March 2020

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

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>03/03/2017</td>
<td>Released draft version.</td>
</tr>
<tr>
<td>0.2</td>
<td>29/03/2017</td>
<td>Changed Upload File API parameters.</td>
</tr>
<tr>
<td>0.3</td>
<td>18/04/2017</td>
<td>Removed Content-Length parameter from Upload File API.</td>
</tr>
<tr>
<td>0.4</td>
<td>19/05/2017</td>
<td>Added error codes for APIs.</td>
</tr>
<tr>
<td>0.5</td>
<td>07/07/2017</td>
<td>Added support for signup flow and invalidate token APIs.</td>
</tr>
<tr>
<td>0.6</td>
<td>17/08/2017</td>
<td>Added APIs for limited input devices.</td>
</tr>
<tr>
<td>0.7</td>
<td>15/01/2018</td>
<td>Internal Release for UMANG</td>
</tr>
<tr>
<td>0.8</td>
<td>27/06/2018</td>
<td>Refresh Token for limited input devices.</td>
</tr>
<tr>
<td>0.9</td>
<td>20/08/2018</td>
<td>Updated Get Device Code API for Limited Input Devices to accept mobile number.</td>
</tr>
<tr>
<td>1.0</td>
<td>25/09/2018</td>
<td>Added Pull Document APIs</td>
</tr>
<tr>
<td>1.1</td>
<td>12/10/2018</td>
<td>Added Verify Account API</td>
</tr>
<tr>
<td>1.2</td>
<td>16/10/2018</td>
<td>Updated Pull Document API to return URI. Change authentication of Issuer, Document and Parameter APIs.</td>
</tr>
<tr>
<td>1.3</td>
<td>07/12/2018</td>
<td>Added provision to accept trusted mobile number in Get Authorization Code sign up flow. Added provision to skip demographic authentication in demographic authentication sign up API.</td>
</tr>
<tr>
<td>1.4</td>
<td>12/04/2019</td>
<td>Added API to provide DigiLocker Id and user details. Added provision to provide certificate data in machine readable format.</td>
</tr>
<tr>
<td>1.5</td>
<td>17/06/2019</td>
<td>Updated Verify Account API to return digilockerid.</td>
</tr>
<tr>
<td>1.6</td>
<td>14/07/2019</td>
<td>Added DigiLocker Id and user details in Get Access Token response. Added Get Statistics API to get general DigiLocker statistics. Added doctype in Get List of Issued Documents API response. Added issuerrid in the response of Get List of Issued Documents API and Get List of Issuers API.</td>
</tr>
<tr>
<td>1.7</td>
<td>12/03/2020</td>
<td>Added eadhaar indicator in token and Get User Detail API. Added Get e-Aadhaar Data in XML Format API. Updated Sign Up API to provide access token if account already exists. The API will not throw any error if account already exists for a give Aadhaar.</td>
</tr>
<tr>
<td>1.8</td>
<td>25/03/2020</td>
<td>Updated Get Authorization Code and Get Access Token APIs to support Proof Key for Code Exchange protocol for better security.</td>
</tr>
</tbody>
</table>
# Table of Contents

**Revision History** .................................................................................................................. 1

**Introduction** .......................................................................................................................... 3

**Authorization APIs (For Server Side Web Applications)** ...................................................... 3
  - Get Authorization Code ........................................................................................................ 3
  - Get Access Token ................................................................................................................ 5
  - Refresh Access Token .......................................................................................................... 7

**Authorization APIs (For Limited Input Devices)** ............................................................... 9
  - Get Device Code .................................................................................................................. 9
  - Get Access Token ................................................................................................................ 11

**Token Revocation API** ......................................................................................................... 12
  - Revoke Token ...................................................................................................................... 12

**Account Detail API** ............................................................................................................. 13
  - Get User Details .................................................................................................................. 13

**File APIs** ............................................................................................................................ 14
  - Get List of Self Uploaded Documents ............................................................................... 14
  - Get List of Issued Documents ......................................................................................... 16
  - Get File from URI ............................................................................................................. 18
  - Get Certificate Data in XML Format from URI ............................................................... 20
  - Get e-Aadhaar Data in XML Format ............................................................................. 21
  - Upload File to Locker ....................................................................................................... 22
  - Pull Document ................................................................................................................... 24

**DigiLocker Sign up APIs using Aadhaar Demographic Authentication** .............................. 26
  - SIGN UP ............................................................................................................................ 27
  - Verify OTP .......................................................................................................................... 29

**DigiLocker Meta APIs** ....................................................................................................... 31
  - Get List of Issuers ............................................................................................................. 31
  - Get List of Documents Provided by an Issuer ............................................................... 33
  - Get Search Parameters for a Document ........................................................................ 35
  - Verify Account .................................................................................................................. 37
  - Push URI to Account ........................................................................................................ 39
  - Get Statistics .................................................................................................................... 41
Authorized Partner API Specification

Introduction
Digital Locker is a key initiative under Digital India program. It aims at eliminating the use of physical documents and enables sharing of verified electronic documents across agencies. Digital Locker provides a dedicated personal storage space of 1GB in the cloud to citizens. It will enable various organizations registered on Digital Locker to push certificates of citizens directly in their Digital Locker in electronic format. Citizens can also upload and securely store the scanned copies of legacy documents in Digital Locker. These legacy documents can be electronically signed using eSign facility. A citizen can share these electronic certificates online with various departments and agencies registered on Digital Locker while applying for the services provided by them. Thus, Digital Locker brings the citizens, issuers and requestors on one platform.

This document provides technical details of DigiLocker integration with applications of trusted partners. This document assumes that the reader is aware of Digital Locker application functionality.

