Interface (Files Block)

Overview

The `FileTable` component is a Vue component that provides a user interface for managing and uploading files. It allows users to view existing files in a table format, filter and search through files, and upload new files to specific locations.

Key Features

1. Table view of existing files with sorting and filtering capabilities
2. File upload functionality with source and data type selection
3. Progress bar for file uploads
4. Refresh functionality to update the file list

Component Structure

Data Properties

- `pagination`: Controls table pagination
- `search`: Stores the current search term
- `selectedType`, `selectedSource`, `selectedStatus`: For filtering the file table
- `typeOptions`, `sourceOptions`, `statusOptions`: Options for the filter dropdowns
- `rows`: Stores the main data displayed in the table
- `columns`: Defines the structure of the table
- `uploadView`: Toggles between file table view and upload view
- `selectedDataTypeUpload`, `selectedSourceUpload`: For selecting upload options
- `dataTypeUploadOptions`, `sourceUploadOptions`: Options for upload dropdowns
- `filesToUpload`: Stores files selected for upload
- `uploadInProgress`, `uploadProgress`: Manages upload state and progress

Computed Properties

- `filteredData`: Returns the filtered and searched data for the table

Methods

1. `changeToUploadFiles()`: Toggles between file table view and upload view
2. `refreshFiles()`: Refreshes the file list by triggering a workflow
3. `startWorkflow(user_code, payload)`: Initiates a new workflow
4. `checkWorkflowStatus(workflowID, resolve, reject)`: Checks the status of a running workflow
5. `fetchFiles()`: Fetches and processes the list of files
6. `handleFileUpload(event)`: Handles file selection for upload
7. `uploadFiles()`: Manages the file upload process
8. `getPath()`: Generates the upload path based on selected options
9. `uploadFile(file, path, currentFileIndex, totalFiles)`: Uploads a single file

Usage Guide

Viewing Files

1. The component displays a table of files with columns for File, Type, Source, and Status
2. Use the search bar to filter files by any field
3. Use the Data Type, Source, and Status dropdowns to further filter the table

Uploading Files

1. Click the '+' button to switch to the upload view
2. Select the Source and Data Type for the upload
3. Choose files to upload using the file input
4. Click "Upload Files" to start the upload process
5. A progress bar will display the upload progress

Refreshing Data

- Click the refresh button (circular arrow icon) to fetch the latest data from the server

API Integration

The component interacts with several API endpoints:

1. File listing: `${apiBaseUrl}/api/v1/explorer/view/?path=components/data_files.json`
2. Workflow start: `${apiBaseUrl}/workflow/api/workflow/run-workflow/`
3. Workflow status check: `${apiBaseUrl}/workflow/api/workflow/${workflowID}/`
4. File upload: `${apiBaseUrl}/api/v1/explorer/upload/`

Notes for Developers

- The component uses custom icons (`IconPlus`, `IconRefresh`) which should be properly imported
- Authorization is handled by the `authorization()` function from `@/utils/customFetch`
- The component relies on the Quasar framework for UI components (e.g., `q-table`, `q-input`, `q-select`)
- File uploads include a timestamp in the filename to ensure uniqueness

Potential Enhancements

1. Implement pagination on the server-side for better performance with large datasets
2. Add more robust error handling and user notifications for upload failures
3. Implement file chunking for large file uploads (commented out in the current version)
4. Add a preview feature for uploaded files
5. Implement drag-and-drop functionality for file uploads

Revision #1
Created 30 August 2024 14:44:28 by Anton Petrushkaneki
Updated 30 August 2024 14:44:48 by Anton Petrushkaneki