Anthony RIMET: IUT Nancy-Charlemagne 2024

TD2 - Vue - Les bases

Introduction

Dans ce TD, vous allez réaliser une mini-application permettant d'interroger l'API Web OMDB. Vous allez utiliser le framework VueJS 3 pour réaliser ce TD.

API Key

Vous devrez utiliser l'API KEY que vous avez générée pour le TD1.

Exercice 1 : Création du projet

1. Créez un projet VueJS 3 en utilisant le CLI de VueJS. (Utilisez les options de base, vous pouvez ajouter ESLINT et Prettier si vous le souhaitez.)
2. Souvent, avec les installations de projet, vous vous retrouvez avec du code inutile. Supprimez les dossiers `components` et `assets` dans un premier temps. Vous pouvez aussi supprimer le contenu du fichier `App.vue`.
3. Dans le fichier `App.vue`, ajoutez un champ de saisie de type texte. Lorsque l'utilisateur tape du texte dans l'input, vous devrez afficher ce texte dans la console.

Exercice 2 : OMDB API

1. Modifiez votre code pour qu'au clic sur un bouton, vous interrogiez l'API OMDB pour rechercher des films.
2. Vous devrez créer une méthode qui fera un appel à l'API OMDB. Vous pouvez utiliser `fetch` pour faire cet appel. Les résultats de la recherche devront être stockés dans une variable `currentResults`.
3. Affichez les résultats sous forme de liste. Vous devrez afficher le titre du film, l'année de sortie et le type de film. Il est important de gérer les différentes étapes de la recherche pour donner du feedback à l'utilisateur (chargement, erreur, pas de résultats). Si un film n'a pas d'image, vous pouvez afficher une image par défaut.

Exercice 3 : Gestion de l'historique

1. L'utilisateur veut pouvoir consulter l'historique de ses recherches. Vous devrez stocker les recherches dans une variable `searches`. Cette variable devra être un tableau d'objets. Chaque objet devra contenir le texte de la recherche et les résultats de la recherche.

2. Affichez l'historique des recherches sous forme de liste. Lorsque l'utilisateur clique sur une recherche, vous devrez afficher les résultats de cette recherche.
3. Ajoutez un bouton pour effacer l'historique des recherches. Affichez la recherche en cours dans une couleur différente.
4. Ajoutez un bouton "removeAll" qui permet de remettre toutes les données à zéro.

Exercice 4 : Détails

1. Lorsque l'utilisateur clique sur un film, vous devrez afficher les détails de ce film. Vous devrez faire un appel à l'API OMDB pour récupérer les détails du film.
2. Il faut indiquer que le film est cliquable. Lors du clic, une méthode doit être appelée pour récupérer les détails du film. Placez le résultat dans une variable `currentMovie`.
3. Affichez les détails du film. Vous pouvez afficher le titre, l'année de sortie, le type, le réalisateur, les acteurs, la note IMDB et le synopsis. Si un détail n'est pas disponible, affichez un message par défaut. Si `currentMovie` n'est pas défini, rien ne doit être affiché.

Exercice 5 : Pimp

1. Dans les détails d'un film, la durée est exprimée en minutes. Vous devrez convertir cette durée en heures et minutes. Par exemple, si la durée est de 120 minutes, vous devrez afficher 2h00. Pour cela, utilisez une propriété calculée (`computed property`).
2. Sur le même principe, vous voulez afficher le nombre de résultats pour la recherche en cours. Par exemple, si vous avez 5 résultats, affichez "5 résultats". Si vous avez 1 résultat, affichez "1 résultat". Utilisez également une propriété calculée pour cela.

Exercice 6 : Conserver les données

1. Si vous rafraîchissez la page, vous perdez toutes les données. Vous devrez stocker les données dans le `localStorage` pour les conserver entre les rafraîchissements de page. Un indice pour vous aider : `localStorage`. Ensuite, regardez du côté du hook `mounted`.
2. Effacez les données du `localStorage` lorsque l'utilisateur clique sur le bouton "removeAll".

Exercice 7 : Bonus

1. Ajoutez un bouton pour ajouter un film à une liste de favoris. Vous pouvez stocker les favoris dans une variable `favorites`. Cette variable devra être un tableau d'objets. Chaque objet devra contenir les détails du film. L'utilisateur doit pouvoir retirer un film de la liste des favoris. Et ces favoris doivent être conservés en mémoire même si la page est rafraîchie.
2. Rendez votre application agréable visuellement. Vous pouvez utiliser des librairies de composants comme Vuetify ou Tailwind pour vous aider.