Authorization APIs (For Server Side Web Applications)
To access files in user’s DigiLocker account from your application, you must first obtain user’s authorization. DigiLocker APIs use the OAuth 2.0 protocol for authorization. DigiLocker supports common OAuth 2.0 scenarios such as those for web server, mobile applications and limited input devices such as printers and scanners. DigiLocker also supports the Proof Key for Code Exchange (PKCE) protocol for higher security of mobile application clients. 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](https://tools.ietf.org/html/rfc6749)), Proof Key for Code Exchange by OAuth Public Clients ([https://tools.ietf.org/html/rfc7636](https://tools.ietf.org/html/rfc7636)) and OAuth 2.0 for Native Apps ([https://tools.ietf.org/html/rfc8252](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 DigiLocker web page that lets the user sign in to Digi Locker and authorize your application to access user’s data. After the user 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://api.digitallocker.gov.in/public/oauth2/1/authorize

**HTTP METHOD**
GET

**PARAMETERS**
- **response_type** *(required)*
  Provide the grant type requested, either token or 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 user 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** *(optional but recommended for server based client applications, required for mobile client applications)* 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`.  

\[
\text{code\textunderscore challenge} = \text{base64\textunderscore url\textunderscore encode\textunderscore without\textunderscore padding(sha256(code\textunderscore verifier))}
\]

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

```c
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 if code_challenge parameter is passed)* 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 user does not have a DigiLocker account and will be directed to the signup flow directly. After the account is created, the user will be directed to the authorization flow. If this parameter is not sent, the user will be redirected to the sign in flow.

• **verified_mobile** *(optional)* Verified mobile number of the user. If this parameter is passed, DigiLocker will skip the mobile OTP verification step during sign up. DigiLocker will treat the mobile number passed in this parameter as a verified mobile number by the trusted client application. This parameter will be used only if `dl_flow` parameter mentioned above is set to `signup` and will be ignored otherwise.
**RETURNS**

Since /oauth2/1/authorize is a website, there is no direct return value. However, once a user successfully authorizes your app, the DigiLocker 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.

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

- **access_token**  The access token that can be used to call the DigiLocker 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.

**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 an 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 user-friendly description of the error that occurred.
- **state**  The state content originally passed to the 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 user 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://api.digitallocker.gov.in/public/oauth2/1/token

**HTTP METHOD**  POST

**HTTP REQUEST HEADER**
PARAMETERS

- **code**(required) The code acquired by directing users 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 DigiLocker 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.
- **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.
- **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.

Sample Response:

```json
{
    "access_token": "bc125c212a4b03a9a188a858be5a163f379e878a",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope":,
    "refresh_token": "a47ab18c593703e4f83a274694db7422a8cfcb8f",
    "digilockerid": "123e4567-e89b-12d3-a456-426655440000",
    "name": "Sunil Kumar",
    "dob": "31121970",
    "gender": "M",
    "eaadhaar": "Y"
}
```
ERRORS
The authorization server responds with an HTTP status code as follows:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Bad request.</td>
</tr>
<tr>
<td>401</td>
<td>If the access token is expired or has been revoked by DigiLocker user.</td>
</tr>
</tbody>
</table>

Refresh Access Token
Access tokens have limited life and expire periodically. Client applications can refresh an access token without requiring the user to provide frequent authorizations by logging in to DigiLocker 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://api.digitallocker.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 DigiLocker 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.
- refresh_token The refresh token used to refresh the above access token when it expires.
- 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.

**Sample Response:**

```json
{
    "access_token": "11d539dafa5e6b11fe39a5ec266f32c902895485",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope":,
    "refresh_token": "780506388c6425e551520316bfee16139c200103",
    "digilockerid": "123e4567-e89b-12d3-a456-42655440000",
    "name": "Sunil Kumar",
    "dob": "31121970",
    "gender": "M",
    "eaadhaar": "Y"
}
```

**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 user-friendly description of the error that occurred.

**Sample Error Response:**

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

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

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client</td>
<td>The client credentials are invalid</td>
<td>400</td>
</tr>
<tr>
<td>invalid_grant</td>
<td>The refresh token is invalid</td>
<td>400</td>
</tr>
<tr>
<td>invalid_grant_type</td>
<td>The grant_type parameter is invalid</td>
<td>400</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>
Authorization APIs (For Limited Input Devices)

This section explains how to implement OAuth 2.0 authorization to access DigiLocker APIs from applications running on devices that either do not have access to a browser or have limited input capabilities such as printers and scanners. Although this API is specifically designed for limited input devices, they can also be used by web applications clients.

When a user decides to use the DigiLocker application from a limited input device that supports DigiLocker APIs, the device first accepts the DigiLocker username of the user. The device calls the DigiLocker API to get the device code for by providing the client_id provided to the device OEM and the username of the user. Digi Locker responds with a device code and then sends an OTP on the mobile number and email address associated with the user's account. The user enters the OTP on the device. The device then sends the device code and OTP to Digi Locker. Digi Locker validates the device code and the OTP and responds with the access token that can be used to access Digi Locker APIs.

Get Device Code

The client device calls the Digi Locker API to get the device code by providing the client_id issued to the device OEM and either the username or the mobile number of the user. Digi Locker responds with a device code and then sends an OTP on the mobile number and email address associated with the user's account. The masked mobile number and email address is also sent in response. The device should use these values to notify the users about the mobile and email address on which the OTP has been sent.

URL STRUCTURE

Production Environment:
https://api.digitallocker.gov.in/public/oauth2/1/code

HTTP METHOD
POST

PARAMETERS

- **response_type** *(required)* The parameter must be set to device_code.
- **client_id** *(required)* Provide the app id/client id that was created during the application registration process.
- **dl_username** *(optional)* Digi Locker username of the user. The client device accepts the username on the device from user and sends the username in this parameter. Either the username or the mobile number must be provided.
- **dl_mobile** *(optional)* Mobile number associated with Digi Locker account of the user. The client device accepts the mobile number on the device from user and sends it in this parameter. Either the username or the mobile number must be provided.

RETURNS

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

- **device_code** The device verification code.
- **dl_masked_mobile** The masked mobile number on which the OTP is sent.
- **dl_masked_email** The masked email on which the OTP is sent.
• **expires_in**  The duration in seconds for which the code is valid.

**Sample Response:**
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

