# Contributing to Reakit

We would love for you to contribute to Reakit and help make it even better. By contributing to Reakit, you agree to abide by the [code of conduct](https://github.com/reakit/reakit/blob/master/CODE\_OF\_CONDUCT.md)

- [Ownership](#ownership)

- [Why do we give out push access?](#why-do-we-give-out-push-access)

- [How can we help you get comfortable contributing?](#how-can-we-help-you-get-comfortable-contributing)

- [Our expectations on you as a contributor](#our-expectations-on-you-as-a-contributor)

- [What about if you have problems that cannot be discussed in a public issue?](#what-about-if-you-have-problems-that-cannot-be-discussed-in-a-public-issue)

- [How to start contributing?](#how-to-start-contributing)

- [Contributing with code](#contributing-with-code)

- [Scripts](#scripts)

## Ownership

\*\*If your Pull Request is merged, regardless of content, you're eligible for push access to the organization on GitHub\*\*. This is checked for on pull request merges and an invite is sent automatically, thanks to [Aeryn](https://github.com/Moya/Aeryn/).

Offhand, it is easy to imagine that this would make code quality suffer, but in reality it offers fresh perspectives to the codebase and encourages ownership from people who are depending on the project. If you are building a project that relies on this codebase, then you probably have the skills to improve it and offer valuable feedback.

Everyone comes in with their own perspective on what a project could/should look like, and encouraging discussion can help expose good ideas sooner.

### Why do we give out push access?

It can be overwhelming to be offered the chance to wipe the source code for a project. Do not worry, we do not let you push to `master`. All code is peer-reviewed and needs an approval of a code owner. Also, we have the convention that someone from the core team should merge non-trivial pull requests.

\*\*As an organization contributor, you can merge other people's pull requests, or other contributors can merge yours.\*\* You likely won't be assigned a pull request, but you're welcome to jump in and take a code review on topics that interest you.

This project is not continuously deployed, there is space for debate after review too. This means there's always a chance to revert, or to make an amending pull request. If it feels right, merge.

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

It's normal for a first pull request to be a potential fix for a problem, and 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. These issues can be fixed without feeling like you're stepping on toes. Ideally, these are non-critical issues that are well defined. They will be purposely avoided by mature contributors to the project, to make space for others.

We aim to keep all technical discussions inside GitHub issues, and all other discussions in our [Spectrum community](https://spectrum.chat/reakit). This is to make sure valuable discussions are public and discoverable via search engines. If you have questions about a specific PR, want to discuss a new API idea etc GitHub issues are the right place. If you have questions about how to use the library, or how the project is running - the [Spectrum community](https://spectrum.chat/reakit) is the place to go.

### Our expectations on you as a contributor

To quote [@alloy](https://github.com/alloy) from [this issue](https://github.com/Moya/Moya/issues/135):

> Don't ever feel bad for not contributing to open source.

We want contributors 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:

The fact that you'll have push access will allow you to:

\* Avoid having to fork the project if you want to submit other pull requests as you'll be able to create branches directly on the project.

\* Help triage issues, merge pull requests.

\* Pick up the project if other maintainers move their focus elsewhere.

It's up to you to use those superpowers or not though ?

If someone submits a pull request that's not perfect, and you are reviewing, it's better to think about the PR's motivation rather than the specific implementation. Having braces on the wrong line should not be a blocker. Though we do want to keep test coverage high, we will work with you to figure that out together.

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

You can send an email to hazdiego@gmail.com.

## How to start contributing?

Helping people on issues and writing docs are definitely the best ways to get started on any open source project.

Not all of us are native English speakers, so it's natural to find some grammar and writing errors. \*\*If you find any, don't hesitate to submit a PR with a correction\*\*. It'll be very much appreciated.

Navigating through our [website](https://reakit.io), if you find anything that could be explained in a better way, just click on the `Edit this page` buttons to go straight to the markdown file and edit it directly on the GitHub interface.

<p align="center">

<img src="https://user-images.githubusercontent.com/3068563/42471092-e93d149c-8392-11e8-8fa9-ef93715de6f1.png" width="350">

</p>

## Contributing with code

If you're a beginner, it'll be a pleasure to help you contribute. You can start by reading [the beginner's guide to contributing to a GitHub project](https://akrabat.com/the-beginners-guide-to-contributing-to-a-github-project/).

This repository is a [monorepo](https://github.com/babel/babel/blob/master/doc/design/monorepo.md) managed by [Lerna](https://github.com/lerna/lerna) and makes use of [yarn workspaces](https://yarnpkg.com/lang/en/docs/cli/workspaces/).

If you haven't already done so, [install yarn](https://yarnpkg.com/en/docs/install) and run these commands:

```sh

git clone https://github.com/reakit/reakit

cd reakit

yarn

yarn website

```

### Branches

`master` branch contains current stable version of `reakit` packages and it is publishing as main `latest` [dist-tag](https://docs.npmjs.com/cli/dist-tag) to npm. PRs with bug fixes could be targeted directly to the `master` branch.

PRs that introduce new features/components should be targeted to the `next` branch. Newly added components and APIs should be prefixed with `unstable\_` to indicate its experimental status.

### Scripts

- `yarn test` runs tests for all packages.

- `yarn coverage` runs tests and open the coverage report on the browser.

- `yarn type-check` checks typescript types.

- `yarn lint` runs `eslint`.

- `yarn lint --fix` runs `eslint` and automatically fix issues.

- `yarn build` builds all `reakit\*` packages.

- `yarn build:fast` builds all `reakit\*` packages ignoring UMD builds.

- `yarn docs` builds docs into README files.

- `yarn website` starts a development server of the website.

- `yarn website:build` builds production files of the website (requires `yarn build` to be executed first).

- `yarn website:serve` starts a production server for the website (requires `yarn website:build` to be executed first).

- `yarn test:all` runs `lint`, `type-check`, `build`, `test`, `website:build` and `website:serve` in sequence.