# Introduction

First, thank you for considering contributing to Node Redis! It's people like you that make the open source community such a great community! ?

We welcome any type of contribution, not just code. You can help with;

- \*\*QA\*\*: file bug reports, the more details you can give the better (e.g. platform versions, screenshots sdk versions & logs)

- \*\*Docs\*\*: improve reference coverage, add more examples, fix typos or anything else you can spot.

- \*\*Code\*\*: take a look at the open issues and help triage them.

- \*\*Donations\*\*: we welcome financial contributions in full transparency on our [open collective](https://opencollective.com/node-redis).

---

## Project Guidelines

As maintainers of this project, we want to ensure that the project lives and continues to grow. Not blocked by any

singular person's time.

One of the simplest ways of doing this is by encouraging a larger set of shallow contributors. Through this we hope to

mitigate the problems of a project that needs updates but there is no-one who has the power to do so.

### Continuous Deployment

<!-- TODO(Salakar) -->

Coming soon.

### How can we help you get comfortable contributing?

It is normal for a first pull request to be a potential fix for a problem but moving on from there to helping the

project's direction can be difficult.

We try to help contributors cross that barrier by offering good first step issues (labelled `good-first-issue`). These

issues can be fixed without feeling like you are stepping on toes. Generally, these should be non-critical issues that

are well defined. They will be purposely avoided by mature contributors to the project, to make space for others.

Additionally issues labelled `needs-triage` or `help-wanted` can also be picked up, these may not necessarily require

code changes but rather help with debugging and finding the cause of the issue whether it's a bug or a users incorrect

setup of the library or project.

We aim to keep all project discussion inside GitHub issues. This is to make sure valuable discussion is accessible via

search. If you have questions about how to use the library, or how the project is running - GitHub issues are the goto

tool for this project.

### Our expectations on you as a contributor

You shouldn't feel bad for not contributing to open source. We want contributors like yourself to provide ideas, keep

the ship shipping and to take some of the load from others. It is non-obligatory; we’re here to get things done in an

enjoyable way. :trophy:

We only ask that you follow the conduct guidelines set out in our [Code of Conduct](/CODE\_OF\_CONDUCT.md) throughout your

contribution journey.

### What about if you have problems that cannot be discussed in public?

You can reach out to us directly via email (`redis[AT]invertase.io`) or direct message us on

[Twitter](https://twitter.com/NodeRedis) if you'd like to discuss something privately.

#### Project Maintainers

- Mike Diarmid ([Salakar](https://github.com/Salakar)) @ [Invertase](https://github.com/invertase)

- Twitter: [@mikediarmid](https://twitter.com/mikediarmid)

- Elliot Hesp ([Ehesp](https://github.com/Ehesp)) @ [Invertase](https://github.com/invertase)

- Twitter: [@elliothesp](https://twitter.com/elliothesp)

- Ruben Bridgewater ([BridgeAR](https://github.com/BridgeAR))

- Twitter: [@BridgeAR](https://twitter.com/BridgeAR)

Huge thanks to the original author of Node Redis, [Matthew Ranney](https://github.com/mranney) and also to

[Ruben Bridgewater](https://github.com/BridgeAR) for handing over this project over to new maintainers so it could be

continuously maintained.

---

## Code Guidelines

### Your First Contribution

Working on your first Pull Request? You can learn how from this \_free\_ series,

[How to Contribute to an Open Source Project on GitHub](https://egghead.io/series/how-to-contribute-to-an-open-source-project-on-github).

### Testing Code

Node Redis has a full test suite with coverage setup.

To run the tests use the `npm test` command. To check detailed coverage locally run the `npm run coverage` command after

testing and open the generated `./coverage/index.html` in your browser.

### Submitting code for review

The bigger the pull request, the longer it will take to review and merge. Where possible try to break down large pull

requests into smaller chunks that are easier to review and merge. It is also always helpful to have some context for

your pull request. What was the purpose? Why does it matter to you? What problem are you trying to solve? Tag in any linked issues.

To aid review we also ask that you fill out the pull request template as much as possible.

> Use a `draft` pull request if your pull request is not complete or ready for review.

### Code review process

Pull Requests to the protected branches require two or more peer-review approvals and passing status checks to be able

to be merged.

When reviewing a Pull Request please check the following steps on top of the existing automated checks:

- Does the it provide or update the docs if docs changes are required?

- Have the tests been updated or new tests been added to test any newly implemented or changed functionality?

- Is the testing coverage ok and not worse than previously?