Backend code demo project

idea-case-backend (Not model for all parts, but contains several good ideas)

19.9.2022

There are max 11 files you need to look into

- **README.md**

- **.gitignore**

- **.env**

- **package.json**

- **src/index.js**

- **src/routes/index.js**

- **src/routes/category.js**

- **src/db/index.js**

- **src/responseHandlers/index.js**

- **src/utils/logger.js**

- **logs/winstonBackendLog.log**

Haaga-Helia

README.md

- *Write here how to setup/install/clone/run the project.*

- *Possibly introduce the libraries and give links to information about each library*

- *Here are some Markdown markup (yep, the name is playing with words) syntax examples: https://github.com/haagahelia/swd4tn023/blob/master/08_seminaari/markdown_file1.md*

.gitignore

- *file that lists what files and folders will not be tracked nor added to the git repository*

- *e.g. we do not want to put the **.env** file anywhere in public place, it might contain secrets, like ip addresses, usernames, passwords…*

  - *Thus you will also not get it with git clone.*

    - *Thus you will manually add that file to your project*

- *if you forget to enter the file / folder in the gitignore the non-ignored file will be in the version history, and could be exposing secrets in remote repo, like github!*

Haaga-Helia

.env file

- *The **.env file (1.)** that the dotenv library (see in the package.json (2.), added there with **npm install dotenv**) reads from file system and offers as 'environment variables' to other modules with this kind of code **process.env.VARIABLE_NAME_HERE (3.)***

- *Here we can keep the settings we want to be able to change without changing the tested code*
  - *e.g. IP address of the database server*
  - *or usernames and passwords*

- *per developer file, thus could also contain developer's preference, like ip to own version of database*

Haaga-Helia

package.json

- *Created when the Node project was created, e.g. with the **npm init** command*

- *Locate e.g. the '**npm start**' script and see how it will run '**node src/index.js**' / '**nodemon src/index.js**'*

- *See also the dependencies and development dependencies*

    - *New libraries can be added there like this: **npm install dot_env***

    - *When you have cloned a project, the dependencies mentioned in package.json can be installed with the **npm install** command*

- *package.json file you want to keep in git, as it's important file that makes a Node/npm project to be a project and also e.g. keeps track of the libraries*

- *(…whereas node_modules folder and package-lock.json file can always be removed if needed for fixing some probs. **npm install** creates them again)*

Haaga-Helia

src/index.js

- *This is the starting point of the Node backend.*

- *Not the starting point of end-point routing, but a file that setups up and configures the Node project*

- *Things here are usually run only ones, at the server startup*

- *Identify where the all the API routing starts!*

Haaga-Helia

src/routes/api/index.js

- *Just routing the endpoint calls to specific routing files, based on part of the URI pattern.*

- *Common file, add your entries to common files fast and deliver them to others also fast*

Haaga-Helia

src/routes/api/category.js

- *An example how the category business object related REST API endpoints are gathered in one file*

  - *Other options could be role based bundling of the endpoint*

- *Now there are about nine endpoints, and file length ~260 lines*

  - *Thus maybe could be divided into category.js and categoryMultiple.js (or something like that),*

  - *first one with endpoints adding, updating, finding, deleting **single** categories. The other one having all the endpoints related to **multiple** category objects.*

- *Here only look at the five endpoints marked with >>> for Exam*

  - *Why? Because they follow the new idea of responseHandler and logging*

  - *Why aren't all endpoints in this project with the new model? That would take time, the five already demonstrate the new model well*

  - *Old examples are also interesting for comparison. They are remarkably longer and have unnecessarily repeating code lines.*

src/db/index.js

- *File that (1.) defines a database configuration object*

  - *(Though the database settings are also originally from the .env file)*

- *…and (2.) exports a database connection object (we call it this time 'knex') based on that configuration object*

- *Other code modules can then execute database operations with that knex object*

Haaga-Helia

src/responseHandlers/index.js

- *File that unifies the handling of the response, after database operation has been completed, either successfully or unsuccessfully*

- *Kind of a utility library, reusable functions. Elsewhere we save a lot of code writing*

- *Writes also to the log using the logger.*

- *Finally gives the caller (e.g. our Frontend, or some test system, like PostMan client) the HTTP response too.*
  - *In case of total success from business point of view: "200" and the possible data*
  - *In case of error that can be somehow pointed at the request: "400" and "Request error"*
  - *In case or error that can be somehow pointed at the backend server or database: "500" and "Server error"*

Haaga-Helia

src/utils/logger.js

- *File first defines configuration for the winston logging library*
    - *E.g. we can define what is the least serious message level we write to log file / console*
    - *Logging levels are: error, warning, info, verbose, silly.*
        - *If you set level to silly, you get all of the above*
        - *If you set level to info, you only get error, warning and info*
- *Then the file creates and exports a logger object based on that configuration*

Haaga-Helia

logs/winstonBackendLog.log

- *File that keeps history of the logged messages*

- *We don't put theis to git repository as it will be generated when needed. That means, when the application is running*

- *From time to time we might empty the log file*

- *This log could be improved by putting date and time stamp to the front of each message. Important fact for debugging.*

Haaga-Helia