Learn-OCaml: How to contribute?

Érik Martin-Dorel

Dept. *Fiabilité des Systèmes et des Logiciels*, Lab. IRIT
Univ. Toulouse III - Paul Sabatier

Wednesday 23 March 2022

Agenda

1. How to contribute code?

2. How to contribute beyond code?

Agenda

1. How to contribute code?

2. How to contribute beyond code?

Environment for learn-ocaml developers

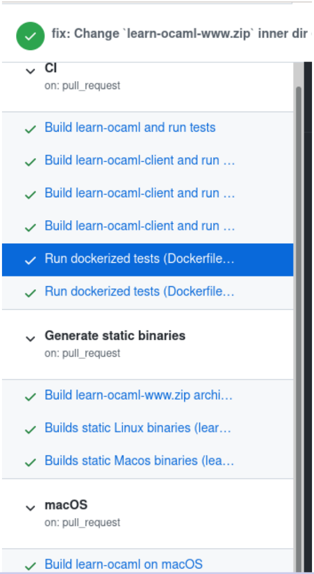
Needs:

- Linux or macOS
- `opam` 2 + `ocaml` 4.12 (currently)
- a "Merlin-aware" OCaml IDE (such as Tuareg+this config)
- Git (+ ideally Magit) + a GitHub account
- Docker Engine + `docker-compose`
  - https://docs.docker.com/engine/install/
  - https://docs.docker.com/compose/install/

Environment for learn-ocaml.el developers

https://gist.github.com/erikmd/35251ac083e7433f3e780f7eb8856782

Continuous Integration



✓ fix: Change `learn-ocaml-www.zip` inner dir

CI
on: pull_request

- ✓ Build learn-ocaml and run tests
- ✓ Build learn-ocaml-client and run ...
- ✓ Build learn-ocaml-client and run ...
- ✓ Build learn-ocaml-client and run ...
- ✓ Run dockerized tests (Dockerfile...
- ✓ Run dockerized tests (Dockerfile...

Generate static binaries
on: pull_request

- ✓ Build learn-ocaml-www.zip archi...
- ✓ Builds static Linux binaries (lear...
- ✓ Builds static Macos binaries (lea...

macOS
on: pull_request

- ✓ Build learn-ocaml on macOS

Learn-OCaml contains "CI tests" to automate the Pull-Request checks

Continuous Deployment

Learn-OCaml actually contains a whole "CI/CD chain" to automate the releases:

- release-please automatically generates the CHANGELOG and computes/updates the version number
- the static binaries are built for Linux and macOS
- a tag and a GitHub release are made, along with the binaries
- three Docker images are pushed to Docker Hub
- and a PR is opened by a bot in the main `opam-repository`

Contributing guidelines

Learn-OCaml has a very permissive open-source license (MIT)

Contributions are very welcome!

- See also CONTRIBUTING.md, in particular:
  - Use Atomic Commits so each commit gathers a single logical change
  - Use Conventional Commits messages (needed by release-please)
- Integration tests are stored in:
  `https://github.com/ocaml-sf/learn-ocaml/tree/master/tests`
- Documentation is stored in:
  `https://github.com/ocaml-sf/learn-ocaml/tree/master/docs`
  and automatically deployed in `https://ocaml-sf.org/learn-ocaml/`

(these links are recalled in so-called "PR templates / issue templates")

Agenda

1 How to contribute code?

2 How to contribute beyond code?

What can we do to contribute to the Learn-OCaml project?

- *Beta-test* `ocamlsf/learn-ocaml:master` before a release
- *Suggest some* quality-of-life improvements
- *Give some feedback about the* roadmap
- *Open issues and PRs to improve the* documentation
- *Make demos of the platform if* other teachers can be interested!
- *Contribute new exercises in* `learn-ocaml-corpus`?
- *Help maintaining* `learn-ocaml-corpus`?
- ?

What can we do to contribute to the Learn-OCaml project?

- *Beta-test* `ocamlsf/learn-ocaml:master` before a release
- *Suggest some* quality-of-life improvements
- *Give some feedback about the* roadmap
- *Open issues and PRs to improve the* documentation
- *Make demos of the platform if* other teachers can be interested!
- *Contribute new exercises in* `learn-ocaml-corpus`?
- *Help maintaining* `learn-ocaml-corpus`?
- ?

What about discussing these topics further at the end of the meeting?