

Deployment Notes:

General:

The git repo contain two projects: the frontend and the backend project. The package.json files of the two projects have been configured, so that the two projects have to be deployed separately. We tested the deployment scripts in the package.json files by deploying our web application to heroku.

Our approach was to treat the two projects as two separate heroku apps and only pushing the respective subfolders frontend and backend. To achieve this we were not able to use the user interface on heroku.com but rather had to use the heroku cli and the workaround found in this tutorial:

<http://adampaxton.com/how-to-deploy-to-multiple-heroku-apps-from-the-same-git-repository>

We used this approach to deploy our web application to heroku. Our instance can be found under:

- frontend: <https://morning-peak-96987.herokuapp.com>
- backend: <https://obscure-waters-56880.herokuapp.com>

Necassary Changes:

There are values to be adjusted before deploying the two projects in order to ensure that the two heroku apps can communicate successfully.

Frontend changes:

- change baseUrl environment variable in environment.prod.ts file in frontend/src/enviroment to the url of the backend app

Backend changes:

- add baseUrl of frontend app to the allowedOrigins Array in backend/app/server.ts
- change the secret in backend/app/config.ts to a more secure value

If you wish to start with an empty database:

- delete the file backend/db.sqlite
- if you do start with an empty database a new admin user can be manually created by using the following curl command:
 - `curl -H "Content-Type: application/json" -H "Accept: application/json" -X POST --data '{"email": "<your admin email>", "password": "<your admin password>"}' "<baseUrl of backend>/user/init"`
 - This interface only works as long as the user database is empty.

Note that all changes need to be committed at least locally in order to push them to the heroku git.