unique 36 character Entity Locker ID of the entity account.
- **name** The name of the entity as registered with Entity Locker.
- **doi** This is date of incorporation of the entity the company was legally formed in DD-MM-YYYY format.
- **email** This is email ID of the entity as registered with Entity Locker.
- **mobile** This is mobile number of the entity as registered with Entity Locker.
- **verified_by** The verification method for the entity, either through the organization’s PAN, Udyam, CIN.

```
Sample Response:
HTTP/1.1 200 OK
Content-Type: application/json
{
    "entitylockerid": "123e4567-e89b-12d3-a456-426655440000",
    "name": "ABC Enterprise",
    "doi": "dd-mm-yyyy"
    "email": null,
    "mobile": null,
    "verified_by": "PAN/UD/CIN"
}
```

ERRORS

If the request fails, corresponding HTTP status code will be returned along with the following parameters in the response:

- **error** An error code.
- **error_description** A Entity-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 500 Internal Server Error
Content-Type: application/json;charset=UTF-8

{
   "error":"unexpected_error",
   "error_description":"Internal server error"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_token	The access token is invalid	401
insufficient_scope	The request requires higher privileges than provided by the access token	403
unexpected_error	Internal server error	530

Get User Details

Client applications can call this API to get the DigiLocker Id, name, date of birth and gender of the account holder. An access token is required to call this API. The API will return the user details of the account with which the access token is linked. It is strongly recommended that the API can be called by client application only once after acquiring the access token. Since the user details do not change, the client application may store the values and use them when necessary than calling this API repeatedly.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/user
```

HTTP METHOD **GET**

HTTP REQUEST HEADER

- *Authorization: Bearer <access token>*

PARAMETERS

There are no parameters for this API.

RETURNS

Returns following user details in JSON format:

- **digilockerid** A unique 36 character DigiLocker ID of the user account.
- **name** The name of the user as registered with DigiLocker.
- **dob** This is date of birth of the user as registered with DigiLocker in DDMMYYYY format.
- **gender** This is gender of the user as registered with DigiLocker. The possible values are M, F, T for male, female and transgender respectively.
- **eaadhaar** This indicates whether eAadhaar data is available for this account. Possible values are Y and N.
- **reference_key** This is DigiLocker account reference key. This is used only as a transient reference for tracing.
- **mobile** This is mobile number of the user as registered with DigiLocker
- **picture** This is picture of the user as registered with DigiLocker
- **email** This is email ID of the user as registered with DigiLocker

```
Sample Response:
HTTP/1.1 200 OK
Content-Type: application/json
{
    "digilockerid": "123e4567-e89b-12d3-a456-426655440000",
    "name": "Ajit Kumar",
    "dob": "31121970",
    "gender": "M",
    "eaadhaar": "Y",
    "reference_key":
"2a33349e7e606a8ad2e30e3c84521f9377450cf09083e162e0a9b1480ce0f972"
    "mobile": "9999999999"
    "picture": "/9j/4AAQSkZJRgABAgAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQ
Ksjdhcvfiasdviasdv/sakbvjhas/aksjbfcijakjadnfcijabuw238923bi2jbc923fkdbc
iub/ksdjnji………………"
    "email": "email@test.com"
}
```

ERRORS

If the request fails, corresponding HTTP status code will be returned along with the following parameters in the response:

- **error** An error code.
- **error_description** A Entity-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 500 Internal Server Error
Content-Type: application/json;charset=UTF-8

{
   "error":"unexpected_error",
   "error_description":"Internal server error"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_token	The access token is invalid	401
insufficient_scope	The request requires higher privileges than provided by the access token	403
unexpected_error	Internal server error	530

Enterprise Vault Document File APIs

These APIs allow your application to get the meta-data about issued and uploaded documents in Entity's Entity Locker. It allows downloading of a file from issued and uploaded documents.

Get List of Self Uploaded Documents

Returns the list of meta-data about documents or folders in Entity's Entity Locker in a specific location.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/entity/files/id
```

HTTP METHOD **GET**

HTTP REQUEST HEADER

- *Authorization: Bearer <access token>*

PARAMETERS

- **id** The id of the folder to list. To list the files of root folder of a Entity's locker, do not send this parameter. This is sent as a part of the URL.

