Codefair - Make Biomedical Research Software FAIR Without Breaking a Sweat

Dorian Portillo, BS[1], Sanjay Soundarajan, MS[1], Jacob Clark, BS[1], Bhavesh Patel, PhD[1]

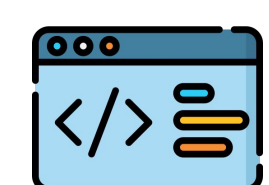[1]FAIR Data Innovations Hub, California Medical Innovations Institute, San Diego, CA, USA

Background

Software created for a research purpose or during the research process are designated as research software. They are developed in different formats and for different applications such as Python code for machine learning model or desktop software to process data.

Making research software Findable, Accessible, Interoperable, and Reusable (FAIR), i.e. optimally reusable, is critical to enable reproducible, transparent research and prevent duplicate effort. This requires several steps, as specified by the FAIR-BioRS guidelines, such as:

- Specify a clear usage license
- Follow best coding practices
- Document software
- Include CITATION.cff and codemeta.json metadata files
- Archive software (e.g., on Zenodo)

Problem

Researchers **lack awareness, training, and time** for making research software FAIR. Most researchers are not formally trained in software development so this could also become **challenging** for them.

Solution: Codefair

We are developing codefair, a **free and open source GitHub app** that acts as a personal assistant for making biomedical research software FAIR. The objective is to minimize developers' time and effort in making their software FAIR so they can focus on the goals of their software.
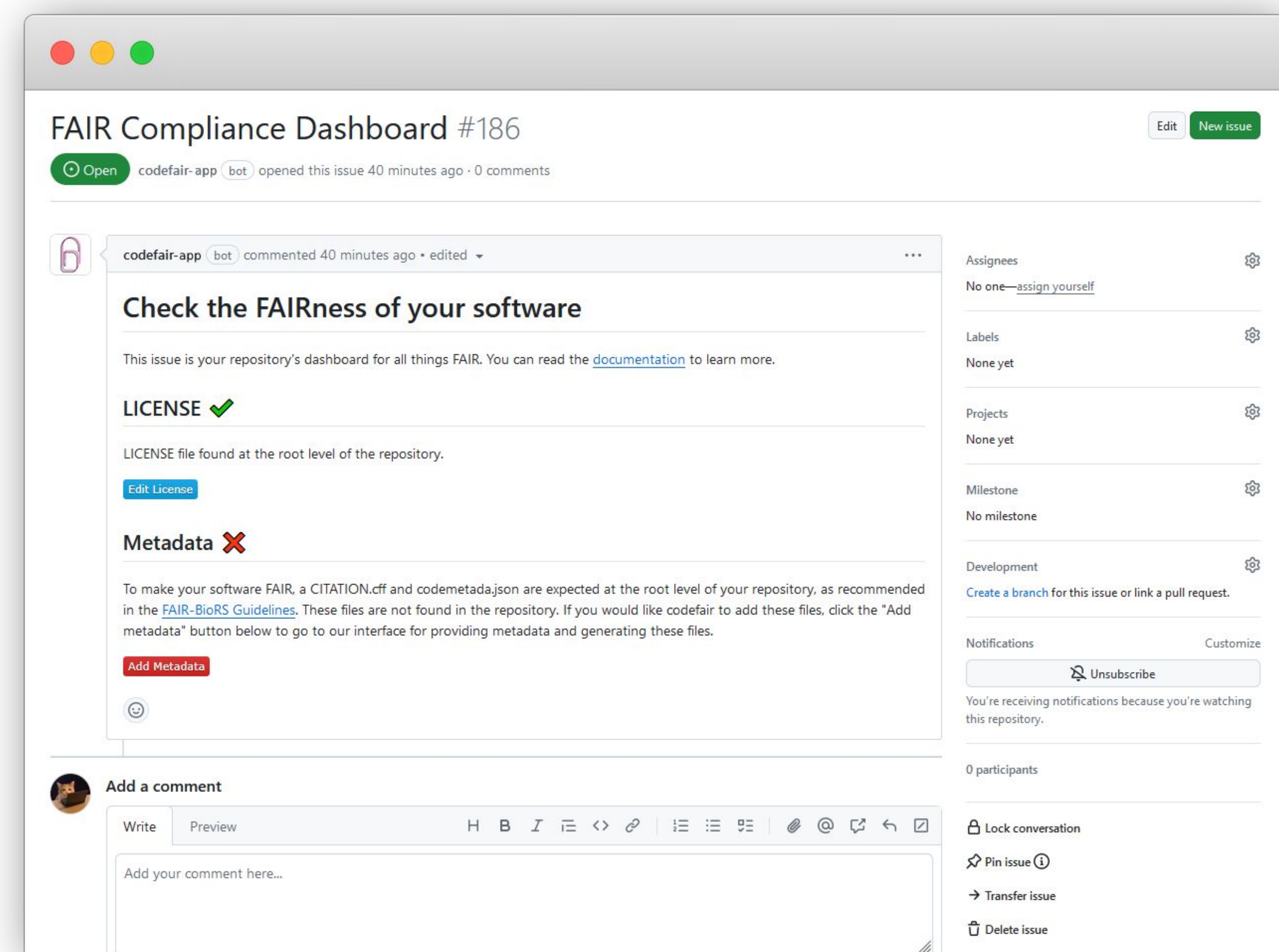

