Mensa Microservice

A microservice for a Chatbot developed at the Beuth University of Applied Sciences Berlin

Table of content

Getting Started

Prerequisites

- node.js
- express.js

Cloning

Get the source code by cloning its repository via https: mensa_microservice

Installing

After cloning the repository, you will need to make sure that you have node and npm installed on your working system. To check if you already have node installed, try

```
node –version
```

Same for checking if npm is installed, just with npm instead of the node command

```
npm –version
```

If you don't have node or npm installed, download the Softare via the links provided in Prerequisites or search for them via your preferred search engine.

After that install all necessesary dependencies

```
npm install
```

Now you can start the local development server to play around with the API and make your calls

```
npm run dev
```

This will fire up a development server that listens on port 8000.

If you direct your browser to http://localhost:8000/meals, you will get a list of the meal plan of the actual day for the mensa at the Beuth University for Applied Sciences.

Overview

The mensa microservice is basically a *Node-Express*-Backend. Incoming requests are checked and specifically handled.

Structure

The microservice consists of four folders containing several scripts, which are designated to perform certain tasks. We have the **scripts**-folder containing scripts, that will be called by cron-jobs mainly for caching purposes. Then we have the **services**-folder containing files, that consist of functions useful to process incoming requests from the chatbot and to generate a formatted answer-string, that contains the requested meal-menu of a specific day from the Beuth mensa. The **routes**-folder consists of all the routes, that can be addressed. In the next chapters we will get into more details about the scripts and their functions.

Functionalities

On request, this microservice makes calls to the OpenMensa API. The received data is processed by services that give a list of filtered and unfiltered meals of the mensa of the Beuth University for Applied Sciences. Mainly this service was built throughout the Masterprojekt module that is a mandatory part of the media informatics master course of the Beuth University for Applied Sciences.

The scripts-folder

This folder contains two scripts, that will be called by a cron job once a day. Probably early in

the morning. `getMealsOfTheDay.js` makes a request to the OpenMensa API and caches the answer. After that `writeResponseFile.json` is called and generates a pretty formatted answer-string. Now everytime a User wants to know the meals for the day, we can just read it out of the cached data.

The services-folder

This folder consists of several services, that perform specific tasks for the microservice.

generateResponse.js

Creates a nicely formatted string from a mensa-JSON-object and caches it.

mealService.js

Makes a request to the OpenMensa-API and caches the response. It can also filter the file for specific meals. For example only vegetarian or vegan meals, etc.

mealsOfSpecificDayService.js

If a day other than the current day is requested, we need to make another request to the OpenMensa-API, fetch the meals for that specific day, cache them and maybe need to filter them. This is all done by this script.

The routes-folder

This folder contains all the routes, that can be addressed on this server. The `index.js` manages all the routes. We've only got two routes in our project. The `/swagger`-route leads you to the swagger documentation of this project.

The `/meals`-route will be called by another component of the Beuthbot via `POST`. It expects a message JSON-object containing the requested date for the meals and the filters - to request only specific meals. It then calls all the functions needed to perform requests and generates an answer, which is finally send back as a response to the Chatbot.

Further Development

This is still a work in progress, so functionalities and structure might still change during development

Further Reading

- OpenMensa API

Built With

- Node.js
- Express.js
- Axios

Versioning

We use SemVer for versioning. For the versions available, see the tags on this repository.

Authors

- **Tolga Karaoglu**
- **Steven Sobkowski**

See also the list of contributors.