Developer Documentation for Napier Map Elites

William Hutcheson

July 17, 2018

Contents

1 Maintenance on commute.napier.ac.uk

These commands use absolute paths which exist on commute.napier.ac.uk. The paths in these commands will likely need to be changed if running on a different machine.

1.1 Switching from testing to production

```
docker-compose down
docker-compose -f
 ↪  /home/40180582/Napier-Map-Elites/docker-compose.production.yml
 ↪  build
docker-compose -f
 ↪  /home/40180582/Napier-Map-Elites/docker-compose.production.yml
 ↪  up -d
```

1.2 Switching from production to testing

```
docker-compose down
docker-compose build
docker-compose up -d
```

1.3 Updating SSL certificates

The server SSL certificates are generated using LetsEncrypt[1]. To update them run the following commands.

```
docker-compose down
docker run --rm -it -p 80:80 -v
 ↪  /etc/letsencrypt:/etc/letsencrypt certbot/certbot
 ↪  renew
cp /etc/letsencrypt/live/commute.napier.ac.uk/*
 ↪  $SSL_LOCATION
docker-compose -f
 ↪  /home/40180582/Napier-Map-Elites/docker-compose.production.yml
 ↪  up -d
```

1.4 Manually adding a solutions zip file

To manually add a solutions file to the server you simply need to copy it to `$SOLUTIONS_FOLDER`. An example is shown below where it is assumed `$SOLUTIONS_FOLDER` has been set to the same value as the remote server and `$SOLUTIONS_FILE`, `$CSV_FILE` and `$REMOTE_USERNAME` have been set.

[1] https://letsencrypt.org/

```
mv "$SOLUTIONS_FILE" "$(md5sum $CSV_FILE | cut -d ' ' -f
↪  1).zip"
scp "$SOLUTIONS_FILES"
↪  "$REMOTE_USERNAME@commute.napier.ac.uk:$SOLUTIONS_FOLDER"
```

2 Code issues and considerations

2.1 Flask Docker Boilerplate

This project is based on a template Flask Docker Boilerplate[2] which is actively maintained by myself. This will likely continue to be updated with bug fixes and new features. Pulling the master branch of this in to the project intermittently could be wise. These changes are not guaranteed to be non-breaking and so testing should be done before pushing these changes up to production.

2.2 Parcoords-es

The parallel coordinates code is forked from https://github.com/BigFatDog/parcoords-es. This has been updated to include a single point brush. The fork is available at https://github.com/hintofbasil/parcoords-es. This code has not been submitted as a pull request due to slight issues in the quality of the code.

This code will likely not be maintained and so a fork should be created. This should try and pull in changes from the original repository which is still actively maintained.

This library is not included in the webpack build script due to incompatibilities. Currently the file 'dist/parcoords.standalone.js' must be copied to 'flask/app/static/js/parcoords.js' whenever the library is updated. To include the sourcemaps copy the file 'dist/parcoords.standalone.js.map' to 'flask/app/static/js/parcoords.js.map'.

2.2.1 Changes

The core changes to the app are all contained in the 'src/brush/points/' folder. Some changes were also made to index.js but only to enable the new points brush.

2.3 Napier Map Elites

The code for the website is contained in the flask/app folder. The following is a breakdown of the folder structure.

main.py The main entrypoint for the code.

config.py The configuration file for the project. This contains three configurations which are loaded based on what $APP_SETTINGS is set to. This is loaded in main.py. If a key error is found when loading the application this is the likely location and the likely cause is an unset environment variable.

images The location to save images. During development these must also be copied to static/images however this is not required in production builds. This will likely be fixed in a future update of Flask Docker Boilerplate.

[2] https://github.com/hintofbasil/Flask-Docker-Boilerplate

js The location to store Javascript. Files in this directory are compiled into files in the static/js folder. Files in subfolders are not compiled but can be included in other Javascript files.

sass The location to store sass/scss files. Files in this directory are compiled into files in the static/css folder. Files in subfolders are not compiled but can be included in other sass/scss files.

solutions The default folder to save solutions into in testing and development builds.

static The folder compiled static files are saved in to. This is in the .gitignore however some files have been force added as they don't compile nicely with existing Javascript files. The build process will likely be updated in the future to allow these to be required in other Javascript files.

templates The folder for html templates to be saved in to. These are rendered by Flask using Jinja2 [3].

views The folder for python code to create endpoints to be saved into. If new files are created they must be added into main.py.

2.4 Max file size

The maximum file size is configured in various different places. To change this is must be updated in all these places.

flask/app/config.py The setting is called MAX_CONTENT_LENGTH. This is set in bytes.

The following files all contain a settings called **client_max_body_size**. This must be set to the same value in all places.

- flask/etc-http/nginx/nginx.conf
- flask/etc-http/nginx/conf.d/http.conf
- flask/etc-production/nginx/nginx.conf
- flask/etc-production/nginx/http.conf
- flask/etc-production/nginx/https.conf

After these changes have been made the docker container must be restarted using the following commands from the root of the project.

[3] http://jinja.pocoo.org/

```
docker-compose -f docker-compose.production.yml build
docker-compose down
docker-compose -f docker-compose.production.yml up -d
```