

Use Case Description for Your Dashboard Application

Overview: This dashboard application features a REST backend and a simple UI frontend, with backend documentation provided by OpenAPI. The application is designed to store simple data and files, which can be viewed and managed through the UI.

Key Functionalities:

1. File Management:

- **Upload File:**

- Endpoint: `POST /api/file`
- Request Body: Accepts a binary file.
- Response: Returns the uploaded file's ID.

- **Retrieve All Files:**

- Endpoint: `GET /api/file`
- Response: Returns a list of all files stored in the system.

- **Retrieve Specific File:**

- Endpoint: `GET /api/file/{fileId}`
- Parameters: `fileId` (string)
- Response: Returns the content of the specified file.

- **Delete Specific File:**

- Endpoint: `DELETE /api/file/{fileId}`
- Parameters: `fileId` (string)
- Response: Confirms file deletion.

2. Data Management:

- **Create New Data Entry:**

- Endpoint: `POST /api/data/{domain}/{category}/{entry}`
- Parameters: `domain`, `category`, `entry` (all strings)
- Request Body: Data structure defined by `DataRequest`.
- Response: Confirms data creation.

- **Delete Data Entry:**

- Endpoint: `DELETE /api/data/{domain}/{category}/{entry}`
- Parameters: `domain`, `category`, `entry` (all strings)
- Optional Query: `softDelete` (default `true`)
- Response: Confirms data deletion.
- **Retrieve All Data Entries:**
 - Endpoint: `GET /api/data`
 - Response: Returns a list of all data entries.
- **Retrieve Domain-Specific Data:**
 - Endpoint: `GET /api/data/{domain}`
 - Parameters: `domain` (string)
 - Response: Returns a list of data entries for the specified domain.
- **Delete Category:**
 - Endpoint: `DELETE /api/data/{domain}/{category}`
 - Parameters: `domain`, `category` (both strings)
 - Response: Confirms category deletion.

3. Data Hierarchy Management:

- **Create Data at Different Levels:**
 - Multiple endpoints to create hierarchical data entries at various levels, specified by path parameters such as `level0`, `level1`, etc.

Example Scenarios:

1. Uploading a File:

- A user uploads a new file using the UI, which triggers a `POST` request to `/api/file`. The backend stores the file and returns an ID for future reference.

2. Viewing All Files:

- On the dashboard, a user clicks to view all files, which triggers a `GET` request to `/api/file`. The UI displays the list of files returned by the backend.

3. Creating a Data Entry:

- A user inputs new data into the form and submits it. This action triggers a **POST** request to `/api/data/{domain}/{category}/{entry}`, storing the data in the backend.

4. Deleting a Specific Data Entry:

- From the UI, the user selects a data entry to delete. This sends a **DELETE** request to `/api/data/{domain}/{category}/{entry}`, which removes the data.