# Contributing

All kinds of contributions to git-remote-dropbox are greatly appreciated. For

someone unfamiliar with the code base, the easiest way to contribute is

probably to submit a [feature request](#feature-requests) or [bug

report](#bug-reports). If you want to dive into the source code, you can submit

a [patch](#patches) as well, either working on your own ideas or [existing

issues][issues].

## Feature Requests

Do you have an idea for an improvement to git-remote-dropbox? Please [submit a

feature request][issue]. It's great to hear about new ideas.

If you are inclined to do so, you're welcome to [fork][fork]

git-remote-dropbox, work on implementing the feature yourself, and submit a

patch. In this case, it's \*highly recommended\* that you first [open an

issue][issue] describing your enhancement to get early feedback on what you're

doing. This will help avoid wasted efforts and ensure that your work is

incorporated into the code base.

## Bug Reports

Did something go wrong with git-remote-dropbox? Sorry about that! Bug reports

are greatly appreciated!

When you [submit a bug report][issue], please include relevant information such

as git-remote-dropbox version, Git version, operating system, error messages,

and steps to reproduce the bug. The more details you can include, the easier it

is to find and fix the bug.

## Patches

Want to hack on git-remote-dropbox? Awesome!

You may find the documentation in [DESIGN.md][design] and the source code

helpful.

If there are [open issues][issues], you're more than welcome to work on those -

this is probably the best way to contribute code to git-remote-dropbox. If you

have your own ideas, that's great too! In that case, before working on

substantial changes to the code base, it is \*highly recommended\* that you first

[open an issue][issue] describing what you intend to work on.

\*\*Patches are generally submitted as pull requests.\*\* Patches are also

[accepted over email][email].

Any changes to the code base should follow the style and coding conventions

used in the rest of the project. The version history should be clean, and

commit messages should be descriptive and [properly

formatted][commit-messages].

---

If you have any questions about anything, feel free to [ask][email]!

[design]: DESIGN.md

[issue]: https://github.com/anishathalye/git-remote-dropbox/issues/new

[issues]: https://github.com/anishathalye/git-remote-dropbox/issues

[fork]: https://github.com/anishathalye/git-remote-dropbox/fork

[email]: mailto:me@anishathalye.com

[gitremote-helpers]: https://www.kernel.org/pub/software/scm/git/docs/gitremote-helpers.html

[commit-messages]: http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html