

Esercitazione su Angular: Gestione Liste Culturali

In questa esercitazione, ci concentreremo sulla creazione di un'applicazione web con Angular dedicata alla gestione e condivisione di liste personali relative a film, serie TV, libri o musica. L'obiettivo sarà sviluppare un'applicazione che permetta agli utenti di catalogare, valutare e condividere le proprie preferenze culturali.

Descrizione dell'esercizio

Immaginiamo di costruire un'applicativo che consenta agli utenti di creare liste personali di film, serie TV, libri o brani musicali. Ogni utente avrà la possibilità di aggiungere elementi alle liste, valutarli e scrivere recensioni.

Obiettivi principali

1. **Creazione di un'applicazione Angular:** Inizializzazione di un nuovo progetto Angular.
2. **Gestione delle Liste:** Implementazione di pagine e funzionalità per la creazione, visualizzazione e modifica delle liste culturali.
3. **Aggiunta di Elementi alle Liste:** Possibilità di aggiungere film, serie TV, libri o brani musicali alle liste degli utenti.
4. **Valutazione e Recensioni:** Consentire agli utenti di valutare gli elementi e scrivere recensioni.
5. **Sistema di Ricerca e Filtri:** Implementazione di un sistema di ricerca avanzato e filtri per semplificare la navigazione tra le liste culturali.

Possibili funzionalità

- **Creazione e Visualizzazione delle Liste:** Pagina per creare nuove liste o visualizzare quelle esistenti.
- **Aggiunta di Elementi:** Possibilità di aggiungere film, serie TV, libri o brani musicali alle liste.
- **Valutazione e Recensioni:** Sezione dedicata alle valutazioni e alle recensioni degli elementi.
- **Condivisione e Community:** Opzioni per condividere le liste con altri utenti e interagire con una community.

Questa esercitazione offrirà agli studenti la possibilità di applicare le conoscenze di Angular per creare un'applicazione dinamica e coinvolgente, consentendo loro di gestire e condividere le proprie preferenze culturali in un ambiente interattivo e divertente.

Ci sono diversi servizi RESTful e API pubbliche che forniscono accesso a database di film, serie TV, brani musicali e libri. Questi servizi consentono di recuperare informazioni dettagliate su queste opere, inclusi titoli, descrizioni, valutazioni, cast, dettagli dell'album e molto altro.

Alcuni dei servizi popolari che offrono accesso a queste informazioni sono:

1. **The Movie Database (TMDb):** TMDb offre API che forniscono informazioni sui film e le serie TV. Puoi accedere a una vasta gamma di dati, tra cui titoli, cast, trame, poster e altro ancora. Visita il [sito ufficiale di TMDb](#) per maggiori dettagli sull'utilizzo delle loro API.

2. **OMDb API:** OMDb (The Open Movie Database) offre un'API che fornisce dettagli sui film, compresi titoli, attori, valutazioni, e molto altro. Puoi trovare ulteriori informazioni sul loro [sito ufficiale](#).
3. **Last.fm API:** Last.fm offre accesso alle informazioni musicali, inclusi dettagli sugli artisti, album e brani musicali. Visita il [loro sito ufficiale](#) per informazioni sull'utilizzo delle loro API.
4. **Google Books API:** Se stai cercando informazioni su libri e pubblicazioni, Google Books API è una fonte affidabile che fornisce dettagli sui libri, recensioni, autori e altro ancora. Maggiori informazioni sono disponibili sulla [documentazione ufficiale](#).

Ricorda che molti di questi servizi potrebbero richiedere la registrazione e l'ottenimento di una chiave API per l'accesso. Assicurati di leggere attentamente la documentazione fornita da ciascun servizio per capire i termini e le condizioni d'uso delle loro API e per comprendere come effettuare le richieste corrette per ottenere i dati necessari.

In alternativa sono disponibili i seguenti endpoint

- [film JSON](#)
- [top250_imdb JSON](#)
- [cartoni JSON](#)
- [50cartoni JSON](#)
- [games JSON](#)
- [pokemon JSON](#)

scriviamo il codice angular per questa esercitazione

Per creare un'applicazione Angular che utilizzi le informazioni fornite da un servizio esterno tramite API, è necessario seguire alcuni passaggi. Nel seguente esempio, utilizzeremo l'API di The Movie Database (TMDb) per ottenere informazioni sui film. Assicurati di aver ottenuto una chiave API valida da TMDb per poter eseguire le richieste.

1. **Configurazione del Progetto Angular:** Assicurati di aver installato Angular CLI e crea un nuovo progetto:

```
ng new nome-progetto
cd nome-progetto
```

2. **Installa HttpClient Module:** HttpClient è un modulo di Angular necessario per effettuare richieste HTTP. Assicurati di averlo installato nel tuo progetto.

```
ng generate service movie
```

Questo comando genererà un nuovo servizio chiamato "movie" che useremo per effettuare le richieste API.