```json
{
  "device_code": "74tq5miHKB",
  "dl_masked_mobile": "XXXX XXX 1234",
  "dl_masked_email": "xxxxxx@gmail.com",
  "expires_in": 600
}
```

**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 user-friendly description of the error that occurred.

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

```json
{
  "error": "invalid_username",
  "error_description": "The dl_username parameter is invalid"
}
```

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
<tr>
<td>invalid_response_type</td>
<td>The response_type parameter is invalid</td>
<td>400</td>
</tr>
<tr>
<td>invalid_username</td>
<td>The dl_username parameter is invalid</td>
<td>400</td>
</tr>
<tr>
<td>invalid_mobile</td>
<td>The dl_mobile parameter is invalid</td>
<td>400</td>
</tr>
<tr>
<td>multiple_mobile_accounts</td>
<td>The mobile number provided is associated with more than one DigiLocker accounts. Please login using username.</td>
<td>400</td>
</tr>
<tr>
<td>otp_error</td>
<td>OTP could not be sent as the account may not have a valid mobile or a valid email.</td>
<td>500</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>
Get Access Token

After receiving the successful device code response, the client device prompts the end user to enter the OTP on the device. The user manually enters the OTP on the device. The device then calls the token endpoint with the OTP and other parameters to acquire a bearer token.

**URL STRUCTURE**

<table>
<thead>
<tr>
<th>Production Environment:</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://api.digitallocker.gov.in/public/oauth2/1/token">https://api.digitallocker.gov.in/public/oauth2/1/token</a></td>
</tr>
</tbody>
</table>

**HTTP METHOD** POST

**HTTP REQUEST HEADER**

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

**PARAMETERS**

- **grant_type** The grant type, which must be "urn:ietf:params:oauth:grant-type:device_code".
- **device_code** The device verification code. The device_code field from Get Device Code Response.
- **client_id** Provide the app id/client id that was created during the application registration process.
- **dl_otp** The OTP collected from the user.

**RETURNS**

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

- **access_token** The access token that can be used to call the DigiLocker 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.
- **refresh_token** The refresh token used to refresh the above access token when it expires. This will value will be returned only in case of web applications and not be returned for limited input devices. Please refer to Refresh Access Token API for more details.

**Sample Response:**

```json
{
  "access_token": "bc125c212a4b03a9a188a858be5a163f379e878a",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": 
  "refresh_token": "ad6f4004cc3e53f56bda391fe485fc4d32928061"
}
```
**ERRORS**

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_grant_type</td>
<td>The grant_type parameter is invalid</td>
<td>400</td>
</tr>
<tr>
<td>invalid_device_code</td>
<td>The device_code parameter is invalid or expired</td>
<td>401</td>
</tr>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
<tr>
<td>invalid_otp</td>
<td>The OTP is invalid.</td>
<td>401</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

**Token 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. DigiLocker 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 user from DigiLocker. This API will work for server based web applications, mobile application and limited input devices.

**URL STRUCTURE**

**Production Environment:**
https://api.digitallocker.gov.in/public/oauth2/1/revoke

**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**

- **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, DigiLocker will look for this token in both access and refresh tokens and then revoke it.

**RETURNS**

The authorization server responds with HTTP status code 200 if the token has been revoked successfully or if the client submitted an invalid token. Invalid tokens do not cause an error response. An invalid token type hint value is ignored by the authorization server and does not influence the revocation response.
Account Detail API

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://api.digitallocker.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.
- **eadhaar** This indicates whether eAadhaar data is available for this account. Possible values are Y and N.

Sample Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "digilockerid": "123e4567-e89b-12d3-a456-42665440000",
    "name": "Sunil Kumar",
    "dob": "31121970",
    "gender": "M",
    "eadhaar": "Y"
}
```
**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 user-friendly description of the error that occurred.

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

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

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

**File APIs**
File APIs allow your application to get the meta-data about issued and uploaded documents in user's DigiLocker. It allows downloading of a file from issued and uploaded documents. It also allows uploading of a file in uploaded document section of user's account.

**Get List of Self Uploaded Documents**
Returns the list of meta-data about documents or folders in user's DigiLocker in a specific location.

**URL STRUCTURE**

**Production Environment:**
https://api.digitallocker.gov.in/public/oauth2/1/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 user'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/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 user in DigiLocker. You will use this identifier to get the actual file from Digi Locker using the API. URI will be blank in case of folder.
- **description**  This is the descriptive document type stored in Digi Locker such as ‘Income Certificate’ or ‘Driving License’.
- **issuer**  The name of the issuer. This is blank in case of uploaded documents and folders.

Sample Response:

```json
{
   "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": "Income Certificate",
"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 user-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:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>invalid_id</td>
<td>The folder does not exist</td>
<td>404</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

Get List of Issued Documents
Returns the list of meta-data about issued documents in user’s DigiLocker.

**URL STRUCTURE**

Production Environment:
https://api.digitallocker.gov.in/public/oauth2/2/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 DigiLocker.
- **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 user in Digi Locker. You will use this identifier to get the actual file from Digi Locker using the API.
- **doctype** A 5 character unique document type provided by Digi Locker.
- **description** This is the descriptive document type stored in Digi Locker such as ‘Income Certificate’ or ‘Driving License’.
- **issuerid** Unique Digi Locker issuer id as mentioned in the URI.
- **issuer** The name of the issuer.

Sample Response:
```json
{
   "items": [
      {
         "name": "Class XII Marksheet",
         "type": "file",
         "size": "",
         "date": "2015-05-12T15:50:38Z",
         "parent": "",
         "mime": "application/pdf",
         "uri": "in.gov.cbse-HSCER-201412345678",
         "doctype": "HSCER",
         "description": "Class XII Marksheet",
         "issuerid": "in.gov.cbse",
         "issuer": "CBSE"
      },
      {
         "name": "Income Certificate",
         "type": "file",
         "size": "",
         "date": "2015-05-12T15:50:38Z",
         "parent": "",
         "mime": [{"application/pdf"}, {"application/xml"}]
      }
   ]
}```
"uri": "in.gov.delhi-INCER-98765432",
"doctype": "INCER",
"description": "Income Certificate",
"issuerid": "in.gov.delhi",
"issuer": "Delhi eDistrict"
}
}

