

Pour ce TP, nous allons créer une application web qui affiche une liste de films Star Wars récupérée depuis l'API SWAPI. L'utilisateur pourra cliquer sur un film de la liste pour afficher ses détails. Ce TP combine l'utilisation de l'API Fetch pour récupérer des données, la manipulation du DOM pour afficher ces données, et la gestion des événements pour interagir avec l'utilisateur.

Étape 1 : Structure HTML

Votre fichier HTML (`index.html`) servira de squelette pour votre application. Il contiendra une liste pour les titres des films et un conteneur pour les détails du film sélectionné.

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>Liste de Films Star Wars</title>
  </head>
  <body>
    <h1>Films Star Wars</h1>
    <ul id="films-list"></ul>

    <div id="film-details" style="margin-top: 20px;">
      <!-- Les détails du film seront injectés ici -->
    </div>

    <script src="script.js"></script>
  </body>
</html>
```

Étape 2 : Script JavaScript

Le fichier JavaScript (`script.js`) contiendra la logique pour interroger l'API SWAPI, afficher la liste des films, et gérer le clic sur un film pour afficher ses détails.

```
document.addEventListener("DOMContentLoaded", () => {
  const filmsListElement = document.getElementById("films-list");
  const filmDetailsElement = document.getElementById("film-details");

  // Fonction pour afficher la liste des films
  function afficherListeFilms() {
    fetch("https://swapi.dev/api/films/")
      .then((response) => response.json())
      .then((data) => {
        data.results.forEach((film) => {
          const li = document.createElement("li");
          li.textContent = film.title;
          li.style.cursor = "pointer";
          li.addEventListener("click", () =>
            afficherDetailsFilm(film.url));
          filmsListElement.appendChild(li);
        });
      });
  }

  afficherListeFilms();
});
```

```

    });
  })
  .catch((error) =>
    console.error("Erreur lors de la récupération des films :", error)
  );
}

// Fonction pour afficher les détails d'un film sélectionné
function afficherDetailsFilm(url) {
  fetch(url)
    .then((response) => response.json())
    .then((film) => {
      filmDetailsElement.innerHTML = `
        <h2>${film.title}</h2>
        <p><strong>Date de sortie :</strong>
${film.release_date}</p>
        <p><strong>Réalisateur :</strong> ${film.director}</p>
        <p><strong>Producteurs :</strong> ${film.producer}</p>
        <p><strong>Résumé :</strong> ${film.opening_crawl}</p>
      `;
    })
    .catch((error) =>
      console.error(
        "Erreur lors de la récupération des détails du film :",
        error
      )
    );
}

afficherListeFilms();
});

```

Explications

1. **Affichage de la liste des films** : À la fin du chargement du DOM, la fonction `afficherListeFilms` est appelée. Elle récupère la liste des films depuis l'API SWAPI et crée un élément `li` pour chaque film, qu'elle ajoute ensuite à la liste `ul` dans le HTML. Chaque élément `li` a un gestionnaire d'événement `click` qui appelle `afficherDetailsFilm` avec l'URL du film.
2. **Affichage des détails d'un film** : Lorsqu'un utilisateur clique sur un titre de film, la fonction `afficherDetailsFilm` est appelée avec l'URL du film sélectionné. Cette fonction récupère les détails du film depuis l'API SWAPI et met à jour le contenu de `film-details` dans le DOM avec ces informations.

Étape 3 : Tester votre application

- Assurez-vous que les fichiers `index.html` et `script.js` sont bien sauvegardés et se trouvent dans le même répertoire.
- Ouvrez `index.html` dans un navigateur pour voir la liste des films s'afficher.
- Cliquez sur un film pour voir ses détails apparaître sous la liste.