

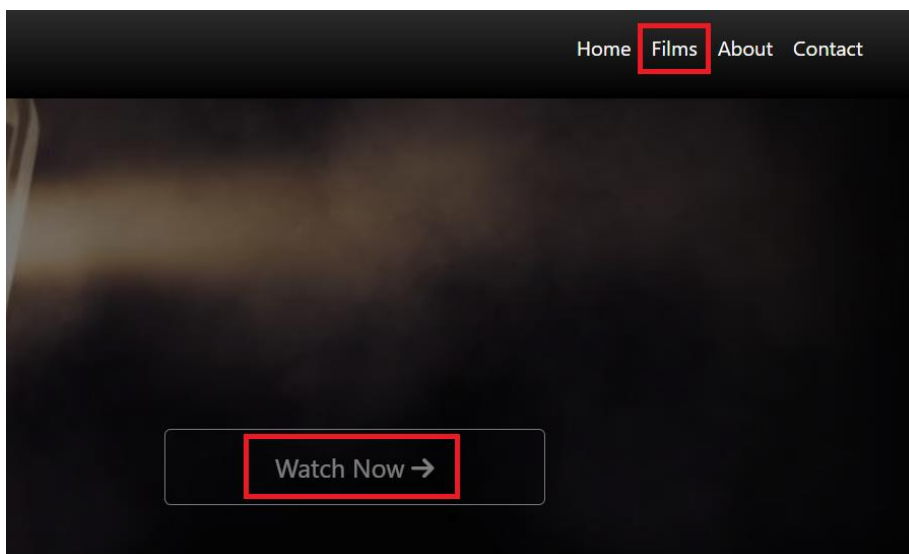
# Création d'une application Angular – TekFlix

## TP 1

### Introduction du TP :

### Configuration du routage pour le MoviesListComponent :

Nous souhaitons ajouter un lien "Films" et un bouton "Watch Now" qui redirigent vers la page contenant la liste des films. Pour cela, nous allons utiliser la directive **routerLink** et le service **Router**.



Dans le fichier `app-routing.module.ts`, ajoutez la route pour `MoviesListComponent`.

```
const routes: Routes = [  
  {path: '|', component: HomeComponent},  
  {path: 'movies-list', component: MoviesListComponent},  
];
```

### On utilisons : RouterLink :

Dans le fichier `header.component.html`, qui contient la barre de navigation, ajoutez la directive `RouterLink` :

```
<li class="nav-item">  
  <a class="nav-link text-white" routerLink="/movies-list">Films</a>  
</li>
```

**RouterLink :** est une directive Angular utilisée pour lier une route à un élément HTML, comme un lien. Lorsqu'un utilisateur clique sur cet élément, il est redirigé vers la route spécifiée.

- L'avantage de RouterLink est qu'il permet de charger un composant spécifique associé à la route, sans recharger toute l'application. Cela offre une expérience plus fluide et rapide, car seule la partie pertinente de l'application (le composant associé à la route) est mise à jour, plutôt que de recharger la page entière.
- Cela rend l'application plus réactive et améliore les performances, particulièrement dans les applications de type Single Page Application (SPA).

## On utilisons le service Router :

Injectez le service Router de **@angular/router** dans le constructeur afin de pouvoir l'utiliser.

Créez la méthode **goToMoviesList** qui permet de nous rediriger vers la page **/movies-list**

```
@Component({
  selector: 'app-home',
  standalone: false,
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent {

  films: Film[] = FILMS;
  constructor(private router: Router) { }

  Ctrl+L to chat, Ctrl+K to generate
  goToMoviesList() {
    this.router.navigate(['/movies-list']);
  }
}
```

Dans **home.component.html**, mettez à jour le bouton "Watch Now" pour ajouter l'événement de redirection.

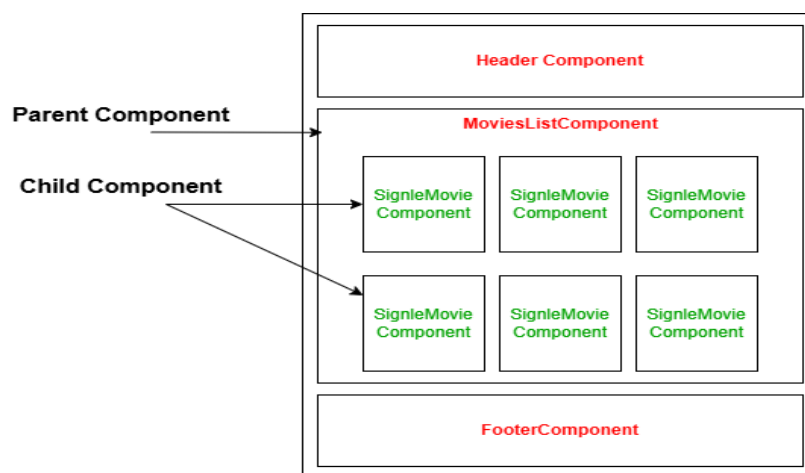
```
<!-- Button positioned at the bottom right -->
<button class="btn btn-outline-light btn-lg" (click)="goToMoviesList()"
  style="height: 60px; width: 300px; position: absolute; bottom: 200px; right: 370px; border-radius: 5px;">
  Watch Now <i class="fas fa-arrow-right"></i>
```

**(click)** : Il s'agit d'un événement Angular. Cela signifie que lorsque l'utilisateur clique sur le bouton, l'action spécifiée à l'intérieur des guillemets sera exécutée.

## Création du MoviesListComponent :

Mettre le contenu des fichiers **movies-list.html** et **movies-list.css** (sur drive) dans **movies-list.component.html** et **movies-list.component.css** respectivement.

