# Contributing

Contributions to this project are [released](https://help.github.com/articles/github-terms-of-service/#6-contributions-under-repository-license) to the public under the [project's open source license](LICENSE.md).

Everyone is welcome to contribute to Hubot. Contributing doesn’t just mean submitting pull requests—there are many different ways for you to get involved, including answering questions in [chat](https://hubot-slackin.herokuapp.com/), reporting or triaging [issues](https://github.com/github/hubot/issues), and participating in the [Hubot Evolution](https://github.com/hubotio/evolution) process.

No matter how you want to get involved, we ask that you first learn what’s expected of anyone who participates in the project by reading the [Contributor Covenant Code of Conduct](http://contributor-covenant.org). By participating, you are expected to uphold this code.

We love pull requests. Here's a quick guide:

1. If you're adding a new feature or changing user-facing APIs, check out the [Hubot Evolution](https://github.com/hubotio/evolution) process.

1. Check for [existing issues](https://github.com/github/hubot/issues) for duplicates and confirm that it hasn't been fixed already in the [master branch](https://github.com/github/hubot/commits/master)

1. Fork the repo, and clone it locally

1. `npm link` to make your cloned repo available to npm

1. Follow [Getting Started](docs/index.md) to generate a testbot

1. `npm link hubot` in your newly created bot to use your hubot fork

1. Create a new branch for your contribution

1. Add [tests](test/) (run with `npm test`)

1. Push to your fork and submit a pull request

At this point you're waiting on us. We like to at least comment on, if not

accept, pull requests within a few days. We may suggest some changes or improvements or alternatives.

Some things that will increase the chance that your pull request is accepted:

\* Make sure the tests pass

\* Update the documentation: code comments, example code, guides. Basically,

update everything affected by your contribution.

\* Include any information that would be relevant to reproducing bugs, use cases for new features, etc.

\* Discuss the impact on existing [hubot installs](docs/index.md), [hubot adapters](docs/adapters.md), and [hubot scripts](docs/scripting.md) (e.g. backwards compatibility)

\* If the change does break compatibility, how can it be updated to become backwards compatible, while directing users to the new way of doing things?

\* Your commits are associated with your GitHub user: https://help.github.com/articles/why-are-my-commits-linked-to-the-wrong-user/

\* Make pull requests against a feature branch,

\* Follow our commit message conventions:

\* use imperative, present tense: “change” not “changed” nor “changes”

\* Commit test files with `test: …` or `test(scope): …` prefix.

\* Commit bug fixes with `fix: …` or `fix(scope): …` prefix

\* Commit features with `feat: …` or `feat(scope): …` prefix

\* Commit breaking changes by adding `BREAKING CHANGE:` in the commit body.

The commit subject does not matter. A commit can have multiple `BREAKING CHANGE:`

sections

\* Commit changes to `package.json`, `.gitignore` and other meta files with

`chore(filenamewithout.ext): …`

\* Commit changes to README files or comments with `docs(README): …`

\* Cody style changes with `style: standard`

\* see [Angular’s Commit Message Conventions](https://gist.github.com/stephenparish/9941e89d80e2bc58a153)

for a full list of recommendations.

# Stale issue and pull request policy

Issues and pull requests have a shelf life and sometimes they are no longer relevant. All issues and pull requests that have not had any activity for 90 days will be marked as `stale`. Simply leave a comment with information about why it may still be relevant to keep it open. If no activity occurs in the next 7 days, it will be automatically closed.

The goal of this process is to keep the list of open issues and pull requests focused on work that is actionable and important for the maintainers and the community.

# Pull Request Reviews & releasing

Releasing `hubot` is fully automated using [semantic-release](https://github.com/semantic-release/semantic-release). Once merged into the `master` branch, `semantic-release` will automatically release a new version based on the commit messages of the pull request. For it to work correctly, make sure that the correct commit message conventions have been used. The ones relevant are

\* `fix: …` will bump the fix version, e.g. 1.2.3 ? 1.2.4

\* `feat: …` will bump the feature version, e.g. 1.2.3 ? 1.3.0

\* `BREAKING CHANGE: …` in the commit body will bump the breaking change version, e.g. 1.2.3 ? 2.0.0