**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/1.1 500 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:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>partner_service_unresponsive</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

**Get File from URI**

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

**URL STRUCTURE**

<table>
<thead>
<tr>
<th>Production Environment:</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://api.digitallocker.gov.in/public/oauth2/1/file/uri">https://api.digitallocker.gov.in/public/oauth2/1/file/uri</a></td>
</tr>
</tbody>
</table>

**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. DigiLocker 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 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": "uri_missing",
  "error_description": "URI parameter missing"
}

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>uri_missing</td>
<td>URI parameter missing</td>
<td>400</td>
</tr>
<tr>
<td>invalid_uri</td>
<td>No file found for given URI</td>
<td>404</td>
</tr>
<tr>
<td>repository_service_unresponsive</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>repository_service_unpublished</td>
<td>Internal server error</td>
<td>503</td>
</tr>
</tbody>
</table>
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 Digital Locker XML Certificate Formats for more details of XML formats of various documents.

URL STRUCTURE

Production Environment:
https://api.digitallocker.gov.in/public/oauth2/1/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. DigiLocker 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 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":"uri_missing",
    "error_description":"URI parameter missing"
}

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>uri_missing</td>
<td>URI parameter missing</td>
<td>400</td>
</tr>
<tr>
<td>invalid_uri</td>
<td>No file found for given URI</td>
<td>404</td>
</tr>
<tr>
<td>repository_service_unresponsive</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>repository_service_unpublished</td>
<td>Internal server error</td>
<td>503</td>
</tr>
<tr>
<td>repository_service_inactive</td>
<td>Internal server error</td>
<td>503</td>
</tr>
<tr>
<td>repository_service_configerror</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>repository_service_resperror</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>repository_service_exception</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

Get e-Aadhaar Data in XML Format
Returns e-Aadhaar data in XML format for the account.

**URL STRUCTURE**

Production Environment:
https://api.digitallocker.gov.in/public/oauth2/1/xml/eaadhaar

**HTTP METHOD**
GET

**HTTP REQUEST HEADER**
- Authorization: Bearer <access token>

**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. DigiLocker 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 e-Aadhaar data in the response body.
**ERRORS**
If the client identifier is missing or invalid, the flow will result in error response with 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": "uri_missing",
    "error_description": "URI parameter missing"
}
```

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>aadhaar_not_linked</td>
<td>Aadhaar is not linked to the account</td>
<td>404</td>
</tr>
<tr>
<td>aadhaar_not_available</td>
<td>Aadhaar data is not available for this user. Please perform Aadhaar eKYC again.</td>
<td>404</td>
</tr>
<tr>
<td>repository_service_unresponsive</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>repository_service_unpublished</td>
<td>Internal server error</td>
<td>503</td>
</tr>
<tr>
<td>repository_service_inactive</td>
<td>Internal server error</td>
<td>503</td>
</tr>
<tr>
<td>repository_service_configerror</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>repository_service_respserror</td>
<td>Internal server error</td>
<td>500</td>
</tr>
<tr>
<td>repository_service_exception</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

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

**URL STRUCTURE**

**Production Environment:**
https://api.digitallocker.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 DigiLocker 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, DigiLocker 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 DigiLocker 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:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>path_missing</td>
<td>Path parameter is missing</td>
<td>400</td>
</tr>
<tr>
<td>contenttype_missing</td>
<td>Content-Type parameter is missing</td>
<td>400</td>
</tr>
<tr>
<td>hmac_missing</td>
<td>HMAC parameter is missing</td>
<td>400</td>
</tr>
<tr>
<td>filename_missing</td>
<td>Filename is missing in path parameter</td>
<td>400</td>
</tr>
<tr>
<td>hmac_mismatch</td>
<td>HMAC does not match</td>
<td>400</td>
</tr>
</tbody>
</table>
Pull Document

This API allows a client application to search a document/certificate from issuer's repository using the parameters provided by a user. The searched document is saved in user's issued document section of DigiLocker if the search is successful.

**URL STRUCTURE**

**Production Environment:**
https://api.digitallocker.gov.in/public/oauth2/1/pull/pulldocument

**HTTP METHOD** POST

**HTTP REQUEST HEADER**

- **Authorization: Bearer <access token>**
- **Content-Type: application/x-www-form-urlencoded**

**PARAMETERS**

- **orgid** The organization id for the issuer in DigiLocker. This organization id is returned in Get List of Issuers API mentioned above.
- **doctype** A 5 character unique document type provided by DigiLocker. The doctype is returned in Get List of Documents API mentioned above.
- **consent** The consent indicator from the user for sharing his Aadhaar details with the issuer for pulling the document into his/her DigiLocker. The Partner Application must capture the consent from the user. The possible values are “Y” and “N”. The pull request will be processed only when this indicator is “Y”. The client application must display following consent text to the user and capture user’s consent:

  
  I provide my consent to share my Aadhaar Number, Date of Birth and Name from my Aadhaar eKYC information with the <Issuer Name> for the purpose of fetching my <Certificate Name> into DigiLocker.

  
  The <Issuer Name> in above text will be replaced by the name of the issuer as returned in Get List of Issuers API. The <Certificate Name> will be replaced by the description returned in Get List of Documents API.
- Other parameters required for fetching a document as listed in paramname field of Get Search Parameters API.

RETURNS
If a document/certificate is found in issuer’s repository with the given criteria, the corresponding URI of the document will be saved in Issued Documents section of DigiLocker and HTTP status code 200 will be returned. This URI will also be returned in response as follows:

- uri indicates URI of the pulled document.

Sample Response:

