Creating the Environment

The repository contains the file `.env.template`. This file is a template for the environment variables that need to be set for the application to run. Copy this file into a file called `.env` at the root level of this repository and fill in all values with the corresponding secrets.

To create the virtual environment in this project you must have `pipenv` installed on your machine. Then run the following commands:

```
# for development environment
pipenv install --dev
# for production environment
pipenv install
```

To work within the environment you can now run:

```
# to activate the virtual environment
pipenv shell
# to run a single command
pipenv run <COMMAND>
```

Build Process

This application is built and tested on every push and pull request creation through Github actions. For this, the `pipenv` environment is installed and then the code style is checked using `flake8`. Finally, the `tests/` directory is executed using `pytest` and a test coverage report is created using `coverage`. The test coverage report can be found in the Github actions output.

In another task, all used packages are tested for their license to ensure that the software does not use any copy-left licenses and remains open source and free to use.

If any of these steps fail for a pull request the pull request is blocked from being merged until the corresponding step is fixed.

Furthermore, it is required to install the pre-commit hooks as described here. This ensures uniform coding style throughout the project as well as that the software is compliant with the REUSE licensing specifications.

Running the app

To run the application the `pipenv` environment must be installed and all needed environment variables must be set in the `.env` file. Then the application can be started via

```
pipenv run python src/main.py
```