**# simetia-backend**

This is library rest api made by nodejs, express, typescript, postgresql and jwt.

This doesnt use ORM.

This project is done using typscript and express.

1. Database setup

- You need to install postgresql on your pc.

- Create a database named as `library`

- Then download or clone the source code and in the project directory, run this command. `npm install`

- Then run `npm run setup`(This will create books table and insert 3 rows.)

- Then run `npm run dev` to run the project after ctrl+c.

- You can see the following output in terminal.

  [✔️] User table created successfully.

  [✔️] Books table created successfully.

  [✔️] User-To-Book table created successfully.

- Database structure.

  There are 3 tables for this project.

  books table: id, title, pub\_year, description

  users table: uid, email, password, created\_date

  users\_books: user\_id, book\_id (For management of favorite books of each user. many-to-many relation)

2. Signup/Login

- Users are created when they are signed up using their email and password.

- First registered user is considered as admin of this application.

- Other users are considered as general users.

3. Authentication

- Admin can add or remove book.

- General users can add or remove book one at a time into their favorite books.

4. Endpoints

- If you run using `npm run dev`, you can see `Running on port 5000.`

- If you type `http://localhost:5000/books` (GET) in your browser, you can see all the books listed.

- http://localhost:5000/users/signup (POST) email and password are needed. You can use postman to test it.

- http://localhost:5000/users/login (POST) email and password are needed. If login successful, you can get the following.

  {

  "id": "c8feaef5-c2c9-409a-b0bf-d8e1e0345a20",

  "email": "test2@gmail.com",

  "password": "$2b$08$y66uwL8/HfwAhFANs0md..kTLk7bOm.vrz.rH9UGElWds.FiPM7Fm",

  "is\_admin": true,

  "created\_date": "2021-04-25T19:28:03.110Z",

  "modified\_date": "2021-04-25T19:28:03.110Z",

  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJjOGZlYWVmNS1jMmM5LTQwOWEtYjBiZi1kOGUxZTAzNDVhMjAiLCJpYXQiOjE2MTk0MTE0MDEsImV4cCI6MTYyMDAxNjIwMX0.1buWPx3DL1KyrUpMAkpQ-9ciApTrWSjJM6HQfloluy8"

  }

  You can use id field to add/remove favorite books using the token.

- http://localhost:5000/users/:uid/add (POST)

  Here you should use uid as the id field in over login result and insert token value into the headers x-access-token field.

  body {"id":4}

- http://localhost:5000/users/:uid/remove

  In the same way as previous one, you should include token and uid.

  body {"id":4}

Above mentioned are addding/removing favorite books of each user.

- Admin:

  Only first registered user is considered as admin. These actions will add/remove the books listed.

  endpoints: http://localhost:5000/books/add (POST)

  body: {

  'title':'anything',

  'pub\_year':2021,

  'description': 'anything about this book...'

  }

  header x-access-token : The token you get by using login with first registered email and password

              http://localhost:5000/books/remove (POST)

             body: {

                 'id':3

             }

             header x-access-token : The token you get by using login with first registered email and password

If you have problem, please message me on skype or email.

My email is xiaoping.jin.210@gmail.com and skype is live:.cid.86d645b8f636cba6

Thank you.