```json
{
  "uri": "in.gov.cbse-HSCER-201412345678"
}
```

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 user-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:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_token</td>
<td>The access token is invalid</td>
<td>401</td>
</tr>
<tr>
<td>invalid_orgid</td>
<td>The orgid parameter is missing or invalid</td>
<td>400</td>
</tr>
<tr>
<td>invalid_doctype</td>
<td>The doctype parameter is missing or invalid</td>
<td>400</td>
</tr>
<tr>
<td>repository_service_configerror</td>
<td>The issuer service is not available. Please contact DigiLocker support.</td>
<td>500</td>
</tr>
<tr>
<td>pull_response_pending</td>
<td>The details provided above do not exactly match with the details on the certificate. Your request has been forwarded to the Issuer</td>
<td>400</td>
</tr>
</tbody>
</table>
department. The Issuer department will verify the details provided above. If the verification is successful, your certificate will be made available in the Issued Documents section of your DigiLocker.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>uri_exists</td>
<td>This document already exists in your Issued Documents.</td>
<td>400</td>
</tr>
<tr>
<td>record_not_found</td>
<td>No record found in issuer database for given criteria. Please check the details and try again. If problem persists, see the FAQ section for possible causes.</td>
<td>404</td>
</tr>
<tr>
<td>aadhaar_not_linked</td>
<td>To access issued documents, please link your Aadhaar number to your DigiLocker account.</td>
<td>400</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

**DigiLocker Sign up APIs using Aadhaar Demographic Authentication**

This API can be used to sign up a user for DigiLocker account using user’s mobile number, Aadhaar number and demographic details. The sign up APIs can be used in 2 scenarios. First, when the user is online or available to perform an OTP validation and second, when user is not online or present to perform OTP validation. The second scenario will be applicable in case the client application wishes to create a citizen’s DigiLocker account in the absence of the citizen. A verification parameter in the API will determine whether the OTP validation will be performed.

The sign up process in scenario one is a two step process and requires two API calls. The first API accepts user’s mobile number, Aadhaar number, name as in Aadhaar, date of birth as in Aadhaar and gender. It first validates the Aadhaar details using demographic authentication provided by UIDAI. If the validation succeeds, it sends an OTP on the mobile number provided by the user. The second API is to verify the OTP. Once the OTP verification is successful, user account is created with the mobile number as the username. The password is sent to the user via a SMS on the mobile number provided.

The sign up process in scenario two is a one step process and requires call to only the first API. Once the validations are successful, user account is created with the mobile number as the username. The password is sent to the user via a SMS on the mobile number provided. The OTP validation will be performed when the user signs in to the DigiLocker account for the first time.

These APIs do not require user’s authorization using OAuth flow as they do not access information from user’s account. These APIs can be accessed using the Client Secret Key provided to your application from the DigiLocker Partner’s Portal.
SIGN UP
This API is used to validate Aadhaar details and verify the mobile number by generating an OTP. This API call, if successful, will be followed by verify OTP call by the partner application if the user is online or available to perform OTP validation as indicated in verification parameter.

URL STRUCTURE

Production Environment:
https://api.digitallocker.gov.in/public/signup/2/demoauth

HTTP METHOD          POST

PARAMETERS
- **clientid** Provide the client id that was created during the application registration process on Partners Portal.
- **uid** This is the Aadhaar number of the user to sign up for DigiLocker account.
- **name** This is name of the user as mentioned in Aadhaar.
- **dob** This is date of birth of the user as mentioned in Aadhaar in DDMMYYYY format.
- **gender** This is gender of the user as mentioned in Aadhaar. The possible values are M, F, T for male, female and transgender respectively.
- **mobile** This is mobile number of the user.
- **verification** The parameter indicates whether the mobile number provided above should be validated by DigiLocker. If this parameter is ‘Y’, the DigiLocker sends an OTP to verify the mobile number. In this case the client application will call the second API to validate the OTP. The user will be signed on only after successful OTP validation. The OTP validation will be performed when the user signs in for the first time in DigiLocker. This flow should be used when the user account needs to be created in the absence of the user.
- **demoauth** The parameter indicates whether Aadhaar demographic authentication must be successful for creating DigiLocker account. The possible values are ‘Y’ and ‘N’. The value of ‘Y’ will perform Aadhaar demographic authentication and will return errors if any in response. The value of ‘N’ will also perform Aadhaar demographic authentication but will return any error in case of authentication failure. It will create a basic mobile based account for the user. Value ‘N’ should be used when the user account needs to be created in the absence of the user.
- **consent** The consent indicator from the user for performing demographic authentication using Aadhaar details. This Partner Application must capture the user consent for performing the Aadhaar demographic authentication. The possible values are ‘Y’ and ‘N’. The sign up request will be processed only when this indicator is ‘Y’.
- **ts** Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.
**hmac** Provide SHA-256 encrypted value of the client secret, `clientid`, `uid`, `name`, `dob`, `gender`, `mobile`, `verification`, `demoauth`, `consent` and `ts` parameters above concatenated in this sequence (client secret, `clientid`, `uid`, `name`, `dob`, `gender`, `mobile`, `verification`, `demoauth`, `consent`, `ts`). The `hmac` parameter is used by DigiLocker to ensure the data integrity and authentication of the request. Use the Client Secret K generated during the application registration process on Partners Portal as the client secret.

**RETURNS**

If `verification` parameter is 'N' and the all validations are successful, user account will be created and HTTP status code 200 will be returned along with a JSON string containing following fields in response:

- **access_token** The access token that can be used to call the DigiLocker 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.
- **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.

**Sample Response:**
```json
{
  "access_token": "bc125c212a4b03a9a188a858be5a163f379e878a",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope":,
  "refresh_token": "a47ab18c593703e4f83a274694db7422a8cfc8f"
}
```

If `verification` parameter is 'Y', a JSON string containing following fields will be returned in response:

- **mobile** The masked mobile number of the user on which the OTP has been sent.

**Sample Response:**
```json
{
  "mobile": "******3712"
}
```

**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 user-friendly description of the error that occurred.

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

