Client-Side Assignment

Game of thrones: Saga

Name: Bikash Kumar Mahanti
Neptun: O8F1L9

Table of Contents

Introduction

The objective of this assignment is to develop a client-side application that provides information about the Game of Thrones book series. This application will utilize an external API to fetch data through HTTP requests.

Aim

The project aims to serve as a comprehensive directory for the renowned book series "Game of Thrones." The user interface (UI) is designed to present extensive data and information regarding each book in the series.

Objective

The objective is to develop an Angular-based application to fulfill the project requirements.

Main functionalities

1. **Display of Books and Related Information:**
    - The application displays all books from the "Game of Thrones" series along with comprehensive information such as title, author, number of pages, ISBN code, etc.
    - Additionally, the application provides a list of all characters mentioned in each book.
2. **Clickable Character Links:**
    - The list of characters is presented as clickable links.
    - Clicking on a character link navigates the user to a detailed description page of the selected character.
3. **List of Houses with Active Links:**
    - The application presents a list of all houses featured in the "Game of Thrones" series.
    - Links associated with each house are active, allowing users to navigate back and forth between different tabs seamlessly.

Main Classes

Main Components:

1. **display-book:**
   - Responsible for displaying all the books in the "Game of Thrones" series.
   - Handles the presentation of book-related information.
2. **display-characters:**
   - Displays a list of all characters mentioned in the "Game of Thrones" series.
   - Handles the rendering of character-related data.
3. **display-houses:**
   - Displays a list of all houses featured in the "Game of Thrones" series.
   - Manages the presentation of house-related information.
4. **display-book-details:**
   - Displays detailed information about a specific book from the series.
   - Handles the rendering of book-specific data.
5. **display-character-details:**
   - Displays detailed information about a specific character from the series.
   - Manages the rendering of character-specific data.
6. **display-house-details:**
   - Displays detailed information about a specific house from the series.
   - Manages the rendering of house-specific data.

Code Snippets

Book Interface:

Angular-pro > src > app > Interfaces > TS Book.ts > ...

```ts
export interface Book{
    url:string;
    name:string;
    isbn:string;
    authors:string[];
    numOfPages:number;
    mediaType:string;
    released:Date;
    characters:string[];
}

```

Character Interface:

Angular-pro > src > app > Interfaces > TS Character.ts > Character

```ts
export interface Character {
    id: string;
    url: string;
    name: string;
    gender: string;
    culture: string;
    born: string;
    died: string;
    titles: string[];
    aliases: string[];
    father: string;
    mother: string;
    spouse: string;
    allegiances: string[];
    books: string[];
    povBooks: string[];
    tvSeries: string[];
    playedBy: string[];
}
```

House Interface:

Angular-pro > src > app > Interfaces > TS House.ts > House

```ts
export interface House{
    id: string;
    url:string;
    name:string;
    region:string;
    coatOfArms:string;
    word:string;
    titles:string[];
    seats:string[];
    currentLord:string[];
    heir:string;
    overlord:string;
    founded: string;
    founder:string;
    diedOut:string;
    ancestralWeapons:string[];
    cadetBranches:string[];
    swornMembers:string[]
}
```

Get book by ID:

```
getBook(){
  const id = Number(this.url.split('/').pop());
  this.bookService
  .getBook(id)
  .subscribe((data) => (this.book = data))
}
onCharacterClick(url:string)
{
    this.router.navigate(['/character/details'],{state:{data:url}})
}
```

Get character by ID + click events:

```
getCharacter(){
  var id = Number(this.url.split('/').pop())
  this.characterService.getCharacter(id).subscribe((data) => (this.character = data))
}
onCharacterClick(url:string){
  this.url = url;
  this.getCharacter();
}
onHouseClick(url:string){
  this.router.navigate(['house/details'],{state:{data:url}})

}
onBookClick(url:string)
{
  this.router.navigate(['book/details'],{state:{data:url}})
}
```

Get all characters:

```
getCharacters(){
  this.characterService.getAllCharacters(this.currentPage).subscribe((data)=>(this.characters = data))
}
paginate(event: PaginatorEvent): void {
```