2021

CAB230 REST API – Server Side

CAB230 Happiness API – The Server Side Application

<Hyeonwoo Lee/s>

<n10661531/s>

11/6/2021

Contents

Introduction

Purpose & description

In this project I tried to make an API server side, which is for assignment 1's website. We should availably make local host web and allow to put in SQL data and shows it. By this website will be available to get data from API especially local SQL data base. This severs side get Data from SQL data base and send to website by show as API website.

Completeness and Limitations

/rankings

Success with get error code and available to get every case of sources by queries. Swagger interacted well.

/countries

Success with get error code. And Success with get sources. Swagger interacted well.

/factors/{year}

available to get every case of sources by queries and parameters.

/user/register

Success with get error codes. And Success with to do register. Swagger interacted well.

/user/login

Success with get error codes (without status 200 case worked well in swagger). And Success with to do register. Swagger interacted well.
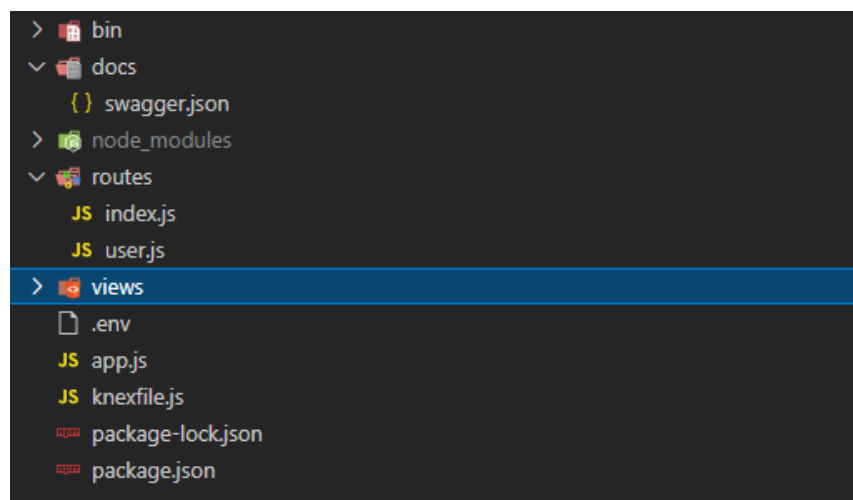
/user/{email}/profile

Available to get and put sources if put table in it, however we must add user table columns. If column added, login and register does not work.

Modules used.

No additional modules used. (all from black board,, post man,, git hub test,,)

Technical Description

Architecture

Firstly, the structure of file is basically followed when I express at first time index.js and user.js are in routes file because those are for router. .env file supposed to be outermost side it is just in my project file not in any specified directory. App.js file is main program so it will be good if it is in main page. Knex file as well to interact other files, it is good to located at project main file. Swagger.json file is in docs directory which will be used localhost:3001/docs like this.

Security

•   Use of knex or other query builder without raw SQL

```
const options = require('./knexfile.js');
const knex = require('knex')(options);
```

Used knex

•   Use of helmet with at least the default settings enabled

```
const helmet = require('helmet');
```

used helmet and the default settings enabled.

•   Use of morgan with a logging level similar to that used in the pracs.

```
var logger = require('morgan');
```

```
logger.token('req', (req, res ) => JSON .stringify(req .headers))
logger.token('res', (req, res ) => {
  const headers = {}
  res.getHeaderNames().map(h => headers[h] = res.getHeader(h))
  return JSON .stringify(headers)
})
```
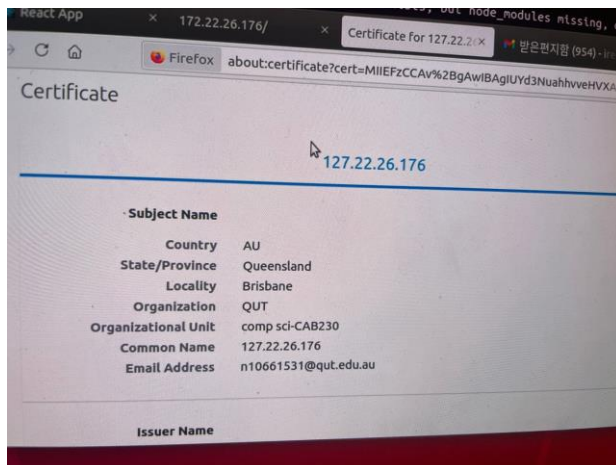
Used morgan based on pracs

•   Appropriate handling of user passwords as described in the JWT Server-Side worksheet.

```
knexfile.js > [@] <unknown>
1    module.exports = {
2      client: 'mysql2',
3      connection: {
4        host: '127.0.0.1',
5        database: 'happiness',
6        user:'root',
7        password: "xxxxxxxxx"
8      },
9    }
```

Appropriate handling data in knex file to the JWT Server-Side.

Worked well on security part

Testing

In this section we expect you to screenshot the html test report:

Since login it is all error, seems something went to wrong to login and authorize.

Difficulties / Exclusions / unresolved & persistent errors /

Major roadblock is to catch errors from each router. especially from rankings and factors. I figured out with if statement from Ranking, however I failed to catch some errors from factors. And one of bug that I did not solve is from error of login's status. From swaggers, it worked properly by 200 statuses however, I do not response in test report.

Installation guide

Download zip file and Open with visual studio and open terminal and type 'npm start'.

Then it will be allowed to access in local host + link. However, recommend to access localhost/docs which is swagger.

Pic of swagger.