

## Objectifs

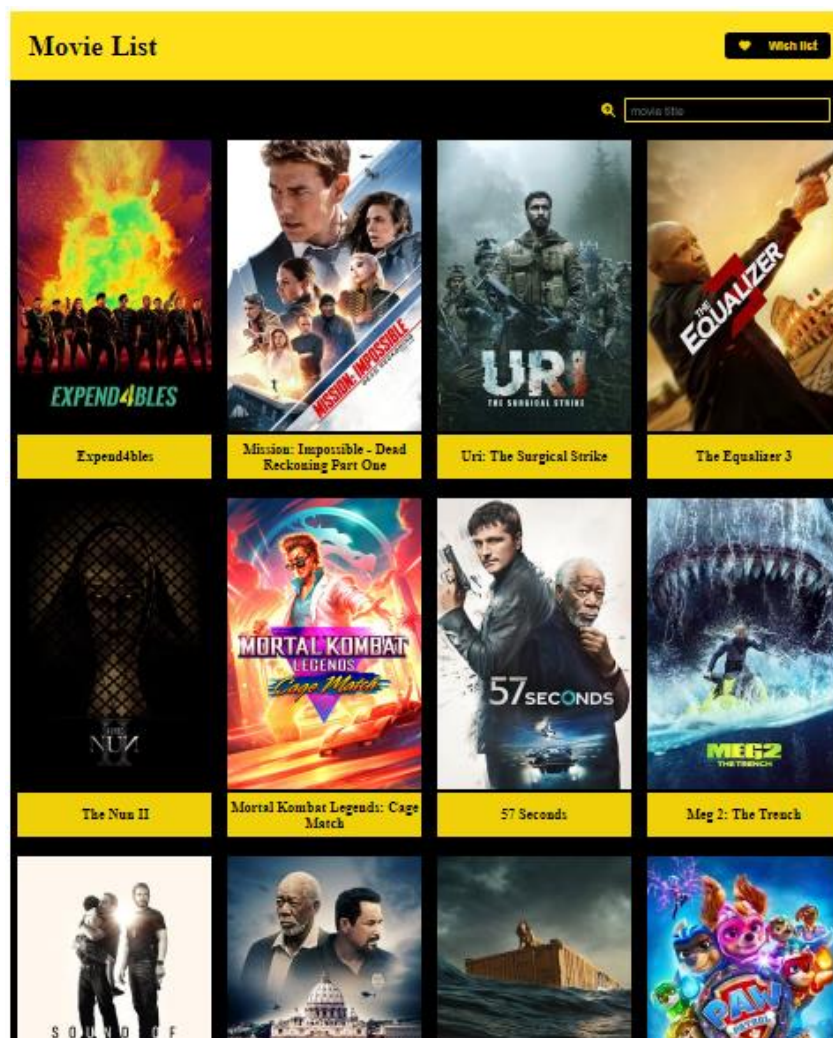
- Les API
- Les requêtes http
- Le routage angular
- Les observables

## Outils de travail

- Inscrivez-vous au site [tmdb.org](https://tmdb.org) et demandez une APIKey pour pouvoir exécuter les différentes requête http du site.
- Accédez au site [www.postman.com](https://www.postman.com) pour tester les APIs

### A- Préparation de la structure de l'application

- 1- Créer une nouvelle application Angular nommée movie-app qui permet d'afficher et de gérer une liste de films. La figure suivante montre une vue au démarrage de l'application.