```json
{
    "error":"parameter_missing",
    "error_description":"Aadhaar number missing"
}
```

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
</tbody>
</table>
| invalid_parameter   | One or more of the mandatory parameters is missing or invalid. The error description text will contain one or more of the following error texts:  
• uid parameter is missing or invalid  
• Name parameter is missing or invalid  
• Dob parameter is missing or invalid  
• Gender parameter is missing or invalid  
• Mobile parameter is missing or invalid  
• Verification parameter is missing or invalid  
• Consent parameter is missing or invalid  
• Timestamp parameter is missing or invalid  
• HMAC parameter is missing or invalid | 400                |
| mobile_exists       | A DigiLocker account already exists with this mobile number                      | 400                |
| auth_failure        | The data on Aadhaar does not match with one or more of Aadhaar, Name, DOB and Gender data provided. Aadhaar demographic authentication failure. | 400                |
| unexpected_error    | Internal server error                                                             | 500                |

**Verify OTP**
This API is used to verify the OTP sent by Digi Locker during the sign up API call above.
**URL STRUCTURE**

<table>
<thead>
<tr>
<th>Production Environment:</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://api.digitallocker.gov.in/public/signup/1/demoauthverify">https://api.digitallocker.gov.in/public/signup/1/demoauthverify</a></td>
</tr>
</tbody>
</table>

**HTTP METHOD**  
**POST**

**PARAMETERS**

- **clientid**  
  Provide the client id that was created during the application registration process on Partners Portal.

- **mobile**  
  This is the mobile number of the user provided in the sign up API.

- **otp**  
  The OTP provided by the user.

- **ts**  
  Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.

- **hmac**  
  Provide SHA-256 encrypted value of the client secret, clientid, mobile, otp and ts parameters above concatenated in this sequence (client secret, clientid, mobile, otp, ts). The hmac parameter is used by DigiLocker to ensure the data integrity and authentication of the request. Use the Client Secret Key generated during the application registration process on Partners Portal as the client secret.

**RETURNS**

If the OTP validation succeeds, HTTP status code 200 will be returned along with a JSON string containing following fields in response:

- **access_token**  
  The access token that can be used to call the DigiLocker 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.

- **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.

**Sample Response:**

```json
{
    "access_token": "bc125c212a4b03a9a188a858be5a163f379e878a",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope":
    "refresh_token": "a47ab18c593703e4f83a274694db7422a8cfcb8f"
}
```

**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 user-friendly description of the error that occurred.

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

```json
{
    "error":"invalid_otp",
    "error_description":"The OTP is invalid."
}
```

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
<tr>
<td>parameter_missing</td>
<td>One or more of the mandatory parameters is missing</td>
<td>400</td>
</tr>
<tr>
<td>invalid_ts</td>
<td>Timestamp parameter is invalid</td>
<td>400</td>
</tr>
<tr>
<td>hmac_mismatch</td>
<td>HMAC does not match</td>
<td>400</td>
</tr>
<tr>
<td>invalid_otp</td>
<td>The OTP is invalid</td>
<td>401</td>
</tr>
<tr>
<td>unexpected_error</td>
<td>Internal server error</td>
<td>500</td>
</tr>
</tbody>
</table>

**DigiLocker Meta APIs**

DigiLocker Meta APIs provide information about a user account, issuers, document types and parameters required to pull a document. These APIs do not require user's authorization using OAuth flow as they do not access documents/folders from user's account. This APIs can be accessed using the Client Secret Key provided to your application from the DigiLocker Partner’s Portal. It is strongly recommended that client applications cache this metadata with them to limit the number of calls to these APIs as this information does not change frequently.

**Get List of Issuers**

Returns the list of issuers registered with DigiLocker.

**URL STRUCTURE**

Production Environment:
https://api.digitallocker.gov.in/public/oauth2/1/pull/issuers

**HTTP METHOD** POST

**PARAMETERS**
- **clientid** Provide the client id that was created during the application registration process on Partners Portal.
- **ts** Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.
- **hmac** Provide SHA-256 encrypted value of the client secret, `clientid` and `ts` parameters above concatenated in this sequence (client secret, `clientid`, `ts`). The `hmac` parameter is used by DigiLocker to ensure the data integrity and authentication of the request. Use the Client Secret Key generated during the application registration process on Partners Portal as the client secret.

**RETURNS**

Returns HTTP status code 200 along with a JSON list of issuers containing following fields:

- **orgid** A unique numeric identifier of the issuer organization.
- **issuerid** Unique DigiLocker issuer id as mentioned in the URI.
- **name** Name of the issuer organization.
- **category** A comma separated list of categories that define this issuer. The predefined issuer categories are Central Government, State Government, Education and Insurance.
- **description** A description about the issuer organization and the document it provides.

**Sample Response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "issuers": [ 
        {
            "orgid": "000018",
            "issuerid": "in.gov.cbse",
            "name": "Central Board of Secondary Education, Delhi",
            "category": "Education,Central Government",
            "description": "CBSE (http://www.cbse.nic.in/) is issuing marksheets, passing certificates, migration certificates etc. through DigiLocker. These are either pushed, or can be pulled by students into their DigiLocker accounts."
        },
        {
            "orgid": "000201",
            "issuerid": "in.gov.aktu",
            "name": "APJ Abdul Kalam Technical University, UP",
            "category": "Education,State Government",
            "description": "APJ Abdul Kalam Technical Univeristy, Uttar Pradesh provides the mark sheets of degree certificates of various technical programs."
        }
    ]
}
```
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 user-friendly description of the error that occurred.

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

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

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
</tbody>
</table>
| invalid_parameter   | One or more of the mandatory parameters is missing or invalid. The error description text will contain one or more of the following error texts:  
|                     | - Timestamp parameter is missing or invalid                  | 400                |
|                     | - HMAC parameter is missing or invalid                       |                    |
| unexpected_error    | Internal server error                                        | 500                |

Get List of Documents Provided by an Issuer
Returns a list of documents/certificates issued by an issuer organization registered with DigiLocker.

