

Sous-Chef Hacknight

Emmanuel Milou

2016-08-25

Mentoring new contributors through contributing their first core commit and growing the community.

Venue

This fourth code sprint, or **hackaton**, on Sous-Chef will take place on **Thursday, August 25th, 2016**.

The venue is [Santropol Roulant](#), in Montréal.

More informations on the [Montreal Python Meetup page](#).

Agenda

18h

Welcome!

18h15

Introduction of the project and the persons involved / discussion, Maison Du Logiciel Libre and Santropol Roulant

18h30-21h30

Coding

Food will be served.

Resources

The project is hosted on GitHub: <https://github.com/savoirfairelinux/sous-chef>

Some useful resources:

- [Coding style guide](#)
- [CI server \(Travis\)](#)
- [Waffle Dashboard](#)
- [Coveralls](#)
- [Creating a Pull Request on GitHub](#)

Communication

Please join [#feastproject](#) on [irc.freenode.net](#).

A [public mailing-list](#) is also available. Feel free to subscribe.

Installation

Please refer to the [installation procedure](#).



Make sure you already have installed the application before arriving.

How to contribute

Contributions to an open-source project can take multiple forms:

- Code
- Documentation
- Tests and bug reporting
- Translations

Code

A pre-requisite is to have Sous-Chef installed on your machine. Issues of all levels will be available.

The installation procedure is based on Docker and is described [on GitHub](#).

Issues

Numerous issues are created and waiting for your love on the [issue board](#).

You may also use [Waffle](#) as a more **Agile-like** dashboard.

Issue trackers

Issues are categorized with the following labels:

- feature
- bug
- i18n (internationalization)
- docs (documentation)
- ux/ui
- tests

Issue categories

Besides those trackers, issues are identified regarding the field of expertise:

- python
- frontend

- database



Sous-Chef project is using [Semantic UI](#) as frontend framework.

Workflow

- Fork the repository and work from there
- Browse Waffle to pick an issue, change the status to Work-in-Progress and assign it to yourself
- Commit and push your changes into your fork
- When you are done, create a [Pull Request](#) in Feast repository

For your Pull-Request to be merged, it should respect the following:



- Tests must pass
- PEP-8 coding style must be respected
- Code coverage must be complete
- The branch can be merged automatically

[TravisCI](#) is configured to run those checks on every Pull-Request. It is recommended you configure your fork to do the same.

Documentation

Documentation is important and can always be improved:

- Installation instructions
- Test procedures
- How to help translating the project
- Updating the [Roadmap](#)

The preferred markdown syntax is [AsciiDoc](#). Markdown is also accepted.

The [project wiki](#) on GitHub could also be used.

Tests and bug reporting

It could be manual testing and reporting. The GitHub issues can be used to report problems.

Writing unit or functional tests is highly encouraged and appreciated. Sous-Chef is using [unittest](#) as test framework.

Translations

Sous-Chef is a bilingual application and needs your talent of translators !

Sous-Chef uses Django multilingual framework, based on Gettext and po files.

Each module has its own set of translation files. To list them all:

```
$ find . -name "*.po"
```

[Poedit](#) can be used to translate those files. Follow the workflow described in the Code section to contribute your translations.



An alternative way to contribute could be to send the po files through emails.

An other way to contribute there is to review the code and make sure all strings are translatable.