RETURNS

Returns meta-data about contents of a given folder in JSON format containing following fields in response:

- **name** The name of the file or folder.
- **type** String dir for folder and string file for file.
- **id** The id if this item is a folder.
- **size** Size of file or folder.
- **date** This contains the date of file upload in case of self uploaded documents.
- **parent** The id of the parent folder.
- **mime** The mime type of the file. This field will contain “application/PDF” for PDF files; “image/png” for PNG files and “image/jpg” or “image/jpeg” for JPG/JPEG files. This will be blank in case of folder.
- **uri** This is the unique identifier of the document shared by the Entity in Entity Locker. You will use this identifier to get the actual file from Entity Locker using the API. URI will be blank in case of folder.
- **description** This is the descriptive document type stored in Entity Locker such as ‘Organisation PAN Verification Record’ or ‘Company Master Details’.
- **issuer** The name of the issuer. This is blank in case of uploaded documents and folders.

```
Sample Response:
{
    "directory":"/",
            "items":[
            {
                    "name":"My Documents",
                    "type":"dir",
                    "id":"5678",
                    "size":"366481",
                    "date":" 2015-05-12T15:50:38Z",
                    "parent":"1234",
                    "mime":"",
                    "uri": "",
                    "description": "",
                    "issuer": ""
            },
            {
                    "name":"myfile.pdf",
                    "type":"file",
                    "id":"",
                    "size":"366481",
                    "date":" 2015-05-12T15:50:38Z",
                    "parent":"1234",
                    "mime":"application/pdf",
                    "uri": "in.gov.digilocker-OTHER-39491058586222",
                    "description": " ",
                    "issuer": ""
                    }
    ]
}
```

ERRORS

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the flow will result in error response.
If the request fails for reasons other than a missing or invalid redirection URI, corresponding HTTP status code will be returned along with the following parameters in the response:

- **error** An error code.
- **error_description** A Entity-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 404 Not Found
Content-Type: application/json;charset=UTF-8

{
   "error":"invalid_id",
   "error_description":"The folder does not exist"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_token	The access token is invalid	401
invalid_id	The folder does not exist	404
insufficient_scope	The request requires higher privileges than provided by the access token	403
unexpected_error	Internal server error	530

Get List of Issued Documents

Returns the list of meta-data about issued documents in Entity's Entity Locker.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/2/entity/files/issued
```

HTTP METHOD GET

HTTP REQUEST HEADER

- *Authorization: Bearer <access token>*

PARAMETERS

There are no parameters for this API.

RETURNS

Returns meta-data about issued documents in JSON format containing following fields in response:

- **name** The name of the certificate.
- **type** String file.
- **size** This will be blank.
- **date** This contains the date on which the certificate was last modified in Entity Locker.
- **parent**This will be blank.
- **mime** The list of mime types for the certificate data. This field will contain “application/PDF” or “application/xml”.
- **uri** This is the unique identifier of the document shared by the Entity in Entity Locker. You will use this identifier to get the actual file from Entity Locker using the API.
- **doctype** A 5 character unique document type provided by Entity Locker.
- **description** This is the descriptive document type stored in Entity Locker such as ‘Organisation PAN Verification Record’ or ‘Company Master Details’.
- **issuerid** Unique Entity Locker issuer id as mentioned in the URI.
- **issuer** The name of the issuer.

```
Sample Response:
{
    "items":[
    {
            "name":"Company Master Details",
            "type":"file",
            "size":"",
            "date":" 2024-05-12T15:50:38Z",
            "parent":"",
            "mime":"application/pdf",
            "uri": "in.gov.mca-CPMTD-201412345678",
            "doctype": "CPMTD",
            "description": "Company Master Details",
            "issuerid": "in.gov.mca",
            "issuer": "MINISTRY OF CORPORATE AFFAIRS"
    },
    {
            "name":"Organisation PAN Verification Record",
            "type":"file",
            "size":"",
            "date":" 2024-05-12T15:50:38Z",
            "parent":"",
            "mime": [{"application/pdf"},{"application/xml"}],
            "uri": " in.gov.pan-OPNCR-98765432",
            "doctype": "OPNCR",
            "description": "Organisation PAN Verification Record",
                "issuerid": "in.gov.pan",
                "issuer": "Income Tax Department"
                }
        ]
}
```

ERRORS

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the flow will result in error response.
If the request fails for reasons other than a missing or invalid redirection URI, corresponding HTTP status code will be returned along with the following parameters in the response:

- **error** An error code.
- **error_description** A Entity-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 500 Internal Server Error
Content-Type: application/json;charset=UTF-8

{
   "error":" partner_service_unresponsive ",
   "error_description":"Internal server error"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_token	The access token is invalid	401
insufficient_scope	The request requires higher privileges than provided by the access token	403
partner_service_unresponsive	Internal server error	530
unexpected_error	Internal server error	530

Get File from URI

Returns a file from URI. This API can be used to fetch both issued document and uploaded document.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/entity/file/uri
```

HTTP METHOD GET

HTTP REQUEST HEADER

- *Authorization: Bearer <access token>*

PARAMETERS

- **uri** The URI of the file to download. This is sent as a part of the url.

HTTP RESPONSE HEADER

- **Content-Type** The mime type of the file e.g. image/jpg, image/jpeg, image/png, application/pdf
- **Content-Length** Size of file.
- **hmac** This is used to verify the integrity of the file data. Entity Locker calculates the hash message authentication code (hmac) of the file content using SHA256 hashing algorithm and the client secret as the hashing key. The resulting hmac is converted to Base64 format and sent in this parameter. It is strongly recommended that the client app calculates the hmac of the downloaded file data and compares it with this hmac.

