

Interface

Overview

This `standard-workflow-interface` module provides a comprehensive interface for managing states and files within a workflow system. The interface is divided into two main components: `StateTable` and `FileTable`, which are displayed side by side in the main application view.

The screenshot displays the user interface for the `standard-workflow-interface` module. It is divided into two main sections: `States` (left) and `Inputs` (right).

States Table:

Name	Items (To do)	Items (Error)	Items	Workers
+ Exante Historical - Step 1 (Download): positions, transactions	0	0	15	3
+ Exante Historical - Step 2 (Preprocess): positions, transactions	0	1	10	3
+ Exante Historical - Step 3 (Download): instruments	0	0	7	3
+ Exante Historical - Step 4 (Preprocess): instruments	0	0	8	2
+ Exante Historical - Step 5 (Import): client data	0	0	24	10
+ Exante Historical - Step 6 (Download): market data	0	0	11	3
+ Exante Historical - Step 7 (Preprocess): market data	0	0	48	3
+ Exante Historical - Step 8 (Import): market data	0	0	48	3

Inputs Table:

File	Type	Source	Status
AAPL.NASDAQ.json	instruments	exante	initial_data
B.ICE.K2024.json	instruments	exante	initial_data
B.ICE.Z2024.json	instruments	exante	initial_data
BLK.CME.H2024.json	instruments	exante	initial_data
BTC.USD.json	instruments	exante	initial_data
IT0005240830.EUR.json	instruments	exante	initial_data
USD.TRY.E.FX.json	instruments	exante	initial_data
aapl.nasdaq_2024-03-15_2024-03-29.json	prices	exante	initial_data
aapl.nasdaq_2024-03-29_2024-04-30.json	prices	exante	initial_data
aapl.nasdaq_2024-04-30_2024-05-31.json	prices	exante	initial_data
aapl.nasdaq_2024-05-31_2024-06-28.json	prices	exante	initial_data
aapl.nasdaq_2024-06-28_2024-07-31.json	prices	exante	initial_data
b.ice.k2024_2024-03-15_2024-03-29.json	prices	exante	initial_data
b.ice.z2024_2024-03-15_2024-03-29.json	prices	exante	initial_data
b.ice.z2024_2024-03-29_2024-04-30.json	prices	exante	initial_data
b.ice.z2024_2024-04-30_2024-05-31.json	prices	exante	initial_data
b.ice.z2024_2024-05-31_2024-06-28.json	prices	exante	initial_data
b.ice.z2024_2024-06-28_2024-07-31.json	prices	exante	initial_data
blk.cme.h2024_2024-03-15_2024-03-29.json	prices	exante	initial_data
btc.usd_2024-03-15_2024-03-29.json	prices	exante	initial_data

Main Application Structure

The main application is defined in `App.vue` and consists of two primary components:

1. `StateTable`
2. `FileTable`

These components are laid out side by side using a flex container for optimal screen utilization.

Key Components

1. StateTable

The StateTable component manages and displays workflow states.

[Interface \(States & Input blocks\)](#)

Key features:

- Displays a list of workflow states
- Allows filtering and searching of states
- Provides detailed view of each state
- Enables starting workflows and managing state status

2. FileTable

The FileTable component manages and displays files within the system.

[Interface \(Files Block\)](#)

Key features:

- Displays a list of files with their types, sources, and statuses
- Allows filtering and searching of files
- Provides file upload functionality
- Enables refreshing the file list

User Interface

The interface is divided into two main sections:

1. **Left Side (StateTable):**
 - Tabs for switching between "States" and "Inputs"
 - Search and filter options
 - Table displaying state information
 - Expandable rows for detailed state information
2. **Right Side (FileTable):**
 - Table displaying file information
 - Search and filter options
 - File upload functionality

Common Features

- Both components feature a refresh button (circular arrow icon) to update their respective data.
- The interface uses a consistent design language, with similar table layouts and filter options.

API Integration

Both components interact with backend APIs for data retrieval and manipulation. The API calls are managed through custom fetch functions defined in `customFetch.js`.

Development Notes

- The project uses Vue 3 with the Composition API.
- Quasar framework is used for UI components.
- Custom icons are used and should be properly imported in each component.
- API base URL is dynamically determined based on the current URL.

Revision #4

Created 2024-08-30 14:48:24 UTC by Anton Petrushkaneki

Updated 2025-01-20 13:46:10 UTC by Anton Petrushkaneki