Wombat Dungeon Master NFT Viewer Documentation

OVERVIEW

The Wombat Dungeon Master NFT Viewer is a React application that displays NFTs owned by a given EOS account. Users can search for their NFTs which are then displayed in cards grouped by collection.

PROJECT STRUCTURE

The project follows a typical React project structure created via Create React App with TypeScript:

- src/ - Contains the source code for the application.
  - ✓ components/ - Holds the reusable React components.
  - ✓ api/ - Contains functions for API calls.
  - ✓ types/ - Includes TypeScript type definitions specific to the application.
  - ✓ App.tsx - The main React component that renders the search form and NFT cards.
  - ✓ index.tsx - Entry point for the React application.
- public/ - Public files like the HTML template and icons.
- package.json - Project metadata and dependencies.

COMPONENTS

NFTFORM

- Purpose: Allows the user to input an EOS account name and submit it for searching NFTs.
- Props:
  - ✓ onSearch: A callback function that gets called when the user submits the form.

- Usage: Rendered inside App.tsx to provide a search interface.

NFTCARD

- Purpose: Displays information about an NFT.
- Props:
  - ✓ nft: An object containing the NFT's data.
- Usage: Rendered by App.tsx for each NFT after a successful search.

API FUNCTIONS

FETCHNFTSFORACCOUNT

- Purpose: Queries the AtomicAssets API for NFTs owned by the specified EOS account.
- Parameters:
  - ✓ accountName: A string representing the EOS account name.
- Returns: A promise that resolves to the list of NFTs owned by the account.

TYPES

INFTASSET

- Purpose: Represents the structure of an NFT asset as returned by the AtomicAssets API.
- Fields:
  - ✓ contract: The contract identifier for the NFT.
  - ✓ collection: An object containing details about the NFT's collection.
  - ✓ data: An object containing the NFT's data, such as name and image.
  - ✓ template_mint: A string representing the mint number of the NFT.

INFTASSETSRESPONSE

- Purpose: Represents the structure of the response from the AtomicAssets API when querying for NFTs.

- Fields:
  - ✓ success: A boolean indicating the success of the API call.
  - ✓ data: An array of INftAsset representing the NFTs.

RUNNING THE PROJECT

Developers can run the project by cloning the repository and installing the dependencies with npm or yarn. The application can be started in development mode with npm start or yarn start.

ADDITIONAL NOTES

The project uses Material-UI components for styling.

Axios is used for making API requests to the AtomicAssets API.

CORS should be enabled for the API requests to succeed.