**URL STRUCTURE**

Production Environment:
https://api.digitallocker.gov.in/public/oauth2/1/pull/doctype

**HTTP METHOD**  POST

**PARAMETERS**

- **clientid**  Provide the client id that was created during the application registration process on Partners Portal.
- **orgid**  The organization id of the issuer in DigiLocker. The API will return the list of document types issued by this issuer. This organization id can be found in the issuer information returned in Get List of Issuers API mentioned earlier. This parameter is sent as a part of the url.

- **ts**  Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.

- **hmac**  Provide SHA-256 encrypted value of the client secret, clientid, orgid and ts parameters above concatenated in this sequence (client secret, clientid, orgid, ts). The hmac parameter is used by DigiLocker to ensure the data integrity and authentication of the request. Use the Client Secret Key generated during the application registration process on Partners Portal as the client secret.

**RETURNS**

Returns HTTP status code 200 along with a JSON list of document types in the response. The JSON containing following fields:

- **doctype**  A 5 character unique document type provided by DigiLocker.
- **description**  The descriptive name of the document type.

**Sample Response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
   "documents": [
      {
         "doctype": "INCER",
         "description": "Income Certificate"
      },
      {
         "doctype": "BTCER",
         "description": "Birth Certificate"
      }
   ]
}
```

**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 user-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:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
</tbody>
</table>
| invalid_parameter   | One or more of the mandatory parameters is missing or invalid. The error description text will contain one or more of the following error texts:  
|                     | • The orgid parameter is missing or invalid                                       | 400                |
|                     | • Timestamp parameter is missing or invalid                                        |                    |
|                     | • HMAC parameter is missing or invalid                                            |                    |
| unexpected_error    | Internal server error                                                             | 500                |

Get Search Parameters for a Document

Returns a list of parameters required to search a document/certificate of an issuer organization registered with DigiLocker. These parameters are used to pull a document from issuer’s repository using Pull Document API mentioned in subsequent section.

URL STRUCTURE

Production Environment:  
https://api.digitallocker.gov.in/public/oauth2/1/pull/parameters

HTTP METHOD POST

PARAMETERS

- **clientid** Provide the client id that was created during the application registration process on Partners Portal.
- **orgid** The organization id for the issuer in DigiLocker. This organization id is returned in Get List of Issuers API mentioned earlier.
- **doctype** A 5 character unique document type provided by DigiLocker. The doctype is returned in Get List of Documents API mentioned earlier.
- **ts** Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.
- **hmac** Provide SHA-256 encrypted value of the client secret, clientid, orgid, doctype and ts parameters above concatenated in this sequence (client secret, clientid, orgid, doctype, ts). The hmac parameter is used by DigiLocker to ensure the data integrity and authentication of the request. Use the Client Secret Key generated during the application registration process on Partners Portal as the client secret.
RETURNS
Returns HTTP status code 200 along with a JSON list of parameters in the response. The parameter list contains following fields:

- **label**  The text label of the parameter. This can be used by the calling application as a label for the input field that accepts the parameter from the user.
- **paramname**  The name of the parameter. This field will be used as the parameter name while calling the Pull Document API.
- **valuelist**  If the parameter takes a value from a set of possible values (dropdown) then this field will provide a comma separated list of values. Otherwise this field will be null.
- **example**  An example of the input value. This can be used by the calling application to show an example of the input value to the user.

Sample Response:

HTTP/1.1 200 OK
Content-Type: application/json

```
{
    "parameters": [
        {
            "label": "Year",
            "paramname": "year",
            "example": "2002",
        },
        {
            "label": "Roll Number",
            "paramname": "rollno",
            "valuelist": null,
            "example": "C525690",
        }
    ]
}
```

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 user-friendly description of the error that occurred.

Sample Error Response:

HTTP/1.1 500 Internal Server Error
Content-Type: application/json;charset=UTF-8
The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
</tbody>
</table>
| invalid_parameter      | One or more of the mandatory parameters is missing or invalid. The error description text will contain one or more of the following error texts:  
  • The orgid parameter is missing or invalid  
  • The doctype parameter is missing or invalid  
  • Timestamp parameter is missing or invalid  
  • HMAC parameter is missing or invalid       | 400                |
| unexpected_error       | Internal server error                                                               | 500                |

**Verify Account**

This API can be used to verify whether a mobile number or Aadhaar number is already registered with DigiLocker.

**URL STRUCTURE**

*Production Environment:*
https://api.digitallocker.gov.in/public/account/2/verify

**HTTP METHOD**

POST

**PARAMETERS**

- **clientid**  
  Provide the client id that was created during the application registration process on Partners Portal.

- **uid**  
  This is the Aadhaar number of the user. DigiLocker will check whether an account exists for this Aadhaar number. Either uid or mobile is required to verify whether an account exists.

- **mobile**  
  This is the mobile number of the user. DigiLocker will check whether an account exists for this mobile number. Either uid or mobile is required to verify whether an account exists.

- **ts**  
  Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.
- **hmac**  Provide SHA-256 encrypted value of the client secret, `clientid, uid/mobie` and `ts` parameters above concatenated in this sequence (client secret, `clientid, uid/mobie, ts`). The hmac parameter is used by DigiLocker to ensure the data integrity and authentication of the request. Use the Client Secret Key generated during the application registration process on Partners Portal as the client secret.

**RETURNS**

The API returns following fields in response:

- **registered**  `true` indicates that the provided Aadhaar/mobile number is already registered and false indicates otherwise.
- **digilockerid**  A unique 36 character DigiLocker Id of the user account.

**Sample Response:**

```
{
  "registered": "true",
  "digilockerid": "123e4567-e89b-12d3-a456-42665440000"
}
```

**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 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": "invalid_parameter",
  "error_description": "Timestamp parameter is missing or invalid"
}
```

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
</tbody>
</table>
| invalid_parameter| One or more of the mandatory parameters is missing or invalid. The error description text will contain one or more of the following error texts:  
  - Either one of uid or mobile number is mandatory  
  - Timestamp parameter is missing                   | 400                |
Push URI to Account

The API can use to push or delete a single URI into Digital Locker using Digi Locker Id acquired using Get User Details API. This API can be used to push the certificate details to Digital Locker as and when a certificate is generated in the issuer system. The issuing departments must register on Digi Locker as a registered Issuer and implement the requisite Issuer APIs as mentioned in Digital Locker Issuer API Specification document prior to pushing certificates using this API.

