
slb59.ocp13

Release 1.0

Sylvie

Dec 21, 2023

CONTENTS:

- 1 Technologies 3**
- 1.1 Installation 3
- 1.2 Application Description 4
- 1.3 Database 5
- 1.4 Development Process 6
- 1.5 deployment Process 8
- 1.6 Indices and tables 9

Orange County Lettings is a real estate rental company. On the site, you can view several rental locations as well as user profiles.



TECHNOLOGIES

It is an application written with Django in python language. The data is stored in the SQLite3 database.

The different tools used for the development and deployment of this application are:

- Gitlab for project management, issue board and milestones
- Visual Studio Code for the development
- Git for storing code and versioning
- Sentry for monitoring site performance
- Docker and docker desktop for code containerization
- Gitlab for continuous code integration and delivery
- Read the docs, to publish the documentation
- AWS as runner for CI-CD process
- Render for déploiement on a public url

1.1 Installation

These steps describe how to install your development environment.

1.1.1 Clone the GitHub repository

```
git clone https://github.com/S1b59/0c-P13.git .
```

1.1.2 Create the virtual environment

Pipenv is a Python virtualenv management tool that supports a multitude of systems and nicely bridges the gaps between pip, python and virtualenv.

- mkdir .venv
- rename the file .env.example en .env
- change variable values to suit your configuration (see “Link the project to Sentry” for sentry configuration)
- pip install pipenv
- pipenv shell

1.1.3 Link the project to Sentry

Sentry is a platform that automatically flags errors and project exceptions. It also allows for performance monitoring.

- Create a Sentry account
- Create a project with the platform
- Retrieve the dsn key and embed it in your “.env” file
- Log in to your Sentry account to view the logs retrieved by Sentry

1.1.4 Run the site

- mkdir logs
- python manage.py runserver
- goto <http://localhost:8000> with your browser
- confirm that the site is working and that it is possible to navigate through the different pages

1.2 Application Description

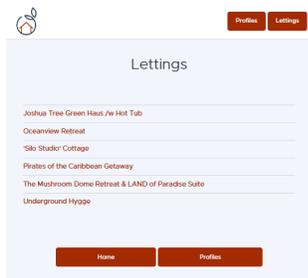
1.2.1 Home page

When you run the application locally with command `python manage.py runserver`, you go to <http://localhost:8000> to access home page.



1.2.2 Lettings application

then you can access the lettings list:



and select one to show the address:

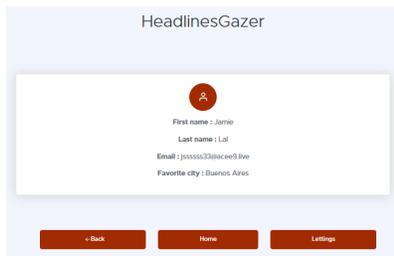


1.2.3 Profiles application

then you can go back home and inspect the profiles list:



and look at a profile data:



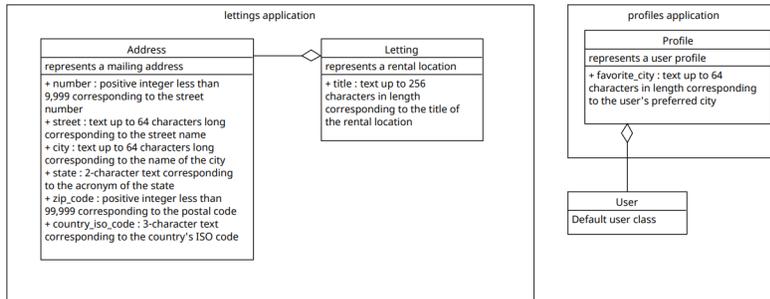
1.3 Database

The data is stored on a SQLite3 database included in the Django project container. Here are some steps to check that the database is working properly.

- Open a sqlite3 shell session
- connect to the database with: `.open oc-lettings-site.sqlite3`
- view tables in the database: `.tables`
- show columns in the profiles table: `pragma table_info(profiles_profile);`
- query the profile table: `select user_id, favorite_city from profiles_profile where favorite_city like 'B%';`
- `.quit` to exit

1.3.1 Models

The project is broken down into 2 applications: lettings and profiles. Here is the description of the models and the links between the different classes.



1.4 Development Process

1.4.1 Run the site locally with Django

- start virtual environment¹
- `python manage.py collectstatic`
- `python manage.py runserver`
- goto <http://localhost:8000> with your browser
- **goto <http://localhost:8000/admin> to access the admin panel**
you can connect with user admin and mot de passe Abc1234!
- goto <http://localhost:8000/sentry-debug/> to generate a `ZeroDivisionError` and verify your Sentry account

1.4.2 Run the site locally via Docker

- create a dockerhub account
- install docker desktop
- retrieve the docker image to run the application locally : `docker pull slb59/lettings`
- make sure the local server is not running
- launch the server : `docker compose -f compose/docker-compose.yml up -d`
- the site should work the same way with the same urls, as if using the local procedure
- To shut down the server without deleting the created resources: `docker compose stop`, and to stop it by destroying all the resources created: `docker compose down`

¹ by setting the `DEBUG` variable in the `.env` file to `true`, you can view the debug-toolbar

