**Backend:**

Az alábbi dokumentáció részletesen ismerteti az alkalmazás kontrollereit és azok működését. A dokumentáció segítségével könnyen megérthető lesz az alkalmazás felépítése és működése, valamint hasznos útmutatást nyújt az új fejlesztőknek és a karbantartóknak.

**Movie API**

**Összegzés:**

Az alkalmazás egy filmadatbázis-kezelő rendszert valósít meg, amely lehetővé teszi a filmek böngészését, részleteinek megtekintését, értékelését és a felhasználók számára a profilkezelést.

**Controllerek:**

**AuthController**

A **AuthController** felelős a felhasználók azonosításáért, regisztrációjáért, kijelentkezéséért és törléséért.

* **login(Request $request)**: Felhasználó bejelentkeztetése.
* **register(Request $request)**: Felhasználó regisztrálása.
* **logout(Request $request)**: Felhasználó kijelentkeztetése.
* **deleteUser(Request $request)**: Felhasználó törlése.
* **getUsers()**: Összes felhasználó lekérdezése.
* **getUser(Request $request)**: Egy adott felhasználó lekérdezése.
* **modifyUser(Request $request)**: Felhasználó adatainak módosítása.

**HomeController**

A **HomeController** az alkalmazás főoldalát kezeli.

* **index()**: Főoldal megjelenítése.

**MovieController**

A **MovieController** felelős a filmekkel kapcsolatos műveletekért.

* **showMovies()**: Összes film lekérdezése.
* **showMovie($id)**: Egy adott film részletes megjelenítése.
* **show($id)**: Egy adott film megjelenítése.
* **updateRating(Request $request, $id)**: Egy adott film értékelésének frissítése.
* **calculateAverageRating($movieId)**: Egy adott film átlagos értékelésének kiszámítása.

**MovieLoaderController**

A **MovieLoaderController** a filmek betöltését végzi a külső API-ból.

* **loadMovies()**: Filmek betöltése külső API-ból.

**ProfileController**

A **ProfileController** felelős a felhasználó profiljának megjelenítéséért és módosításáért.

* **show($id)**: Felhasználó profiljának megjelenítése.
* **update(ProfileUpdateRequest $request)**: Felhasználó profiljának frissítése.
* **destroy(Request $request)**: Felhasználói fiók törlése.

**Models**

Az alkalmazásban használt fő modellek:

* **User**: Felhasználó model.
* **Movie**: Film model.

**Requests**

Az alkalmazásban használt HTTP kérések validálásához használt osztályok.

* **ProfileUpdateRequest**: Felhasználó profiljának frissítésére szolgáló kérés validációja.

**Middlewares**

Az alkalmazásban használt middleware-ek:

* **EnsureAdmin**: Ellenőrzi, hogy a felhasználó rendelkezik-e adminisztrátori jogosultsággal.

**Routes**

Az alkalmazásban használt fő útvonalak:

* **/login**: Felhasználó bejelentkezése.
* **/register**: Felhasználó regisztrálása.
* **/logout**: Felhasználó kijelentkeztetése.
* **/delete-user**: Felhasználó törlése.
* **/movies**: Összes film megjelenítése.
* **/movie/{id}**: Adott film részletes megtekintése.
* **/profile/{id}**: Felhasználó profiljának megjelenítése.
* **/update-profile**: Felhasználó profiljának frissítése.
* **/delete-profile**: Felhasználói fiók törlése.

### ProfileController

A **ProfileController** felelős a felhasználói profil kezeléséért. Ebben a vezérlőben találhatóak azok az eljárások, amelyek lehetővé teszik a felhasználói profil megtekintését, frissítését és törlését.

* **show($id)**: Ez a metódus a megadott felhasználó azonosító alapján lekéri és visszaadja a felhasználó profiladatait JSON formátumban. Ha a felhasználó nem található, akkor hibaüzenetet küld vissza a megfelelő státusszámokkal.
* **update(ProfileUpdateRequest $request)**: Ez a metódus felelős a felhasználói profil frissítéséért. A ProfileUpdateRequest osztály validálja a bemeneti adatokat, majd a metódus frissíti a felhasználó adatait az újonnan érkezett adatok alapján. Ha az e-mail cím megváltozik, az e-mail megerősítési dátumot nullára állítja. A frissítés után a felhasználót átirányítja a frissített profil megtekintésére, és egy üzenetet küld a státuszkóddal együtt.
* **destroy(Request $request)**: Ez a metódus a felhasználói fiók törléséért felelős. Ellenőrzi a felhasználó által megadott jelszót, majd ha helyes, kijelentkezteti a felhasználót, törli a fiókját, érvényteleníti a munkamenetet és újraindítja a munkamenet tokenjét. Ha a jelszó helytelen, hibaüzenetet küld vissza a megfelelő státusszámokkal.

### Dokumentáció

A dokumentáció részletes leírást nyújt az alkalmazás különböző részeiről, struktúrájáról és működéséről. A dokumentáció tartalmazza a funkciók részletes leírását, a használt technológiák, könyvtárak és függőségek felsorolását, valamint a fejlesztői környezet beállításának lépéseit.

A dokumentáció célja, hogy segítse a fejlesztőket az alkalmazás megértésében és fejlesztésében, valamint megkönnyítse az új fejlesztők bevezetését a projektbe. Fontos, hogy a dokumentáció naprakész legyen és minden releváns információt tartalmazzon az alkalmazásról és annak használatáról.