**URL STRUCTURE**

Production Environment:
https://api.digitallocker.gov.in/public/account/1/pushuri

**HTTP METHOD** POST

**PARAMETERS**

- **clientid** Provide the client id that was created during the application registration process on Partners Portal.
- **digilockerid** This is the Digi Locker Id of the user that was acquired using Get User Details API.
- **uri** This is the unique identifier of the document.
- **doctype** A 5 character unique document type provided by Digi Locker.
- **description** The descriptive name of the document type.
- **docid** A unique number of the document. This id will be unique within the document type issued by the issuer.
- **issuedate** The issue date of the document in DDMMYYYY format.
- **validfrom** (optional) The date from which the document is valid in DDMMYYYY format. This may be same as the issue date.
- **validto** (optional) The expiry date of the document in DDMMYYYY format.
- **action** Action that needs to be taken for the Aadhaar number and URI combination. Possible values are A for adding a new URI, U for updating an already added URI details or D for deleting the URI.
- **ts** Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.
- **hmac** Provide SHA-256 encrypted value of the client secret, clientid, digilockerid, uri, doctype, description, docid, issuedate, validfrom, validto, action and ts parameters above concatenated in this sequence (client secret, clientid, digilockerid, uri, doctype, description, docid, issuedate, validfrom, validto, action and ts). The hmac parameter is used by Digi Locker to ensure the data integrity and authentication of the request. Use the Client Secret Key generated during the application registration process on Partners Portal as the client secret.
**RETURNS**
If the API call succeeds, HTTP status code 200 will be returned. This API does not return any specific values.

**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 user-friendly description of the error occurred.

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

```
{
  "error":"invalid_parameter ",
  "error_description":"Timestamp parameter is missing or invalid"
}
```

The following table lists the possible error codes:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
<tr>
<td>invalid_digilocker_id</td>
<td>The digilockerid parameter is invalid</td>
<td>404</td>
</tr>
</tbody>
</table>
| invalid_parameter      | One or more of the mandatory parameters is missing or invalid. The error description text will contain one or more of the following error texts:  
  - URI parameter is missing or invalid
  - Doctype parameter is missing or invalid
  - Description parameter is missing or invalid
  - Docid parameter is missing or invalid
  - Issuedate parameter is missing or invalid
  - Timestamp parameter is missing or invalid
  - HMAC parameter is missing or invalid
  - URI already exists in this account
  - URI already exists in another account | 400                |
| unexpected_error       | Internal server error                                                             | 500                |
Get Statistics
Returns DigiLocker statistics such as the count of users, authentic documents, issuers and requesters as on a specific date.

URL STRUCTURE

Production Environment:
https://api.digitallocker.gov.in/public/statistics/1/counts

HTTP METHOD
POST

PARAMETERS
- **clientid** Provide the client id that was created during the application registration process on Partners Portal.
- **ts** Provide a timestamp value in UNIX (or POSIX) format in IST time zone in seconds. This timestamp value must not be older than 30 minutes.
- **hmac** Provide SHA-256 encrypted value of the client secret, clientid and ts parameters above concatenated in this sequence (client secret, clientid, ts). The hmac parameter is used by DigiLocker to ensure the data integrity and authentication of the request. Use the Client Secret Key generated during the application registration process on Partners Portal as the client secret.

RETURNS
Returns HTTP status code 200 along with a count of various parameters in JSON format as follows:

- **users** Count of registered users on DigiLocker.
- **authentic_documents** Count of authentic documents available through DigiLocker.
- **issuers** Count of issuer organizations registered on DigiLocker.
- **requestors** Count of requester organizations registered on DigiLocker.
- **count_as_on** The date on which this statistics is generated.
- **monthwise_registrations** List of last 12 months cumulative user registrations
  - **id** Unique id of the list item
  - **month** Month of the year in numeric format with January as 1
  - **year** Year in YYYY format
  - **count** Count of cumulative user registrations in the given month
- **yearwise_authentic_documents** List of cumulative year-wise counts of authentic documents in DigiLocker
  - **id** Unique id of the list item
  - **year** Year in YYYY format
  - **count** Count of cumulative user authentic documents in the given year

Sample Response:

HTTP/1.1 200 OK
Content-Type: application/json

{
    "users":"25873490",
    "authentic_documents":"3520475277",
    "issuers":"134",
    "requesters":"45",
    "count_as_on":"31072019",
    "monthwise_registrations": [
        {
            "id":"18496",
            "month":"5",
            "year":"2018",
            "count":"124794"
        },
        {
            "id":"18497",
            "month":"6",
            "year":"2018",
            "count":"135455"
        }
    ],
    "yearwise_authentic_documents": [
        {
            "id":"18347",
            "year":"2016",
            "count":"10134567"
        },
        {
            "id":"18350",
            "year":"2017",
            "count":"1000548093"
        },
        {
            "id":"18354",
            "year":"2018",
            "count":"2410543684"
        },
        {
            "id":"18567",
            "year":"2019",
            "count":"3520475277"
        }
    ]
}
**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 user-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:

<table>
<thead>
<tr>
<th>Error</th>
<th>error_description</th>
<th>HTTP Response Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>invalid_client_id</td>
<td>The client_id parameter is invalid</td>
<td>401</td>
</tr>
</tbody>
</table>
| invalid_parameter   | One or more of the mandatory parameters is missing or invalid. The error description text will contain one or more of the following error texts:  
  - Timestamp parameter is missing or invalid  
  - HMAC parameter is missing or invalid     | 400                |
| unexpected_error    | Internal server error                                                              | 500                |