Watchlist Web Application

Notice: Before running the app, you need to download the requirements (saved in requirements.txt) using pip and/or a virtual environment.

Mongo DB collections:

In the database called “watchlist\_app”, there exists the collections:

* Shows: holds the movies and series in the database. Example:

{

"\_id": {"$oid": "61e28811c5c9d9ac5c9b968b"},

"episodes\_num": 0,

"genre": "Adventure",

"name": "Star Wars: A new hope",

"rating": 4.724,

"rating\_number": 25,

"type": "Movie"

}

Note that:

* + Rating is between 0 and 5 inclusive
  + Rating number holds the number of reviews
  + The name is always unique (mongo index has been created as unique)
  + Episode number is 0 for movies and more for series.
* Users: Holds the users and their watchlist. Example:

{

"\_id": {"$oid": "61e2b3942d30979d7a175335"},

"username": "LDC",

"full\_name": "Lionardo Di Caprio",

"hashed\_pass":"$5$rounds=535000$Aws.x2BWvgiRCy7P$AlnrfL0.crGHlCLpsQ0FTP0E5mEGCTUi/98x5TBuL1C",

"watchlist": ["61e281dd174a1790ee2e8e32", …]

}

Note that:

* + Username is unique
  + Password is hashed
  + Watchlist is an array of strings where each string is the Object id of a show

Authentication:

All paths that use authentication use JWT authentication: add “Bearer …token…” in the authentication header. (or write it in the box in the swagger ui)

Paths that use authentication:

* GET: /user/watchlist

Gets a user’s watchlist, private to the user

* POST: /user/rate

User can rate a show on a scale of 0 to 5 (as integer)

* PATCH: /user/watchlist/{show\_id}

Add show to user’s watchlist (watchlist elements are unique , show can only be added once, unless it was deleted)

* DELETE: /user/watchlist/{show\_id}

Delete show from user’s watchlist

Functionalities:

* Register user
* Login user
* Add show
* Delete show
* Get a show
* Get Shows (alphabetical order by name implemented with pagination)
* Get user’s watchlist
* Delete show from user’s watchlist
* Add show to users watchlist
* User can rate a show on a scale of 0 to 5 (as integer), that affects the current rating in the shows collection

RUN THE APP

1. Go to codegen server

2. Run: python -m openapi\_server

3. Go to: <http://localhost:8080/ui> to get the swagger ui

4. Run the route you want.

Publish on docker:

Go to the location that holds the docker file (ie. codegen server)

1) docker build -t docker-image .

2) docker run -dp 8080:8080 docker-image

3) Go to <http://localhost:8080/ui> to get the swagger ui