



Integrate DigiLocker for Age Verification

To verify users' age via DigiLocker, first register as an **Authorized Partner** on the DigiLocker portal (API Setu). This gives you a `client_id` and `client_secret`, and lets you configure your redirect URI and required scopes ¹. In your OAuth2 setup, include the `openid` scope so that DigiLocker returns an ID token with user profile claims (including date of birth) ² ³. For example, when directing the user to DigiLocker's authorization endpoint (<https://api.digitallocker.gov.in/public/oauth2/1/authorize>), include query parameters like: `response_type=code`, your `client_id`, `redirect_uri`, a `state` value, and your **PKCE** code challenge. Use `code_challenge_method=S256` with a Base64URL-encoded SHA256 hash of your `code_verifier` ⁴ ⁵. This enforces a secure PKCE flow (required for mobile/web clients) and ensures an authorization code is issued to your app ⁶ ⁴. A sample authorization URL might look like:

```
https://api.digitallocker.gov.in/public/oauth2/1/authorize?  
  response_type=code  
  &client_id=YOUR_CLIENT_ID  
  &redirect_uri=https://yourapp.example.com/callback  
  &state=xyz123  
  &scope=openid  
  &code_challenge=<BASE64URL(sha256(code_verifier))>  
  &code_challenge_method=S256
```

Make sure to also register this exact redirect URI in the DigiLocker partner console ⁷. (You can set up a test page or route in your app to catch the redirect, e.g. `/callback`.)

- **Generate PKCE values:** Create a high-entropy `code_verifier` string (43–128 chars using [A-Z,a-z,0-9,-_,~]) and compute its SHA256 hash. Base64-url-encode the hash (no padding) to get `code_challenge` ⁴ ⁵. Keep `code_verifier` secret on the client side.
- **Scopes:** At minimum use `scope=openid` so DigiLocker returns an `id_token` with user data ² ³. You can also add other scopes (e.g. `files.issueddocs` to fetch documents), but for age/identity, `openid` is essential. The access token response will list scopes like `openid` among others ⁸.

Exchange Code for Tokens

After user consent, DigiLocker redirects to your callback with an authorization `code`. Now exchange that code for tokens. Call DigiLocker's token endpoint (POST to <https://digilocker.meripehchaan.gov.in/public/oauth2/2/token>) with form parameters:

- `grant_type=authorization_code`
- `code` (the code from redirect)
- `redirect_uri` (must match the original URI)

- `client_id` and `client_secret` (or use HTTP Basic auth) [9](#)
- `code_verifier` (the original PKCE verifier) [9](#) [10](#)

For example:

```
POST /oauth2/2/token HTTP/1.1
Host: digilocker.meripehchaan.gov.in
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&code=AUTH_CODE_HERE
&redirect_uri=https://yourapp.example.com/callback
&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_SECRET
&code_verifier=ORIGINAL_CODE_VERIFIER
```

In response you'll get a JSON containing `access_token`, `expires_in`, etc. Since you requested `openid`, the response will also include an `id_token` (a JWT) and possibly a `refresh_token` [11](#) [12](#). For example, a successful response might contain:

```
{
  "access_token": "abc123...",
  "expires_in": 3600,
  "token_type": "Bearer",
  "scope": "openid files.issueddocs partners.PANCR partners.DRVLC",
  "id_token": "<JWT with user info>",
  "consent_valid_till": 1684731048
}
```

The `id_token` (if present) carries the user's DigiLocker profile claims as a JWT [3](#). (It will contain fields like name, gender, DigiLocker ID, etc.)

Extract DOB and Calculate Age

The user's **date of birth** is provided by DigiLocker in the profile data. In the response above, the `id_token` will include `dob` in DDMMYYYY format (as registered in DigiLocker) [13](#). If for any reason you only have an `access_token`, you can also call the "Get User Details" API to retrieve the user profile (which likewise returns `dob`). In either case, **capture the DOB value**.

Next, compute the user's age from the DOB. For example, parse the `DDMMYYYY` string into day/month/year and compare to the current date. Subtract birth year from current year and adjust if the birth date hasn't occurred yet this year. If the resulting age is **18 or above**, classify as adult; if **below 18**, classify as minor.

(DigiLocker itself does *not* perform this age logic — it simply hands you the raw DOB ¹³. Your app must enforce the 18+ rule.)

- *No special DigiLocker flag is needed for “age verification.”* You will rely solely on the retrieved DOB to check age. In other words, DigiLocker supplies the data (DOB) but your system applies the rule (≥ 18 adult, <18 minor).

Testing Your Integration

Before going live, test thoroughly. API Setu provides a **Sandbox** environment for DigiLocker where you can use fake/sample data safely ¹⁴. In sandbox mode, use the designated test endpoints or domains (e.g. via `sandbox.api-setu.in`) and sample credentials. The sandbox portal lets you simulate API calls with pre-filled demo data ¹⁴. Build a test page or UI that triggers the OAuth flow and handles the callback (for example, a “Login with DigiLocker” button). Use your sandbox client ID/secret and sample user account to verify each step: authorization, token exchange, and profile fetch. Confirm that the `dob` is correctly retrieved and your age-calculation logic categorizes users as intended.

Summary of Key Steps

- **Add Authorized Partner:** Register on DigiLocker’s Partner (API Setu) portal to get your application’s credentials ¹. Specify your redirect URI and choose needed OAuth scopes (e.g. `openid`).
- **OAuth2 Flow with PKCE:** Implement the standard auth-code flow. Direct users to DigiLocker’s `/authorize` URL with your `client_id`, `scope=openid`, `code_challenge`, etc. DigiLocker supports PKCE for enhanced security ⁶.
- **Token Exchange:** POST the auth code (with `client_secret` and `code_verifier`) to `/oauth2/2/token` ⁹ ¹⁰. Retrieve the `access_token` and `id_token`.
- **Get DOB:** Extract `dob` from the response (it’s in DDMMYYYY format ¹³).
- **Compute Age:** Calculate age from `dob` and apply your minor/adult logic (≥ 18 adult, <18 minor).
- **Testing:** Use the API Setu sandbox and a test page or callback to ensure the flow works end-to-end ¹⁴.

By following these steps, your AI agent can integrate DigiLocker into your platform to fetch a user’s date-of-birth, calculate their age, and classify them as adult or minor without relying on any built-in age flag.

Sources: DigiLocker Authorized Partner API docs ¹³ ⁷ ⁴ ² ⁹ ¹⁰ ³ and API Setu Sandbox documentation ¹⁴.

¹ cf-media.api-setu.in

<https://cf-media.api-setu.in/resources/Partners-SOP.pdf>

² ³ ⁸ ⁹ ¹⁰ ¹¹ ¹² Microsoft Word - Digital Locker Authorized Partner API Specification v2.2.docx
<https://cf-media.api-setu.in/resources/DigitalLocker-AuthorizedPartnerAPI-Specificationv2.2.pdf>

⁴ ⁵ ⁶ ⁷ ¹³ Authorized Partner API Specification

<https://img1.digitallocker.gov.in/assets/img/Digital%20Locker%20Authorized%20Partner%20API%20Specification%20v1.11.pdf>

¹⁴ Sandbox : APISetu testing environment

<https://sandbox.api-setu.in/>