

Date: 07/04/2025 - 11/04/2025

SPRINT 4 BRIEF 1

Brief 5 : Application de consultation de films avec consommation d'une API externe

Contexte professionnel

404.js souhaite développer une **application de films** permettant aux utilisateurs de **consulter des informations à jour** sur les films, séries et acteurs via une **API externe**. Ce projet a pour objectif de maîtriser la **consommation d'API publiques**, la **gestion des requêtes asynchrones**, l'**affichage dynamique des données** et la **navigation dans une application React**.

L'application doit offrir une **expérience utilisateur fluide et responsive**, avec une interface bien structurée et intuitive.

Fonctionnalités attendues

- **Consommation d'une API externe** (ex : TMDb, OMDb).
- **Recherche de films ou séries** par titre.
- **Affichage des résultats** avec affiches, titres, notes et descriptions.
- **Page de détails** pour chaque film.
- **Filtrage** par catégories (populaire, tendances, etc.).

- **Navigation fluide** avec **React Router**.
 - **Mode sombre / mode clair** (Dark/Light Mode).
 - **Loader animé** pendant le chargement des données.
-

Architecture technique

- **Frontend uniquement** (React JS avec Tailwind CSS).
 - App basé sur une structure **MVC côté client** (organisation des composants, routes, services).
 - **Utilisation de fetch ou axios** pour les appels API.
 - Stockage des favoris ou de l'historique en **localStorage** (optionnel).
-

Exploitation des fonctions d'ordre supérieur (HOFs)

L'application devra intégrer l'utilisation de **fonctions d'ordre supérieur** telles que :

- `.map()` pour afficher dynamiquement les listes de films.
 - `.filter()` pour filtrer selon des critères (score, date, etc.).
 - `.sort()` pour trier les films par popularité ou date.
 -
-

Modalités pédagogiques

- **Travail** : individuel.
 - **Durée** : 5 jours (17/03/2025 - 21/03/2025)
 - **Évaluation** : Présentation du projet, démonstration de l'authentification, revue de code et questions techniques.
-


Livrables

1. **Code source** (GitHub).
 2. **Maquette de l'application** (Figma ou Adobe XD).
 3. **Lien de planification** (Trello, Jira, etc.)
 4. **Présentation du projet** (Canva, PowerPoint).
-


Critères de performance

- ✓ **Consommation correcte d'une API externe** (fetch ou axios).
 - ✓ **Recherche fonctionnelle avec affichage dynamique.**
 - ✓ **Navigation fluide avec React Router.**
 - ✓ **Interface responsive et bien structurée** (avec Tailwind CSS).
 - ✓ **Code propre et bien organisé** avec séparation des composants et services.
-

Ressources utiles

 **TMDb API** → <https://developer.themoviedb.org/>

 **OMDb API** → <http://www.omdbapi.com/>

 **React JS** → <https://react.dev/>

 **React Router** → <https://reactrouter.com/>

 **Tailwind CSS** → <https://tailwindcss.com/>

 **Axios** → <https://axios-http.com/>

les compétences techniques visées : C1N2, C2N2, C3N2, C4N2.

**les compétences transversales visées : CT1N2, CT4N2,
CT6N2,CT8N2.**