



Node.JS

Mid Project

You need to build a web site for managing Movies data.

All the data for you web site are managed by 2 sources :

- 1 . A REST API at <https://api.tvmaze.com/shows>
2. A JSON file called NewMovies.json that will store all new created movies data like : id, name, language and an array of genres ("Comedy", "Family" ...)
3. A JSON file called Users.json that will store all the users data that allow to work with the system. Each User Entry will have:
 - Username
 - Password
 - Created Date
 - Num of Transactions – The number of operations the user allows to do EACH DAY, for example : Get a movie data and create a new one are 2 Operations

The initial data in the Users.json is only the Admin Data. Only Admin can edit, create and update users.

Web Site Pages

1 – Login Page

This login page will check that the user credentials are valid AND the user still has "credits" for the current day (According to his "Num Of Transactions" data. If allowed – he will be redirected to the "Menu Page"



2. Menu Page

This page contains 2 links :

- "Search for movies" – will redirect to "Search Movies Page"
- "Create new movie" – will redirect to "Create Movie Page"
- "Edit Users" – **Will be presented ONLY for the Admin** – will be redirected to "Users Management Page"

3. Create new movie Page

This page will present a form for creating a new movie:

- Name
- Language
- An array of genres.

This data will be saved at the "NewMovies.json" file. His id will be set incrementally, for example :

If the web service holds 100 movies, the first movie in the json file will be with id of 101. The next one will be with 102 and so on..

A successful new movie saving will redirect to the "Menu Page" again.

4. Search Movies Page

This page will allow the user to search for movies by the following form:

- Name - A text box for a full or partial name
- Genre – A drop down with all genres
- Language – A drop down will all languages
- A "Search" Button.

The user can use one or all the conditions, pressing "Search" (count as 1 operation) and be redirected to the "Search Results" page.

This page also have "Back" link to the "Menu Page"



5. Search Results Page

This page will present a table with 2 columns :

The left column in each table row will present a link with a name of a movie that meets the search requirements.

The right column in each table row will present a list of movies names that are from the SAME GENRE as the current movie,

The data will be pulled from the REST API **AND** from the "NewMovies.json" file.

Each link click (both on the movies names in the left column and the movies names on the right column) will redirect to the "Movie Data" Page

This page also have "Back to Menu" link to the "Menu Page"

6. Movie Data Page

This page will present the requested (clicked linked) movie data, as follows :

- ID
- Name
- Genres
- Language
- A picture of the movie (only if pulled from the REST API)

This page also have "Back to Menu" link to the "Menu Page"

And a "back" link to the "Search results page"



7. Users Management Page (Can be accessed ONLY for the Admin)

This page will present a list of all users and a “Add New” link that will be redirect to the “User Data Page”

Each list item will present the user’s name (NOT as link), a “delete” link (for deleting him from the users.json file” and a “update” link that redirect him to the “User Data Page”

8. User data Page (Can be accessed ONLY for the Admin)

This page will a form fo user data:

If been redirected from the “update” link, then the form will be populated with the user’s data and an “Update” button

If been redirected from “Add New” link, the form will be empty and with a “Save” button for saving a new user in the users.json file.

Notes :

1. At each stage, if the user has no more credits for that day, he will be redirected to the “login page” again with a proper message

Counted operations are :

- Search movies
- Creating a new movie
- Click on a movie link

The Admin has NO LIMIT on the number of operations.

2. Use Proper controllers, routers, Services Layers, models etc..