Drive-In Cinema API Documentation

Welcome to the Drive-In Cinema API! This API allows you to integrate with our drive-in cinema system, providing access to movie schedules, ticket bookings, and more. Whether you're building an app for movie enthusiasts or managing cinema operations, our API offers the tools you need.

Key Features

Data Storage: The application stores data in MySQL databse two primary data types with the following attributes:

- Films: *id, title, age_classification, language, cover_image*
- Film sceenings: *id, date, available_seats, film_id*

Endpoints

All of the endpoints starts with /api url prefix. Each endpoint returns a HTTP/200 with json response with data in it. Except if some operation was unsuccessfull, then it gives back HTTP/503 error response with the error message in it.

Film:

GET:

- /all : calls the FilmController's getAll() method which gives back all of the films in the database.
- /{id} : calls the FilmController's get(int id) method and gives back the film instance requested by the given {id} parameter
- /screenings/{id} : calls the FilmController's getScreenings() method and gives back the screenings related tot he film instance requested by the given {id} parameter

POST:

- /new : calls the FilmController's new() method which saves a new instance of a film with the given post data.
- /{id} : calls the FilmController's set() method which fills the film instance requested by the given {id} parameter and saves it.

DELETE:

- /{id} : calls the FilmController's delete(int id) method which deletes the film instance from the database requested by the given {id} parameter

Screening:

GET:

- /all : calls the ScreeningController's getAll() method which gives back all of the films in the database.
- /{id} : calls the ScreeningController's get(int id) method and gives back the film instance requested by the given {id} parameter

POST:

- /new : calls the ScreeningController's new() method which saves a new instance of a film with the given post data.
- /{id} : calls the ScreeningController's set() method which fills the film instance requested by the given {id} parameter and saves it.

DELETE:

- /{id} : calls the ScreeningController's delete(int id) method which deletes the film instance from the database requested by the given {id} parameter

All of the routes are protected by Laravel Sanctum.

Installation

1. First you need to clone the Git repository for the source files. (https://github.com/HUNAF-Maverick/Drive-In-Cinema)
2. Make sure you have Docker installed on your computer.
3. Then you need to run docker-compose up --build command from your terminal or command line interface of your operating system. It installs all the needed dependencies required to run the application.
4. The API is available threw http://localhost:9000 url