Noroff Backend Exam Project

Backend solution for RestAPI for Exam Project Assignment.

Created with ExpressJS. No views or view engine are included, RestAPI endpoints only.

Installation and Usage

Application require NodeJS, and a MySQL database for persistence.

1. Clone or fork this repo.
2. Run npm install to install packages.
3. Set up database and set required environment variables *(see example below)*
4. Run application least once to migrate database models from sequelize to database.
5. SEND a POST **/setup** to fetch or create initial data.

Setting db logging should be set to false in production!

Credentials for admin is currently hardcoded in **AuthService.createAdmin()** and meet expectations found in the Course Assignment Resources.

Important:

To successfully run tests, a database must be connected and the application must run at least once to migrate datamodels to database.

Run Scripts:

- **npm start** : Run application in production
- **npm run dev** : Development mode with nodemon
- **npm test** : Run tests

Known issues:

- Category *"Laptop"* **was not** provided as category name from the initial data. Thereby searching for for all items with category *"Laptop"* will not produce one item as indicated in the exam instruction. To meet the exam instructions, the test is written to expect at least one result. Unless provided data is changed: THIS TEST WILL FAIL!
  - *instruction reference:* 7. post /search – search for all items with the category name "Laptop" (one item should be returned from the initial data).
  - *source data* as json file with the content and format of data available when this solution was created.

Environment

Application using environment variables. In development, variables are extracted from a local .env file. Example below lists environment variables.

```
HOST = "localhost"                   # Database host
ADMIN_USERNAME = "admin"             # Db admin account
ADMIN_PASSWORD = "admin_password"    # Db admin password
DATABASE_NAME = "db_name"            # DB name
DIALECT = "mysql"                    # SQL Dialect
PORT = "3000"                        # Application port, default = 3000
DB_LOGGING = false                   # Sequelize SQL Logging enable/disabled,
                                     #   default = False
TOKEN_SECRET = secret token          # Token Secret
```

Packages

Packages used in this application are listed here for easy documentation lookup.

- ExpressJS : Application Framework
- Sequelize : DB Orm
- MySql2 : MySql library
- Jsend : Middleware for response.
- JsonWebToken : Implementing JWT for Auth.
- dotenv : loading environment variables-
- validator : used for validating data.

Dev packages:

- Nodemon : Restart on change in dev mode.
- Jest : Test library
- Supertest : Test library

Acknowledgment

Solution code is based on documentation listed in packages above or by previous assignment solutions defined by course material. Other sources are listed below.

Documentation

The **Documentation** has been compiled from three PDFs as required by the course assignment.

The PDFs have also been included for this solution under *./documentaion/parts/* and include - This README.md converted to PDF. - Standalone Postman Documentation.pdf - The Retrospective Report - Postman Documentation Online

This solution was created in accordance with Course Assignment Resources given for this exam project. PDF printout contain timestamp. Changes made to the assignment text after this time and date, have been ignored, with the exception where I have received direct and explicit notification from those responsible about changes that have been made.