MAuthN OAuth2 API Documentation

This document outlines the OAuth2 API endpoints for authorization, token management, and user data retrieval. These endpoints follow standard OAuth2 specifications but do not require `client_id` or `client_secret` as authentication is not needed in this implementation. All endpoints assume a stateless, standard OAuth2 flow.

Base URL: `https://mauthn.mukham.in`

Endpoints

1. `/oauth/authorize`

Description: Opens the authorization web page where the user grants access to their resources.

- **Method**: `GET`
- **Path**: `/oauth/authorize`
- **Purpose**: Initiates the OAuth2 authorization code flow by redirecting the user to a web page for consent.

Query Parameters

Parameter	Type	Required	Description
`response_type`	String	Yes	Must be set to `code` for authorization code flow.
`redirect_uri`	String	Yes	The URI to redirect the user to after authorization (must be registered).
`scope`	String	Optional	Space-separated list of scopes (e.g., `read profile`).
`state`	String	Optional	A unique value to prevent CSRF attacks; returned unchanged in the response.

Example Request

```
GET /oauth/authorize?
response_type=code&redirect_uri=https://app.example.com/callback&scope=read%20profile&state=xyz123
```

Response

- Redirects to the `redirect_uri` with the following parameters appended:
  - `code`: The authorization code (if approved).
  - `state`: The original `state` value (if provided).
- **Success Redirect**:

  ```
  https://app.example.com/callback?code=abc123&state=xyz123
  ```

- **Error Redirect** (if user denies or an error occurs):

  ```
  https://app.example.com/callback?error=access_denied&state=xyz123
  ```

---

2. `/oauth/token`

Description: Exchanges an authorization code for a bearer token or uses a refresh token to obtain a new bearer token.

- **Method**: `POST`
- **Path**: `/oauth/token`
- **Purpose**: Completes the authorization code flow or refreshes an existing token.

Request Parameters (Form Data)

Parameter	Type	Required	Description
`grant_type`	String	Yes	`authorization_code` for code exchange or `refresh_token` for token refresh.
`code`	String	Yes*	The authorization code from `/oauth/authorize` (*if `grant_type=authorization_code`).
`redirect_uri`	String	Yes*	Must match the `redirect_uri` from `/oauth/authorize` (*if `grant_type=authorization_code`).
`refresh_token`	String	Yes*	The refresh token (*if `grant_type=refresh_token`).

Example Requests

1. **Exchange Authorization Code**:

   ```
   POST /oauth/token
   Content-Type: application/x-www-form-urlencoded

   grant_type=authorization_code&code=abc123&redirect_uri=https://app.example.com/callback
   ```

2. **Refresh Token**:

   ```
   POST /oauth/token
   Content-Type: application/x-www-form-urlencoded

   grant_type=refresh_token&refresh_token=def456
   ```

Response (JSON)

- **Success** (HTTP 200):

  ```
  {
    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "token_type": "Bearer",
    "expires_in": 3600,
    "refresh_token": "def456",
    "scope": "read profile"
  }
  ```

- **Error** (HTTP 400):

  ```
  {
    "error": "invalid_grant",
    "error_description": "The authorization code is invalid or expired."
  }
  ```

---

3. `/oauth/revoke`

Description: Revokes a bearer token, rendering it invalid for future use.

- **Method**: `POST`
- **Path**: `/oauth/revoke`
- **Purpose**: Allows the client to revoke an access or refresh token.

Request Parameters (Form Data)

Parameter	Type	Required	Description
`token`	String	Yes	The access or refresh token to revoke.

Example Request

```
POST /oauth/revoke
Content-Type: application/x-www-form-urlencoded

token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Response

- **Success** (HTTP 200):

  ```
  {}
  ```

- **Error** (HTTP 400):

  ```
  {
    "error": "invalid_token",
    "error_description": "The token is already revoked or invalid."
  }
  ```

---

4. `/userinfo`

Description: Returns user data as a GET request using the bearer token.

- **Method**: `GET`
- **Path**: `/userinfo`
- **Purpose**: Retrieves basic user information for the authenticated user.

Headers

Header	Type	Required	Description
`Authorization`	String	Yes	Bearer token (e.g., `Bearer <token>`).

Example Request

```
GET /userinfo
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Response (JSON)

- **Success** (HTTP 200):

```
{
    "date-of-birth": "2002-01-01",
    "email": "demo@example.com",
    "name": "John Doe",
    "state": "63937ecfa4015c2c3331a41313d20078"
}
```

- **Error** (HTTP 401):

```
{
  "error": "invalid_token",
  "error_description": "The access token is invalid or expired."
}
```

---

5. `/userimage`

Description: Returns user data including an image, using the bearer token.

- **Method**: `GET`
- **Path**: `/userimage`
- **Purpose**: Retrieves user information along with a profile image.

Headers

Header	Type	Required	Description
`Authorization`	String	Yes	Bearer token (e.g., `Bearer <token>`).

Example Request

```
GET /userimage
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Response (JSON)

- **Success** (HTTP 200):

```
{
    "date-of-birth": "2002-01-01",
    "email": "demo@example.com",
    "name": "John Doe",
    "image": "<Base64 Encoded image>"
    "state": "63937ecfa4015c2c3331a41313d20078"
}
```

- **Error** (HTTP 401):

```
{
  "error": "invalid_token",
  "error_description": "The access token is invalid or expired."
}
```

Notes

- **Token Lifetime**: Access tokens expire in 600 seconds (10 minutes) unless revoked or refreshed.
- **Scopes**: Supported scopes include `user`, `read` and `profile`. Additional scopes can be added as needed.
- **Security**: Use HTTPS for all requests to ensure data integrity and confidentiality.

If you’d like this formatted differently (e.g., as an OpenAPI spec, Markdown file, or with a visual diagram), let me know! Also, if you want an image (like a flow diagram), please confirm, and I’ll generate one for you.