

## AppSuiviFilm - Application de Suivi de Films

### Description

AppSuiviFilm est une application mobile développée avec **NativeScript Vue** qui permet aux utilisateurs de suivre les films qu'ils ont regardés, attribuer des notes et rédiger des critiques. Elle inclut un système d'authentification, une liste de films, ainsi qu'un module de notation et de commentaires.

### Fonctionnalités

**Authentification des utilisateurs** (inscription et connexion)

**Consultation de la liste des films**

**Ajout d'une note et d'un commentaire sur un film**

**Indiquer si un film a été vu ou non**

**Navigation entre les différentes pages**

### Technologies Utilisées

- **NativeScript Vue** (framework mobile basé sur Vue.js)
- **Axios** (pour les requêtes API)
- **Vuex** (gestion de l'état - si applicable)
- **ApplicationSettings** (stockage des tokens et préférences utilisateurs)
- **CSS/SCSS** (pour le design)

### Structure du Projet

AppSuiviFilm/

|— app/

| |— components/

| | |— Home.vue      # Page d'accueil

| | |— Login.vue      # Connexion utilisateur

| | |— Register.vue    # Inscription utilisateur

```
| | |— MovieList.vue    # Liste des films
| | |— MovieReview.vue  # Page d'évaluation d'un film
| |
| |— api.js             # Gestion des requêtes API
| |— app.js             # Point d'entrée de l'application
| |— app.scss           # Styles généraux
| |— fonts/             # Polices utilisées dans l'application
|
|— node_modules/        # Dépendances installées
|— package.json         # Dépendances et scripts npm
|— webpack.config.js    # Configuration Webpack
|— nativescript.config.ts # Configuration NativeScript
|— .gitignore           # Fichiers à ignorer par Git
```

## Installation et Lancement

### Prérequis

- **Node.js**
- **NativeScript CLI** installé globalement :

```
npm install -g nativescript
```

```
npm install
```

### Exécution

- **Lancer l'application sur un émulateur Android :**

```
ns run android
```

## **API**

L'application communique avec une API REST pour la gestion des films et des utilisateurs.

### **Authentication**

- **POST /login** : Connexion (renvoie un token)
- **POST /register** : Inscription d'un nouvel utilisateur

### **Films**

- **GET /movies** : Récupère la liste des films
- **POST /reviews** : Ajoute une critique sur un film { movieId, rating, watched }

**Toutes les requêtes nécessitent un token d'authentification (Authorization: Bearer <token>).**