

Dans la première partie du fil rouge, nous avons reconçu notre application en utilisant bootstrap .

Dans la seconde partie, nous avons « refactoré » en utilisant des composants React et react-bootstrap.

Dans la précédente partie, nous avons intégré les données en provenance de l'api externe et mis en œuvre les hook useState et useEffect

Dans cette partie, vous allez finaliser la consultation du catalogue avec la gestion des liens vers les différentes pages, en utilisant les composants et hook de react-router

I. Configuration des routes

Modifiez le fichier index.js pour qu'il réponde aux routes telles que définies :

Route			Composant
/			App
	home		Home
	genre		Genre
		<i>index</i>	GenreList
		:id	GenreDetails
	actor		Actor
		<i>index</i>	ActorList
		actor/:id	ActorDetails
	movie		Movie
		<i>index</i>	MovieList
		:id	MovieDetails

NB : Le tableau présente l'arborescence (imbrication) des routes

- Les composants en vert seront à créer 😊. Dans un premier temps laissez-juste le squelette, et testez que vos routes s'affichent convenablement, ne pas oublier la gestion des erreurs 😊

II. Paramétrage de la navigation

A. Header

- Modifiez les éléments du menu pour appeler les routes

Lien	Route
Logo	/
Accueil	/
Tous	/movie
Genres/*	/genre/:id
Acteurs/*	/actor/:id

Afin d'améliorer l'affichage du fil d'ariane, les éléments du menu (genres et acteurs) feront le lien sur l'attribut **name** et non pas l'id ,

- Modifiez le composant DropdownItem pour qu'il reçoive via les props , le type d'élément (genre ou actor) qui sera affiché
- Ajoutez le lien vers la route

```
<a href="/genre/Action">Action</a> ,  
<a href="/actor/3D" >3D</a> ,
```

B. HomeCarousel

Modifiez le carrousel en ajoutant un lien sur le titre. Ce lien pointera sur la route du film sélectionné

```
<a href="/movie/Rebel Moon - Partie 1 : Enfant du feu">Rebel Moon - Partie 1 : Enfant du  
feu (2023)</a>
```

C. MovieThumbnail

Modifier les liens pour pointer vers la route du film en utilisant son attribut title

```
<a href="/movie/Expend4bles">Expend4bles</a>
```

III. Affichage des pages via les routes

A. /

Cette route permet d'afficher la page racine du site via le composant <App>

1. App

Ce composant est découpé en sous-composant :

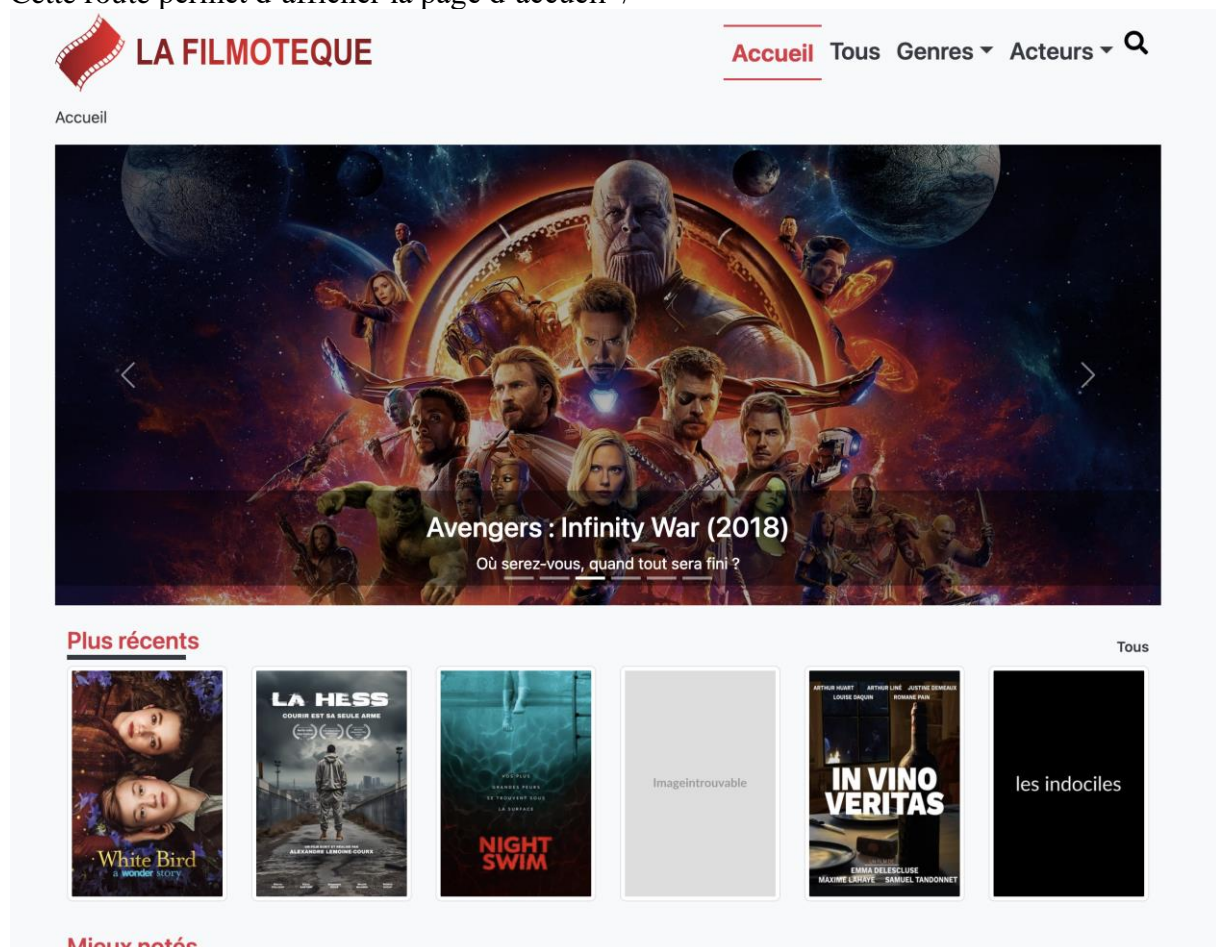
- <Header> pour l'affichage de l'entête
- AppBreadcrumb pour le fil d'ariane. (fourni)
- Une zone dynamique, pour l'affichage des autres des routes enfant
- <Footer> pour le pied de page

