

Recommended Practices for LF AI Projects on GitHub

- Use the REPOLINTER [tool](#) created by the TODO Group to identify common issues in GitHub repos.
- Secure your GitHub account with two-factor authentication.
- Ensure that every repo includes a LICENSE file.
- Add a README file to your repos welcoming new community members to the project and explaining why the project is useful and how to get started.
- Add a CONTRIBUTING file to your repos explaining to other developers and your community of users how to contribute to the project. At a high level, the file would explain what types of contributions are needed and how the process works.
- Add CODEOWNERS file to define individuals or teams that are responsible for code in a repository.
- Add a CODE_OF_CONDUCT file that sets the ground rules for participants' behavior associated and helps to facilitate a friendly, welcoming environment. While not every project has a CODE_OF_CONDUCT file, its presence signals that this is a welcoming project to contribute to, and defines standards for how to engage with the project's community. You are welcome to use the Linux Foundation's [Code of Conduct](#) if project specific CoC does not exist.
- Provide documentation on the release methodology, cadence, criteria, etc.
- Document your project governance and make it available on the project's repo.
- Add a SUPPORT file to let users and developers know about ways to get help with your project. You can either add in this file how and where security issues are handled, or put it at the top level readme for the project, or alternatively refer to security documentation.

- Archive inactive repos to flag to your users and other developers that you're not maintaining them.
- Setup issue template and pull request templates that help you customize and standardize the information you'd like contributors to include when they open issues and pull requests in your repository.
- Achieve and maintain a [Core Infrastructure Initiative Best Practices Badge](#) passing badge for your project.
- Identify who on the project will be handling security issues (could be a team) and set up a separate email account. Consider having the project become a CNA (CVE Numbering Authority).
- Include an [SPDX short-form identifier](#) in a comment at the top of each file in the repo, wherever reasonably possible.
- Depending on whether your project uses the DCO and/or CLAs:
 - DCO: Include a copy of or reference to the [Developer Certificate of Origin \(DCO\)](#) in your CONTRIBUTING file. Set up a bot to enforce a "Signed-off-by:" tag in each commit (<https://github.com/apps/dco>)
 - CLAs: Use the Linux Foundation's EasyCLA tool (<https://lfcla.com/>) to enforce signed CLAs before contributions are accepted.
 - In either case: Configure the Github repo settings so that administrators are not able to bypass the DCO or CLA checks.
- Use English as the default universal language for anything you publish on GitHub. You can support a second language but English should be the primary language of communication towards a universal audience.

These practices will help you improve your GitHub presence in an effort to help you attract more users and developers to your project, secure your account, be precise about licensing, and maintain good housekeeping.