Client Side Technologies Homework

Game Of Thrones

Name : Abderrahmen Malouche

Neptun Code : BCZ07V

Developer Documentation

Introduction

The Game of Thrones application provides an interactive interface to explore the world of the popular book series "A Song of Ice and Fire" by George R.R. Martin. The application leverages the Ice And Fire API (https://anapioficeandfire.com/) to fetch data related to books, characters, and houses within the series. Users can search and filter through books, characters, and houses, and view detailed information about each entity. From the detailed view, users can navigate to related entities, ensuring an engaging and seamless browsing experience.

Architecture

The application follows a React component-based architecture, ensuring modularity and reusability.

The primary components are:

- **Books:** Displays a list of books with search and filter functionality.
- **SingleBook: S**hows detailed information about a selected book.

- **Characters:** Displays a list of characters with search and filter functionality.
- **SingleCharacter:** Shows detailed information about a selected character.
- **Houses:** Displays a list of houses with search and filter functionality.
- **SingleHouse:** Shows detailed information about a selected house.
- **Home:** Displays the homepage with introductory content.

Architecture Diagram

Components and Classes

Components

1. **Books:**
    - Fetches and displays a list of books.
    - Allows searching and filtering of books.
2. **SingleBook:**
    - Fetches and displays detailed information about a selected book.
3. **Characters:**
    - Fetches and displays a list of characters.
    - Allows searching and filtering of characters.
    - Includes pagination for handling large datasets.
4. **SingleCharacter:**
    - Fetches and displays detailed information about a selected character.

5. **Houses:**
    - Fetches and displays a list of houses.
    - Allows searching and filtering of houses.
    - Includes pagination for handling large datasets.
6. **SingleHouse:**
    - Fetches and displays detailed information about a selected house.
7. **Home:**
    - Displays the homepage with a banner or introductory content.

Helper Classes

RelatedElements:

Displays related characters and houses for a given entity.

Uses Swiper for improved UI and navigation experience.

Client-Server Communication

The application uses the Ice And Fire API to fetch data about books, characters, and houses. Axios is used for making API requests due to its simplicity and ease of use.

Example: Fetching Books

Here is an example of how the application fetches a list of books and displays them.

API Call Process

1. **Initiating the Call**:
    - The **useEffect** hook in the **Books** component calls the **fetchBooks** function when the component mounts.

2. **Making the Request**:
    - **fetchBooks** function uses Axios to make a GET request to the Ice And Fire API endpoint for books: **https://anapioficeandfire.com/api/books**.
3. **Handling the Response**:
    - Upon a successful response, the list of books is stored in the component's state using **setBooks**.
4. **Displaying the Data**:
    - The list of books is filtered based on the search query and displayed using the **BookPreview** component. Each book is wrapped in a **NavLink** to enable navigation to the detailed view (**SingleBook**).

This process ensures that the application dynamically fetches and displays data, providing an interactive experience for users. Each component follows a similar pattern for fetching and displaying data, allowing for consistency and maintainability.