Päevapakkumiste juturobot (pp-bot)

Lunch offers chatbot readme document. This document explains the contents of the files and how to run the application locally.

Architecture

The source code of the application is in the /src directory.

/src directory

- `commands.ts` - request handlers for incoming requests by Slack slash commands.
- `constants.ts` - holds constants used throughout the application, to make changing them easy and keep other files cleaner.
- `controller.ts` - request handlers for GET endpoints
- `db.ts` - connects to the db and is responsible for all queries that are made to the mysql database and for transforming the data to the right type.
- `functions.ts` - helper functions that do not belong elsewhere.
- `queue.ts` - defines the queue and function to initialize it and add the server routines to the queue(repeatable jobs)
- `recommender.ts` - is responsible for integrating with the pp-classifier recommender service, that helps classify meals and and make recommendations to the users.
- `server.ts` - this is the entrypoint of the application, which starts the server, initializes the queue, defines the routes and attaches listeners. This is started by executing `npm start`
- `service.ts` - contains the functions that call the services and save the results to the database.
- `types.ts` - contains data interfaces
- `worker.ts` - responsible for processing jobs on the queue. This is started as a separate process from the main server by executing `npm run worker`.

Job handlers

The `/handlers` directory contains job handlers. In `workers.ts`, every job is attached to a handler with the same name.

Slack

The `/slack` directory contains files that handle communication with Slack.

- `suggestionsMessageBuilder.ts` - Builds the blocks for Slack message payloads.
- `requestListener.ts` - handles requests that Slack makes to the server when a user interacts with the messages the app has sent.
- `service.ts` - handles exchanging the temporary authorization grant for a an authorization token, also has a helper function to extract data from request Slack makes to the server on slash commands.
- `bot.ts` is uses message builder to build and finally send the various different messages to a Slack user.

Files outside of /src directory

- `.env` - this file is not in the repository, but is needed to run the application locally.
- `.env.example` - an example configuration file with all the required variables left blank.
- `.eslintrc.js` - ESLint linting configuration file. Linting helps keeps the code quality to specific set of rules.

- `.gitignore` - contains pathname patterns that need to be excluded from the repository.
- `.prettierrc.js` - ESLint plugin Prettier configuration file. Prettier helps keep the code formatted to a specific set of rules.
- `DOKKU_SCALE` - defines the how many processes of each that are defined in `Procfile` will be started when deployed to Dokku.
- `package-lock.json` - saves the exact package versions that installed during `npm install`. Is used when deploying to install the same exact versions that were used for development.
- `package.json` - contains the scripts, runtime and development dependency requirements of this project.
- `Procfile` - specifies the commands that are executed to start the app. Relevant to Dokku (also handled the same way in Heroku).
- `tables.sql` - contains the sql for setting up the database.
- `tsconfig.json` - TypeScript compiler configuration.
- `tslint.json` - TypeScript linter configuration.

Compiled code will be in the `dist` directory and dependencies will be downloaded to `node_modules` directory by the npm.

`.vscode/settings.json` contains Visual Studio Code related settings to run formatting on the code when saving a file.

Development

- You need to have a local redis and mysql database.
- You need to set config values in `.env` following the config example file `.env.example`.

You can then run the application on your local machine after running `npm install` by running these three commands in separate terminals:

1. `npm run watch-ts` This watches for changes in source files and recompiles when detected.
2. `npm run watch-node` This watches for changes in the compiled `.js` files and restarts the server when detected.
3. `npm run watch-worker` This watches for changes in the compiled `.js` files and restarts the worker when detected.

Commands or actions from Slack will not make it to your local server, but sending messages to Slack works. It is possible to set up a secure tunnel to your local app from the internet using `ngrok`, but it was not used for development of this application and is not covered here.

Known bugs

- It is possible for the user to add so many favorites, that it will exceed the Slack message block limit (which is 50).
- Even though exponential backoff is used, it is possible that a meal fetch job fails 3 times in a row. Users with that city selected will not be able to get suggestions or search results on that day.
- Make sure the server time is set to your local time, otherwise suggestions will be sent at the wrong time.