imageAPI Documentation

Specification

Frontend: React

Backend: Express/Node.js

Database: Mongodb/Mongoose

Demo video

https://drive.google.com/file/d/1RQLIJUaHo-N1WNMhZpxZnU5SM52Ui65n/view?usp=sharing

Initialize Project

With Docker

Open terminal and run the following steps:

(To clean up previous docker image, please run `docker-compose down -v`)

1. Go to project repository:

   `cd imageAPI`

   Open docker

   `open -a docker`

   Run make file script or use docker command

   `make run-dev`(`docker-compose up`)

   Open browser, go to: http://localhost:3000/

Without Docker

1. Edit code in `server/index.js` and change the code as follow.

```
mongoose.set("strictQuery", false);
mongoose
    .connect("mongodb://localhost:27017/imageAPI")
    // .connect("mongodb://mongo:27017/imageAPI")
    .then(() => {
        console.log("Connect to mongodb!");
    })
    .catch((e) => {
        console.log(e);
    });
```

2. You need to open two terminal, react, nodejs, mongodb need to be installed in advance.

First terminal:

```
cd imageAPI/server
```

```
npm install
```

```
npm run start
```

Second terminal:

```
cd imageAPI/client
```

```
npm install
```

```
npm start
```

Open browser, go to: http://localhost:3000/

Dockerfile

Frontend/ Backend image - node:16-slim

https://github.com/nodejs/docker-node/blob/3f8018043408490439723ed3b71ab5578d69ea70/16/buster-slim/Dockerfile

Database - mongo:4.4-bionic

https://hub.docker.com/layers/amd64/mongo/4.4.0-bionic/images/sha256-ebd31eaac273a9544a33387aa859b0a8676565340a40fc824fa7bda686f462f1?context=explore

Structure Diagram

home-component.js
General home page

login-component.js
Login page

register-component.js
Registration page

profile_component.js
User main page.

share-component.js
share image page

components
User interface
functional components

index.js
Export all routes

auth.js
Export login/register route

image-route.js
Export image route

share.js
Export share link route

routes
Wrap up controllers
attribute to to server

App.js
Entire design of React frontend

index.js
Wrap up App.js and sent it to index.html

src
All the js file for React

client
Frontend React framework

server
Backend Express server

models
Schema for attributes in database

image-model.js
Image schema

user-model.js
User schema

auth.service.js
Access user information from backend
and authenticate the correct user

image.service.js
RESTful API for image and access
database data through backend server

share.service.js
Access date from backend and
post image in the share link

services
Front-end access from back-end

insex.js
Wrap up app.js with mongoDB
and send it to local host

uploads
Store the image array

validation.js
Validate user login and register input

config/passport.js
Data protection via JWT token