


Welcome to GitHub docs contributing guide

Thank you for investing your time in contributing to our project! Any contribution you make will be reflected on docs.github.com ✨.

Read our [Code of Conduct](#) to keep our community approachable and respectable.

In this guide you will get an overview of the contribution workflow from opening an issue, creating a PR, reviewing, and merging the PR.

Use the table of contents icon  on the top left corner of this document to get to a specific section of this guide quickly.

New contributor guide

To get an overview of the project, read the [README](#). Here are some resources to help you get started with open source contributions:

- [Finding ways to contribute to open source on GitHub](#)
- [Set up Git](#)
- [GitHub flow](#)
- [Collaborating with pull requests](#)

Getting started

To navigate our codebase with confidence, see [the introduction to working in the docs repository](#) 🧑‍🔬. For more information on how we write our markdown files, see [the GitHub Markdown reference](#).

Check to see what [types of contributions](#) we accept before making changes. Some of them don't even require writing a single line of code ✨.

Issues

Create a new issue

If you spot a problem with the docs, [search if an issue already exists](#). If a related issue doesn't exist, you can open a new issue using a relevant [issue form](#).

Solve an issue

Scan through our [existing issues](#) to find one that interests you. You can narrow down the search using [labels](#) as filters. See [Labels](#) for more information. As a general rule, we don't assign issues to anyone. If you find an issue to work on, you are welcome to open a PR with a fix.

Make Changes

Make changes in the UI

Click Make a contribution at the bottom of any docs page to make small changes such as a typo, sentence fix, or a broken link. This takes you to the `.md` file where you can make your changes and [create a pull request](#) for a review.

Help us make these docs great!

All GitHub docs are open source. See something that's wrong or unclear?
Submit a pull request.

 **Make a contribution**

Or, [learn how to contribute](#).

Make changes in a codespace

For more information about using a codespace for working on GitHub documentation, see "[Working in a codespace](#)."

Make changes locally

1. Fork the repository.
 - Using GitHub Desktop:
 - [Getting started with GitHub Desktop](#) will guide you through setting up Desktop.
 - Once Desktop is set up, you can use it to [fork the repo](#)!
 - Using the command line:
 - [Fork the repo](#) so that you can make your changes without affecting the original project until you're ready to merge them.
2. Install or update to Node.js, at the version specified in `.node-version`. For more information, see [the development guide](#).
3. Create a working branch and start with your changes!

Commit your update

Commit the changes once you are happy with them. Don't forget to [self-review](#) to speed up the review process ⚡.

Pull Request

When you're finished with the changes, create a pull request, also known as a PR.

- Fill the "Ready for review" template so that we can review your PR. This template helps reviewers understand your changes as well as the purpose of your pull request.
- Don't forget to [link PR to issue](#) if you are solving one.
- Enable the checkbox to [allow maintainer edits](#) so the branch can be updated for a merge. Once you submit your PR, a Docs team member will review your proposal. We may ask questions or request additional information.
- We may ask for changes to be made before a PR can be merged, either using [suggested changes](#) or pull request comments. You can apply suggested changes directly through the UI. You can make any other changes in your fork, then commit them to your branch.
- As you update your PR and apply changes, mark each conversation as [resolved](#).
- If you run into any merge issues, checkout this [git tutorial](#) to help you resolve merge conflicts and other issues.

Your PR is merged!

Congratulations 🎉🎉 The GitHub team thanks you ✨.

Once your PR is merged, your contributions will be publicly visible on the [GitHub docs](#).

Now that you are part of the GitHub docs community, see how else you can [contribute to the docs](#).

Windows

This site can be developed on Windows, however a few potential gotchas need to be kept in mind:

1. Regular Expressions: Windows uses `\r\n` for line endings, while Unix-based systems use `\n`. Therefore, when working on Regular Expressions, use `\r?\n` instead of `\n` in order to support both environments. The Node.js `os.EOL` property can be used to get an OS-specific end-of-line marker.
2. Paths: Windows systems use `\` for the path separator, which would be returned by `path.join` and others. You could use `path.posix`, `path.posix.join` etc and the [slash](#) module, if you need forward slashes - like for constructing URLs - or ensure your code works with either.
3. Bash: Not every Windows developer has a terminal that fully supports Bash, so it's generally preferred to write [scripts](#) in JavaScript instead of Bash.
4. Filename too long error: There is a 260 character limit for a filename when Git is compiled with `msys`. While the suggestions below are not guaranteed to work and could cause other issues, a few workarounds include:
 - Update Git configuration: `git config --system core.longpaths true`
 - Consider using a different Git client on Windows