1.4.3 quality control

Linting

- activate the virtual environment
- Flake8 is a wrapper around these tools:
 - PyFlakes
 - pycodestyle
 - Ned Batchelder’s McCabe script

```
flake8
```

isort

- **isort is a Python utility / library to sort imports alphabetically, and automatically separated into sections and by type**

```
isort . --check
```

black

- **black is the uncompromising Python code formatter.**

```
black . --check
```

pylint

- **pylint is a static code analyser for Python 2 or 3.**

```
pylint . --recursive=y > logs/pylint.txt
```

then you can check the logs/pylint.txt file

pytest

- **pytest framework makes it easy to write unit tests**

```
pytest
```

You can check the tests coverage with:

```
pytest --cov=. --cov-report=html
```

then check the result in htmlcov.index.html

You can also check the html report logs/pytest-report.html with:

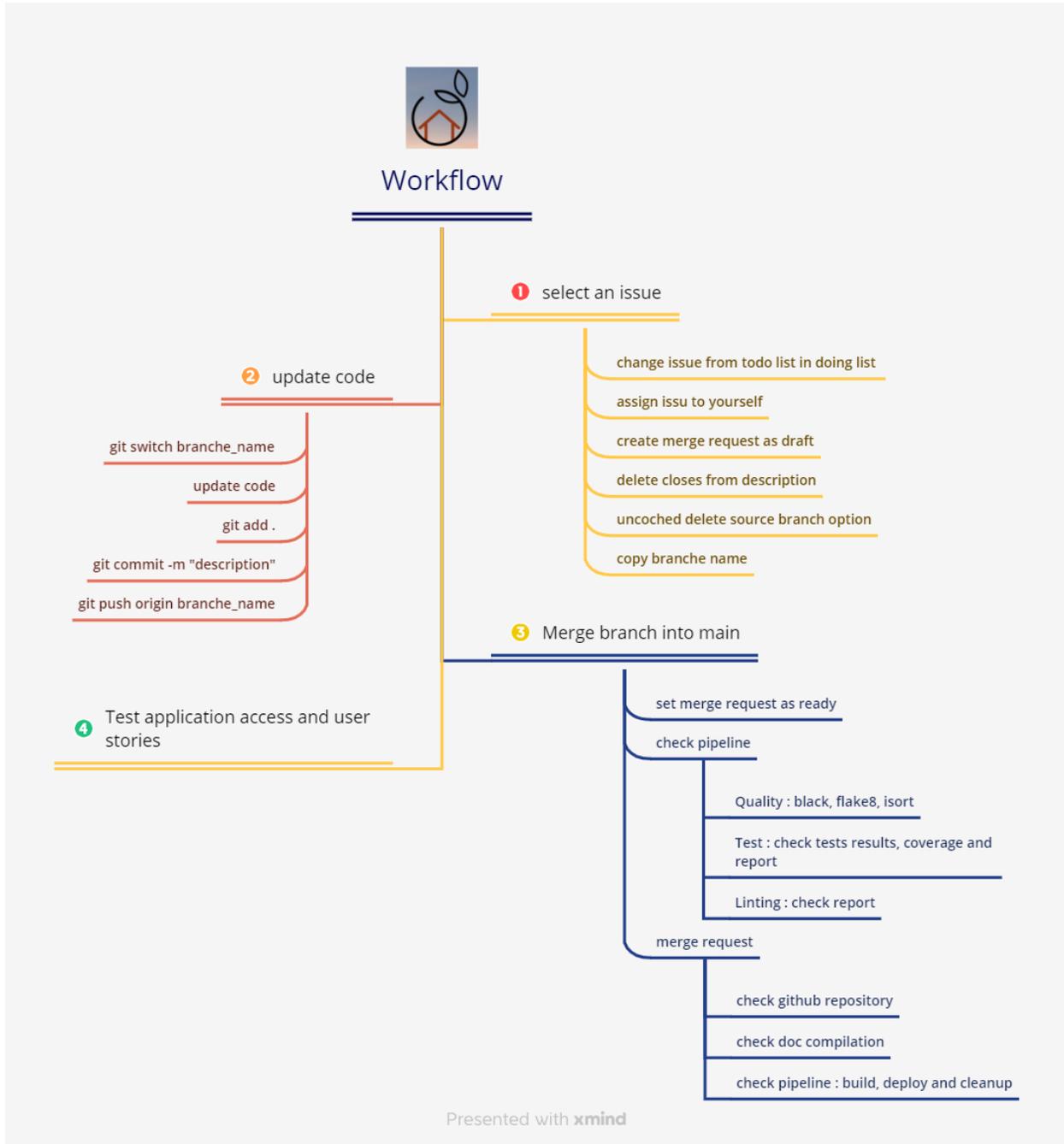
```
pytest --html=logs/pytest-report.html
```

1.5 deployment Process

This project is managed by gitlab : <https://gitlab.com/slb591/lettings>

1.5.1 workflow description

The diagram below describes the steps for modifying and deploying a modification.



1.5.2 versioning

- when a merge request is created, a development branch is generated with the name of the issue
- next the branch is merged in the main branch
- then the main branch can be merged in a stable version

1.5.3 pipeline

- **on commit,**
 - the files are updated in gitlab and github
 - a first check is running for quality
 - then test and linting are executed
 - then the pipeline check if tests are 80% coverage
- **on merge request into main branch,**
 - the latest version of documentation is updating
 - then complete build is generated
 - the deployment is running on docker hub and render
- **on merge request into a stable branch,**
 - the stable version of documentation is updating
 - the deployment is running on docker hub and render

1.6 Indices and tables

- genindex
- modindex
- search