Par défaut la zone dynamique affichera le composant <Home> répondant à la route /home

L'entête et le pied de page seront affichés sur toutes les pages

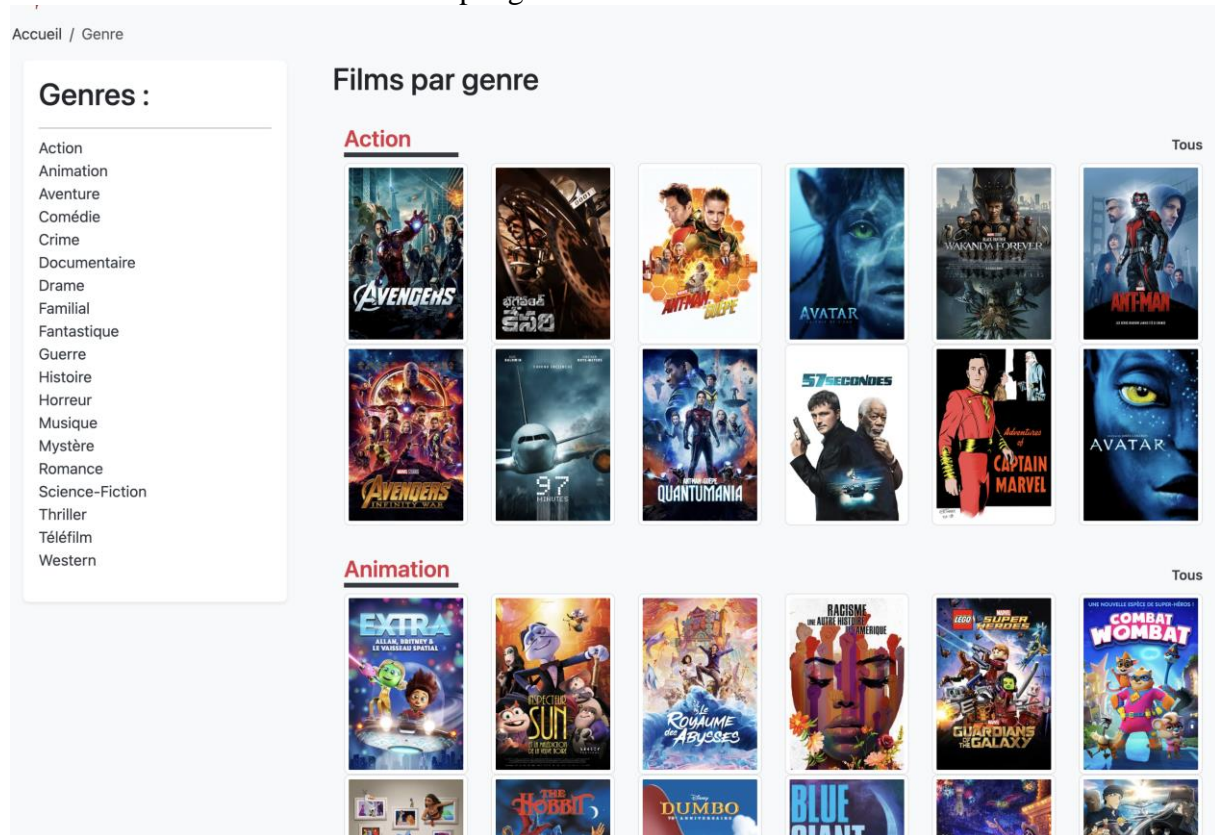
B. /home

Cette route permet d'afficher la page d'accueil /



C. /genre

Cette route sert à afficher les films par genre



L'affichage est découpé en 2 zones :

- A gauche, la zone fixe de choix du genre via un composant à créer <GenreNav>
- Au centre la zone dynamique issue de la sélection d'un genre

Par défaut la zone dynamique affiche le composant <GenreList>

Quand un genre est sélectionné, la zone dynamique affiche le composant <GenreDetails>

1. GenreNav

Ce composant permet :

- De lister les genres pour faciliter la recherche de film par genre