Afin d'appliquer la relation **parent-enfant** dans notre application, nous allons créer et utiliser un composant **SingleMovie**. Ce composant servira à afficher les détails d'un film individuel et sera intégré dans le composant parent qui gère la liste des films. Cela permettra une meilleure organisation du code, une réutilisation efficace du composant et une communication fluide entre le parent et l'enfant à l'aide de **@Input()** et **@Output()** si nécessaire.



Pour créer le composant **SingleMovie**, utilisez la commande Angular CLI suivante dans votre terminal :

```
C:\Users\TEK-UP\Desktop\Tek-up\Angular\TekFlix>ng g single-movie
```

Dans MoviesListComponent (**Parent Component**) :

```
3
4 @Component({
5   selector: 'app-movies-list',
6   standalone: false,
7   templateUrl: './movies-list.component.html',
8   styleUrls: ['./movies-list.component.css']
9 })
10 export class MoviesListComponent {
11
12   films : Film[] = FILMS;
13   // déclarer un attribut films // initialiser cette liste avec des films que nous
14   // comme une liste de Film // venons de créer dans Film.ts.
15 }
```

Dans `movies-list.component.html`, insérez le composant `SingleMovieComponent` à l'aide de son sélecteur et affichez la liste des films en utilisant la directive `*ngFor`.

```
<div id="movieCarousel" class="carousel slide" data-bs-ride="carousel">
  <div class="d-flex justify-content-center mt-4">
  </div>
  <div class="carousel-inner">
    <div class="carousel-item active">
      <div class="row mx-auto" style="max-width: 1200px;">
        <div class="col-sm-6 col-md-3" *ngFor="let film of films">
          <app-single-movie [film]="film"></app-single-movie>
        </div>
      </div>
    </div>
  </div>
</div>
```

`*ngFor="let film of films"`

- Cette directive **boucle** sur la liste `films` et affiche un film à chaque tour.
- `film` représente un **élément unique** de la liste `films`.

`<app-single-movie [film]="film"></app-single-movie>`

- Affiche le composant `SingleMovieComponent` pour chaque film.
- `[film]="film"` permet d'envoyer les **données du film** au composant enfant `SingleMovieComponent`.

Dans `SingleMovieComponent` (**Child Component**) :

```
@Component({
  selector: 'app-single-movie',
  standalone: false,
  templateUrl: './single-movie.component.html',
  styleUrls: ['./single-movie.component.css']
})
export class SingleMovieComponent {
  @Input() film!: Film;
}
```

## Explication :

**@Input() movie!: Film;** permet à un composant enfant (SingleMovieComponent) de recevoir un film depuis son composant parent (MoviesListComponent).

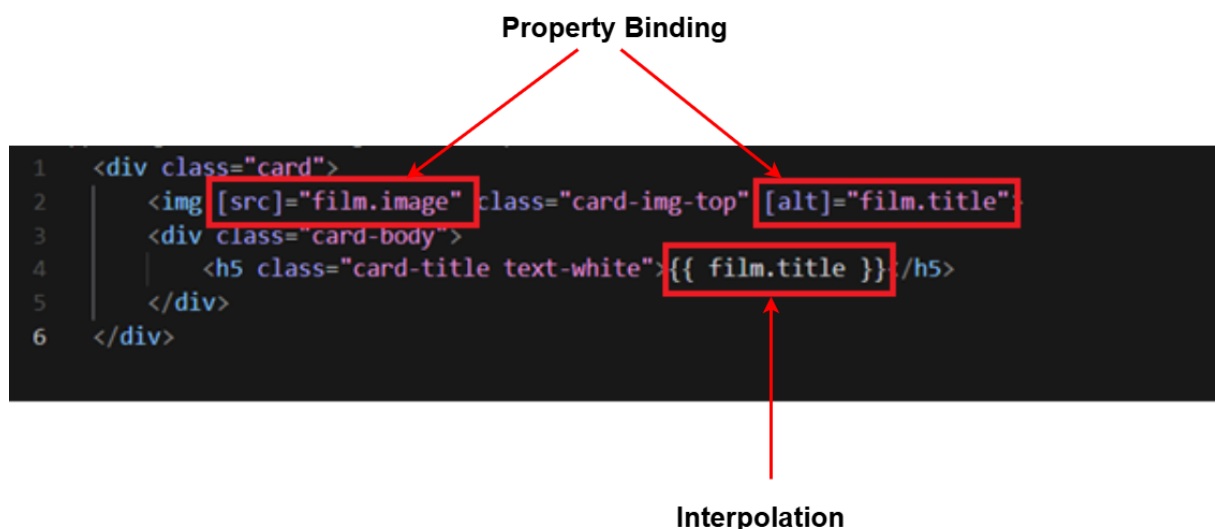
**film!** : movie est une variable qui va stocker un film.

**Le ! (non-null assertion) :** signifie que cette variable **ne peut pas être null ou undefined**.

**@Input :** est un décorateur en Angular qui permet à un composant enfant de recevoir des données d'un composant parent

Dans SingleMovieComponent (**Child Component**) :

Ajoutez le code suivant dans votre composant pour afficher les informations d'un film sous forme de carte :



## Explication :

`<img [src]="film.image" class="card-img-top" [alt]="film.title">`

**film.image** est une liaison de propriété qui définit l'URL de l'image du film.

**[alt]="film.title"** améliore l'accessibilité en affichant le titre du film si l'image ne charge pas.

Dans **single-movie.component.css** ajoutez le style qui permet de changer le couleur du fond en noir :

```
* {  
  background-color: black;  
}
```

### Résultat à obtenir

