

MyFlix App

Objective

To build the server-side component of a “movies” web application. The web application will provide users with access to information about different movies, directors, and genres. Users will be able to sign up, update their personal information, and create a list of their favorite movies.

User stories

- As a user, I want to be able to receive information on movies, directors, and genres so that I can learn more about movies I've watched or am interested in.
- As a user, I want to be able to create a profile so I can save data about my favorite movies.
- As a user, I want to be able to edit my personal information as well as deregister whenever I want.

User Flow

- The user sends a GET method to /users
- API returns a list of ALL movies to the user

- The user sends a GET method to /users/title (title as the parameter)
- API returns data (description, genre, director, image URL, whether it's featured or not) about a single movie by title to the use

- The user sends a GET method to /genres/title (title as the parameter)
- API returns data about a genre (description) by name/title (e.g., “Thriller”)

- The user sends a GET method to /directors/name (name as the parameter)
- API returns data about a director (bio, birth year, death year) by name

- The user sends a POST method to /users including user data in the request body
- API returns data by which user registered

- The user sends a POST method to /login including user data in the request header
- API returns data by which user registered including a token
-
- The user sends a GET method to /users/username (username as the parameter)
- API returns data about specific user

- The user sends a PUT method to /users/username (username as the parameter) including updated data in the request body
- API returns updated data of the user

- The user sends a POST method to /users/username/movieid (username and movieid as parameters)
- API returns updated data of the user including the specific movieid in his favorite list

- The user sends a DELETE method to /users/username/movieid (username and movieid as parameters)
- API returns updated data of the user without the specific movieid in his favorite list

- The user sends a DELETE method to /users/username (username as the parameter)
- API returns a message of successful action

Key features

- Return a list of ALL movies to the user
- Return data (description, genre, director, image URL, whether it's featured or not) about a single movie by title to the use
- Return data about a genre (description) by name/title (e.g., "Thriller")
- Return data about a director (bio, birth year, death year) by name
- Allow new users to register
- Allow users to update their user info (username, password, email, date of birth)
- Allow users to add a movie to their list of favorites
- Allow users to remove a movie from their list of favorites
- Allow existing users to deregister

Note: The full documentation about this API you can find [documentation.html](#) placed in public folder