- 3. Modifica del servizio "movie":** Nel servizio "movie", inserisci il codice per effettuare la chiamata all'API di TMDb per ottenere informazioni sui film. Assicurati di sostituire 'YOUR_API_KEY' con la tua chiave API di TMDb.

```
// movie.service.ts
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Injectable({
  providedIn: 'root'
})
export class MovieService {
  private apiKey = 'YOUR_API_KEY';
  private apiUrl = 'https://api.themoviedb.org/3';

  constructor(private http: HttpClient) { }

  getMovies() {
    return this.http.get(`${this.apiUrl}/movie/popular?api_key=${this.apiKey}`);
  }
}
```

- 4. Componente per visualizzare i film:** Crea un componente per visualizzare i film ottenuti dall'API.

```
ng generate component movie-list
```

- 5. Modifica del componente "movie-list":** All'interno del componente "movie-list", richiama il servizio "movie" per ottenere i dati e visualizzali.

```
// movie-list.component.ts
import { Component, OnInit } from '@angular/core';
import { MovieService } from '../movie.service';

@Component({
  selector: 'app-movie-list',
  templateUrl: './movie-list.component.html',
  styleUrls: ['./movie-list.component.css']
})
export class MovieListComponent implements OnInit {
  movies: any;

  constructor(private movieService: MovieService) { }

  ngOnInit(): void {
    this.movieService.getMovies().subscribe((data: any) => {
      this.movies = data.results;
    });
  }
}
```

```
}
}
```

- 6. Modifica del template HTML del componente "movie-list":** Modifica il file HTML per mostrare i film ottenuti dall'API.

```
<!-- movie-list.component.html -->
<div *ngIf="movies">
  <h2>Elenco dei Film Popolari</h2>
  <ul>
    <li *ngFor="let movie of movies">
      {{ movie.title }}
    </li>
  </ul>
</div>
```

Assicurati di effettuare il binding dei dati ottenuti dall'API all'interno del tuo componente. Questo esempio mostra come visualizzare i titoli dei film popolari ottenuti dall'API di TMDb. Puoi estendere questo esempio aggiungendo più funzionalità, come la visualizzazione di immagini, descrizioni, valutazioni, ecc., a seconda dei dati forniti dall'API.

Gestisci le valutazioni

Per attribuire una valutazione ai film e memorizzarla all'interno dell'applicazione, puoi utilizzare vari metodi. Un'opzione è quella di creare un modello per le valutazioni dei film e utilizzare un servizio per gestire e memorizzare queste valutazioni.

Ecco un'idea di come potresti gestire la valutazione dei film all'interno del tuo progetto Angular:

- 1. Creazione di un Modello per le Valutazioni:** Crea un modello TypeScript per le valutazioni all'interno del progetto Angular. Questo modello rappresenterà la struttura di una valutazione.

```
// rating.model.ts
export interface Rating {
  movieId: number;
  value: number;
}
```

- 2. Modifica del Servizio per Gestire le Valutazioni:** Aggiungi al servizio `MovieService` metodi per gestire le valutazioni. Ad esempio, un metodo per aggiungere una valutazione al film.

```
// movie.service.ts
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
```

```
import { Observable } from 'rxjs';
import { Rating } from './rating.model';

@Injectable({
  providedIn: 'root'
})
export class MovieService {
  private apiKey = 'YOUR_API_KEY';
  private apiUrl = 'https://api.themoviedb.org/3';
  private ratings: Rating[] = []; // Array per memorizzare le valutazioni

  constructor(private http: HttpClient) {}

  getMovies(): Observable<any> {
    return this.http.get(`${this.apiUrl}/movie/popular?api_key=${this.apiKey}`);
  }

  // Metodo per aggiungere una valutazione
  addRating(movieId: number, value: number): void {
    this.ratings.push({ movieId, value });
  }

  // Metodo per ottenere le valutazioni
  getRatings(): Rating[] {
    return this.ratings;
  }
}
```

- 3. Aggiunta di un Componente per Valutare i Film:** Crea un componente che consenta agli utenti di assegnare una valutazione a un film. Ad esempio, un componente con una lista di film e la possibilità di assegnare una valutazione.

Questo componente potrebbe includere un form o un sistema di valutazione (es. stelle) per consentire agli utenti di dare una valutazione a ciascun film.

- 4. Salvataggio delle Valutazioni:** Quando l'utente assegna una valutazione, utilizza il metodo del servizio `MovieService` per salvare questa valutazione nel servizio stesso. In questo esempio, le valutazioni vengono memorizzate in un array all'interno del servizio stesso, ma in un'applicazione reale, potresti voler salvare queste informazioni in un database o in un backend.

- 5. Visualizzazione delle Valutazioni:** Per visualizzare le valutazioni assegnate, potresti utilizzare il servizio `MovieService` per ottenere le valutazioni salvate e mostrarle insieme ai dettagli del film.

Ricorda che questo esempio è semplificato e serve come base per gestire le valutazioni all'interno dell'applicazione. Potresti voler estendere queste funzionalità, ad esempio aggiungendo la gestione degli utenti e delle loro valutazioni, la persistenza dei dati, o altre funzionalità in base alle esigenze dell'applicazione.