-Figure 1. Ecran de démarrage de movie-app-

- 2- Installer les packages angular fontawesome pour insérer les icônes dans l'application en exécutant les commande npm suivantes :
  - npm install @fortawesome/fontawesome-svg-core
  - npm install @fortawesome/free-solid-svg-icons
  - npm install @fortawesome/angular-fontawesome
- 3- Pour apprendre comment insérer une icône dans une application Angular, voir la documentation sous :  
<https://www.npmjs.com/package/@fortawesome/angular-fontawesome>.
- 4- Créez un fichier app/src/environments/environment.ts, dans lequel sera conservée votre API key.
- 5- Créer les composants head-bar, search-bar, movie-list, movie-detail, movie-search et wish-list.
- 6- Créer un service movie dans lequel seront implémentées les méthodes getMovieList, getMovieById et getMovieByTitle.
- 7- Créer une interface movie. Définir l'interface Movie, selon la structure d'un film dans TMDb.org. Vous pouvez utiliser l'adresse <https://transform.tools/json-to-typescript>.

#### **B- Le service movie.service**

Dans le service movie-service, injectez le service HttpClient et créez les méthodes :

- getMovieList() : retourne la liste des films de TMDb.org à l'aide de l'API [https://api.themoviedb.org/3/movie/now\\_playing?api\\_key=\\${this.apiKey}](https://api.themoviedb.org/3/movie/now_playing?api_key=${this.apiKey})
- getMovieById(id) : cherche un film dont l'identifiant est donné en paramètre à l'aide de l'API [https://api.themoviedb.org/3/movie/\\${id}?api\\_key=\\${this.apiKey}](https://api.themoviedb.org/3/movie/${id}?api_key=${this.apiKey})
- getMovieByTitle(title) : cherche un film par son titre à l'aide de l'API [https://api.themoviedb.org/3/search/movie?query=\\${title}&api\\_key=\\${this.apiKey}](https://api.themoviedb.org/3/search/movie?query=${title}&api_key=${this.apiKey})

#### **N.B**

Pour utiliser le service HttpClient, dans la version Angular 18, il faut :

- L'injecter dans le service movie.service à l'aide de `private httpClient=inject(HttpClient)`
- Dans le fichier app.config.ts, ajouter `provideHttpClient(withFetch())` dans la liste de la propriété providers

#### **C- Le routage**

Le routage permet de naviguer entre les composants en définissant un chemin pour chaque composant. Pour créer le routage avec Angular 18 :

- Dans le fichier app.routes.ts, définir les chemin de l'application, à l'aide de la variable routes :  
Routes =[ {path : 'chemin', component : nom\_classe\_composant},...]
- Dans le fichier app.config.ts, importer le service importer provideRouter(routes), tel que routes est le tableau qui contient toutes les routes de l'application et qui est défini dans le fichier app.routes.ts

#### D- Le composant head-bar

- 1- Le template du composant affiche le titre h1 "Movie List" et un bouton "Wish list" avec l'icône "heart"
- 2- Consulter <https://fontawesome.com/icons> et chercher « heart » pour voir comment insérer l'icône dans l'application.
- 3- Ajouter les styles dans le fichier de style du composant.
- 4- Insérer le composant dans le composant principal.



-Figure 2. Head-bar-

#### E- Le composant search-bar

Ce composant permet au visiteur de trouver un film en tapant le titre dans la zone de recherche.

- 1- Utiliser les événements input dans la zone de saisie du titre pour faire appel à une méthode searchMovieByTitle(titre) qui retrouve un film par son titre.
- 2- Ajouter les styles dans le fichier css du composant pour mettre en forme le template.



-Figure 3. Search-bar-

#### F- Le composant movie-list

Ce composant affiche la liste des films de TMDB.org.

Implémentez une méthode getMovieList() dans le service movie-service.ts

Ce composant héberge le composant search-bar. Il affiche la liste des films de TMDB.org ou la liste des films correspondant au titre saisi par l'utilisateur. La méthode.

#### NB.

- Le composant fils search-bar doit passer le titre choisi au composant père movie-list.
- Utiliser la directive ngIf ; else et ng-template dans le template du composant movie-list.

**G- Le composant movie-detail**

Affiche les détails sur un film choisi par l'utilisateur dans la barre de recherche. La méthode `getMovieById(id)` sera appelée pour retrouver le film choisi dans la liste des films.



-Figure 4. Movie-detail-