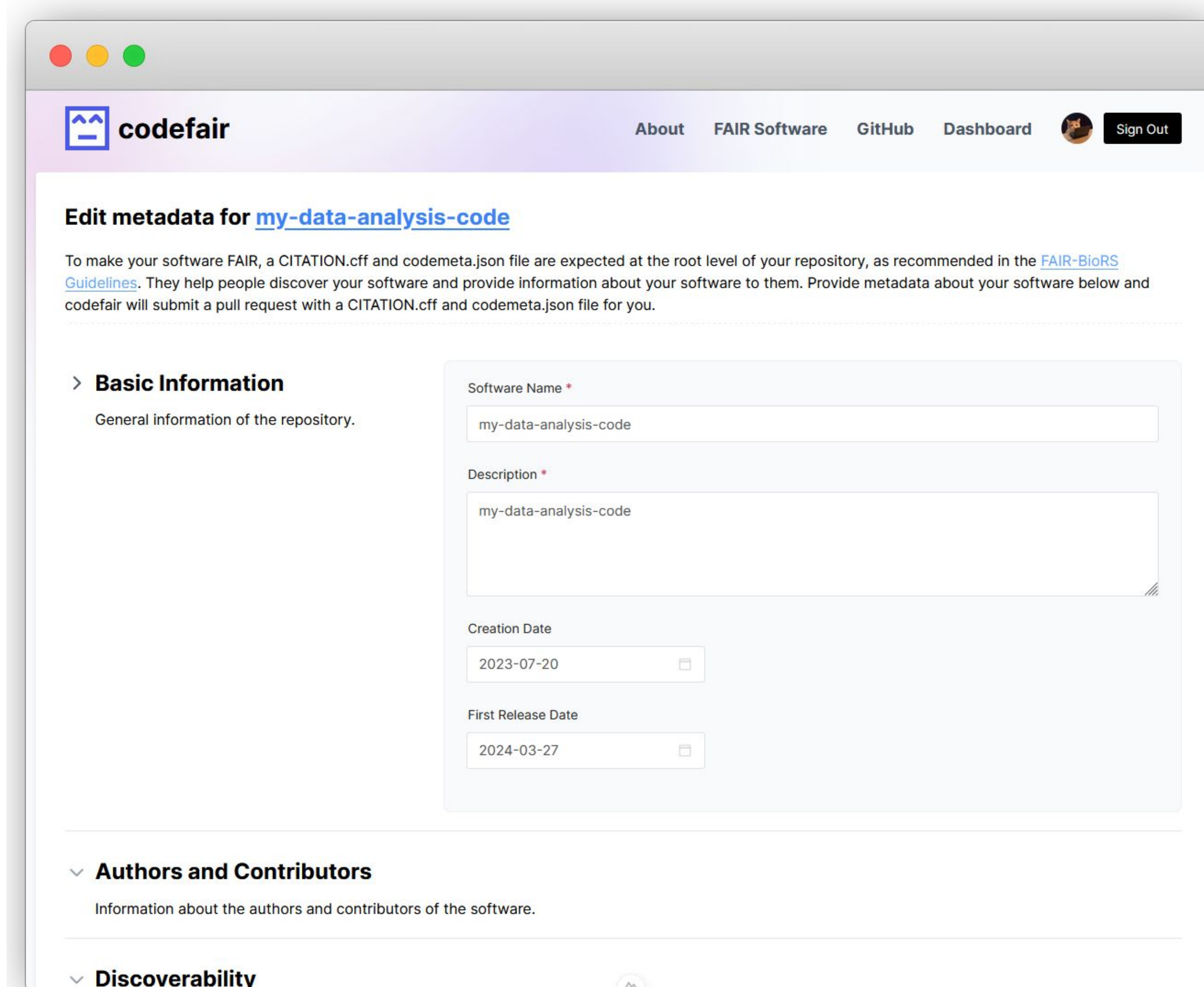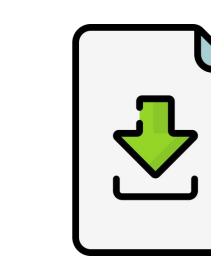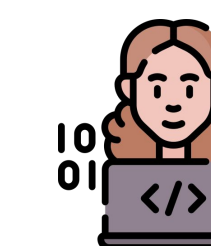**Figure 1**. Screenshot of a codefair GitHub Issue dashboard for a GitHub repository.

Figure 2. The codefair user interface for providing metadata and push CITATION.cff and codemeta.json metadata files to a GitHub repository.
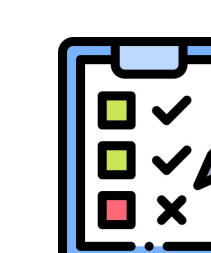
Using Codefair

Codefair is designed to be easy to use:

- Install codefair-app from the GitHub marketplace
- Code as usual
- Address FAIR compliance issues through the codefair issue dashboard and the codefair user interface

By leveraging tools such as Probot and GitHub API, codefair monitors activities on the software repository and communicates with the developers via a GitHub issue "dashboard" that lists issues related to FAIR-compliance (**Figure 1**).

For each issue, there is a link that takes the developer to the codefair user interface (built with Nuxt 3 and Typescript) where they can better understand the issue, address it through an intuitive interface, and submit a pull request automatically to fix the issue (**Figure 2**).

Automation, such as pre-populating metadata and getting license terms, is included to support users further.

Future work

Additional features are being added to provide support for complying with best coding practices, archiving on Zenodo and Software Heritage, registering on bio.tools, and much more to cover all the requirements for making software FAIR.

The impact of codefair can be very broad as it can be extended to other fields outside of biomedical and also provide support for aspects outside of FAIR, such as quality and security.

We believe codefair is an essential tool for enabling software curation at scale and turning FAIR software into reality.

Try codefair today at codefair.io!

Dorian Portillo
dportillo@calmi2.org
fairdataihub.org

Bhavesh Patel
bpatel@calmi2.org
fairdataihub.org

Find this poster and all related references here

bit.ly/codefairBOSC