- La sélection du genre provoque l'affichage de la route genre/xxx

Les genres sont récupérés via l'api

<http://filmotheque.e-mingo.net/api/catalog/genre>

2. GenreList

Ce composant permet l'affichage d'une sélection de films pour chaque genres

Les données sont récupérées via l'api

<http://filmotheque.e-mingo.net/api/catalog/movie/?genre>

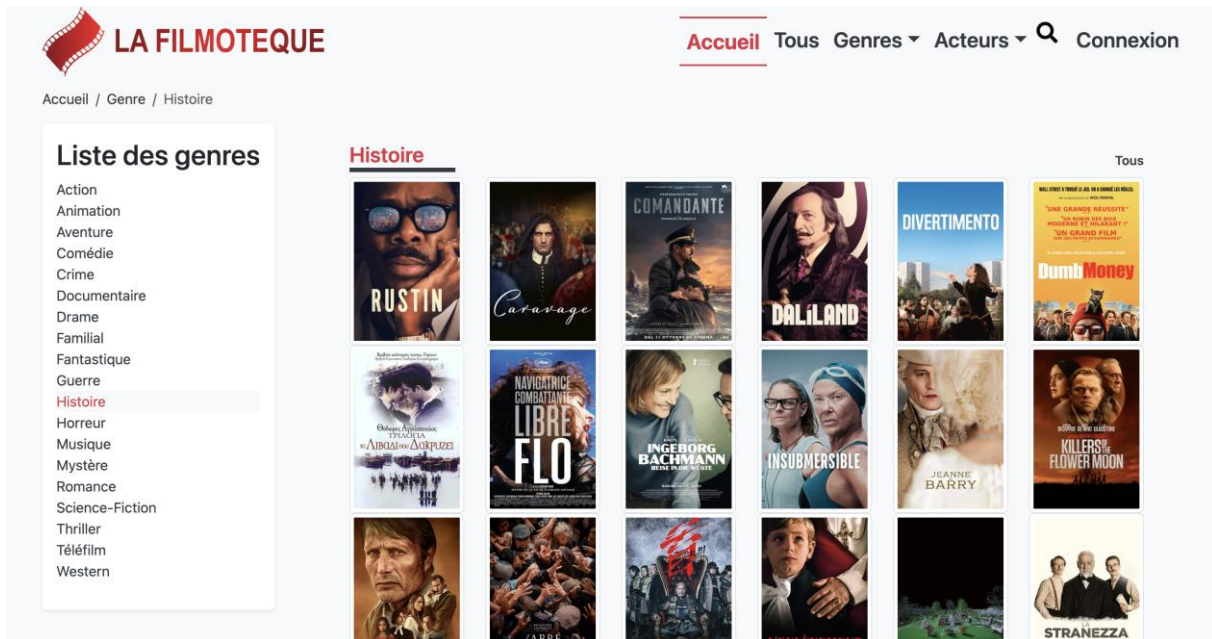
Pour chaque genre les données sont affichées via le composant ContentBox

D. /genre/xxx

Cette route permet d'affiche les films du genre spécifié via le composant <GenreDetails>

1. GenreDetails

Ce composant sert à l'affichage de l'ensemble des films d'un genre cinématographique



Les données sont récupérées via l'api

<http://filmotheque.e-mingo.net/api/catalog/genre/<Nom ou id du genre>>