RETURNS

Returns data of the file in the response body.

ERRORS

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the flow will result in error response.
If the request fails for reasons other than a missing or invalid redirection URI, corresponding HTTP status code will be returned along with the following parameters in the response:

- **error** An error code.
- **error_description** A Entity-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
   "error":"uri_missing",
   "error_description":"URI parameter missing"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_token	The access token is invalid	401
uri_missing	URI parameter missing	400
insufficient_scope	The request requires higher privileges than provided by the access token	403
invalid_uri	No file found for given URI	404
repository_service_resperror	Internal server error	530
repository_service_exception	Internal server error	530

Get Certificate Data in XML Format from URI

Returns the certificate data in machine readable XML format for a URI. This API can be used to only for issued documents. The XML data may not be available for all documents. If the XML data is available for a particular document, the mime parameter in Get List of

Issued Documents API will contain application/xml. Please refer to Entity Locker XML Certificate Formats for more details of XML formats of various documents.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/entity/xml/uri
```

HTTP METHOD GET

HTTP REQUEST HEADER

- *Authorization: Bearer <access token>*

PARAMETERS

- **uri** The URI of the file to download. This is sent as a part of the url.

HTTP RESPONSE HEADER

- **Content-Type** The mime type of the file which will be application/xml
- **Content-Length** Size of file.
- **hmac** This is used to verify the integrity of the file data. Entity Locker calculates the hash message authentication code (HMAC) of the file content using SHA256 hashing algorithm and the client secret as the hashing key. The resulting HMAC is converted to Base64 format and sent in this parameter. It is strongly recommended that the client app calculates the HMAC of the downloaded file data and compares it with this HMAC.

RETURNS

Returns XML file containing certificate data in the response body.

ERRORS

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the flow will result in error response.
If the request fails for reasons other than a missing or invalid redirection URI, corresponding HTTP status code will be returned along with the following parameters in the response:

- **error** An error code.
- **error_description** A Entity-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
   "error":"uri_missing",
   "error_description":"URI parameter missing"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_token	The access token is invalid	401

uri_missing	URI parameter missing	400
insufficient_scope	The request requires higher privileges than provided by the access token	403
invalid_uri	No file found for given URI	404
repository_service_resperror	Internal server error	530
repository_service_exception	Internal server error	530

Upload File to Entity Locker

This API can be used to save/upload a file to uploaded documents in Entity Locker. The allowed file types are JPG, JPEG, PNG and PDF. The file size must not exceed 10MB.

URL STRUCTURE

```
Production Environment:
https://entity.digilocker.gov.in/public/oauth2/1/file/upload
```

HTTP METHOD **POST**

HTTP REQUEST HEADER

- *Authorization: Bearer <access token>*
- **Content-Type** The mime type of the file e.g. image/jpg, image/jpeg, image/png, application/pdf
- **path** The destination path of the file in Entity Locker including filename.
- **hmac** This is used to verify the integrity of the file data. The client app calculates the hash message authentication code (HMAC) of the file content using SHA256 hashing algorithm and the client secret as the hashing key. The resulting HMAC is converted to Base64 format and sent in this parameter. Upon upload of file, Entity Locker calculates the HMAC of the file data and compares it with this HMAC.

HTTP REQUEST BODY

Provide data of the file in the request body.

RETURNS

Returns meta-data about the uploaded document in JSON format containing following fields in response:

- **path** The destination path of the file in Entity Locker including filename.
- **size** Size of file.

ERRORS

If the request fails due to a missing, invalid, or mismatching redirection URI, or if the client identifier is missing or invalid, the flow will result in error response.

If the request fails for reasons other than a missing or invalid redirection URI, corresponding HTTP status code will be returned along with the following parameters in the response:

- **error** An error code.
- **error_description** A user-friendly description of the error that occurred.

```
Sample Error Response:
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
   "error":"contenttype_missing",
   "error_description":"Content-Type parameter is missing"
}
```

The following table lists the possible error codes:

Error	error_description	HTTP Response Code
invalid_token	The access token is invalid	401
insufficient_scope	The request requires higher privileges than provided by the access token	403
path_missing	Path parameter is missing	400
contenttype_missing	Content-Type parameter is missing	400
hmac_missing	HMAC parameter is missing	400
filename_missing	Filename is missing in path parameter	400
hmac_mismatch	HMAC does not match	400
invalid_filename	Restricted characters (\ / : * ? < > | ' ^ and ~) are not allowed in file name	400
invalid_filesize	The file size exceeds maximum allowed file size of 10MB	400
invalid_filetype	The file type is not allowed	400
invalid_path	The destination folder does not exist	400
file_data_missing	Missing file content in the request	400
mimetype_mismatch	The mimetype provided in Content-Type parameter does not match with the mimetype of the file	400
unexpected_error	Internal server error	530