API Documentation

Overview

This document provides a comprehensive guide to the REST API endpoints offered by the AutoSync Cloud Service. Each endpoint is described with its purpose, request structure, parameters, and response formats.

- **Base URL:**
  - Development: `http://localhost:3000/api`
  - Staging: `https://staging.example.com/api`
  - Production: `https://api.example.com/api`

---

1. Authentication Endpoints

1.1 Login

- **Endpoint:** `POST /auth/login`
- **Description:** Authenticates the user and returns a JWT token.
- **Headers:**

  ```
  json
  ```

  Copy code

  `{ "Content-Type": "application/json" }`
- **Request Body:**

  ```
  json
  ```

  Copy code

  `{ "username": "user@example.com", "password": "yourpassword" }`
- **Response:**
  - **Success (200):**

    ```
    json
    ```

    Copy code

    `{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." }`
  - **Error (401):**

    ```
    json
    ```

    Copy code

    `{ "error": "Invalid credentials" }`

1.2 Token Validation

- **Endpoint:** `GET /auth/validate`
- **Description:** Validates the provided JWT token.
- **Headers:**

```
  json
  ```

  Copy code

  `{ "Authorization": "Bearer <token>" }`

- **Response:**
  - **Success (200):**

    ```
    json
    ```

    Copy code

    `{ "status": "valid" }`

  - **Error (401):**

    ```
    json
    ```

    Copy code

    `{ "error": "Token is invalid or expired" }`

---

2. File Sync Endpoints

2.1 Upload File

- **Endpoint:** `POST /files/upload`
- **Description:** Uploads a file to the cloud storage.
- **Headers:**

  ```
  json
  ```

  Copy code

  `{ "Authorization": "Bearer <token>", "Content-Type": "multipart/form-data" }`

- **Request Body (FormData):**
  - `file`: The file to be uploaded.
  - `metadata`: JSON string for optional metadata.
- **Response:**
  - **Success (201):**

    ```
    json
    ```

    Copy code

    `{ "fileId": "12345", "status": "uploaded" }`

  - **Error (400):**

    ```
    json
    ```

    Copy code

    `{ "error": "Invalid file format" }`

2.2 Sync Status

- **Endpoint:** `GET /files/sync-status`
- **Description:** Retrieves the sync status for all files.

- **Headers:**

  ```
  json
  ```

  Copy code

  `{ "Authorization": "Bearer <token>" }`

- **Response:**
  - **Success (200):**

    ```
    json
    ```

    Copy code

    `[ { "fileId": "12345", "status": "synced" }, { "fileId": "67890", "status": "syncing" } ]`

---

3. Conflict Resolution Endpoints

3.1 List Conflicts

- **Endpoint:** `GET /files/conflicts`
- **Description:** Lists all unresolved sync conflicts.
- **Headers:**

  ```
  json
  ```

  Copy code

  `{ "Authorization": "Bearer <token>" }`

- **Response:**
  - **Success (200):**

    ```
    json
    ```

    Copy code

    `[ { "fileId": "12345", "conflictType": "version-mismatch", "details": "Two versions of the file exist." } ]`

3.2 Resolve Conflict

- **Endpoint:** `POST /files/resolve`
- **Description:** Resolves a file sync conflict.
- **Headers:**

  ```
  json
  ```

  Copy code

  `{ "Authorization": "Bearer <token>", "Content-Type": "application/json" }`

- **Request Body:**

  ```
  json
  ```

  Copy code

  `{ "fileId": "12345", "resolution": "use-server-version" }`

- **Response:**

- **Success (200):**

  ```
  json
  ```

  Copy code

  `{ "fileId": "12345", "status": "resolved" }`

---

4. Pagination and Filtering

Most endpoints support pagination and filtering for large datasets.

- **Query Parameters:**
  - `limit` : Number of records per page.
  - `offset` : Starting point for the records.
  - `filter` : JSON string for advanced filtering.

Example:

```
bash
```

Copy code

`GET /files/sync-status?limit=10&offset=0&filter={"status":"syncing"}`

---

5. Rate Limiting

To ensure fair usage, the API enforces rate limits:

- **Authenticated Users:** 1000 requests/hour.
- **Unauthenticated Users:** 100 requests/hour.

Responses exceeding the limit will return a `429 Too Many Requests` error.

Retry-After Header Example:

```
json
```

Copy code

`{ "error": "Too many requests", "retryAfter": "60" }`

---

6. API Workflow Diagram

(Below is a diagram showing how various API endpoints interact in typical use cases.)