<p align="center">

<a href="https://sentry.io" target="\_blank" align="center">

<img src="https://sentry-brand.storage.googleapis.com/sentry-logo-black.png" width="280">

</a>

<br />

</p>

# Contributing

We welcome suggested improvements and bug fixes to the `@sentry/\*` family of packages, in the form of pull requests on [`GitHub`](https://github.com/getsentry/sentry-javascript). The guide below will help you get started, but if you have further questions, please feel free to reach out on [Discord](https://discord.gg/Ww9hbqr).

## Setting up an Environment

To run the test suite and our code linter, node.js and yarn are required.

[`node` download](https://nodejs.org/download)

[`yarn` download](https://yarnpkg.com/en/docs/install)

`sentry-javascript` is a monorepo containing several packages, and we use `lerna` to manage them. To get started, install all dependencies, use `lerna` to bootstrap the workspace, and then perform an initial build, so TypeScript can read all of the linked type definitions.

```

$ yarn

$ yarn lerna bootstrap

$ yarn build

```

With that, the repo is fully set up and you are ready to run all commands.

## Building Packages

Since we are using [`TypeScript`](https://www.typescriptlang.org/), you need to transpile the code to JavaScript to be

able to use it. Every package has a `build` script which takes care of everything. You can also run `build` on all of the

packages at once by calling `yarn build` in the project root.

## Adding Tests

\*\*Any nontrivial fixes/features should include tests.\*\* You'll find a `test` folder in each package.

Note that \_for the `browser` package only\_, if you add a new file to the [integration test suite](https://github.com/getsentry/sentry-javascript/tree/master/packages/browser/test/integration/suites), you also need to add it to [the list in `shell.js`](https://github.com/getsentry/sentry-javascript/blob/b74e199254147fd984e7bb1ea24193aee70afa74/packages/browser/test/integration/suites/shell.js#L25) as well. Adding tests to existing files will work out of the box in all packages.

## Running Tests

Running tests works the same way as building - running `yarn test` at the project root will run tests for all packages, and running `yarn test` in a specific package will run tests for that package. There are also commands to run subsets of the tests in each location. Check out the `scripts` entry of the corresponding `package.json` for details.

Note: you must run `yarn build` before `yarn test` will work.

## Linting

Similar to building and testing, linting can be done in the project root or in individual packages by calling `yarn lint`.

Note: you must run `yarn build` before `yarn lint` will work.

## Final Notes

When contributing to the codebase, please make note of the following:

- Non-trivial PRs will not be accepted without tests (see above).

- Please do not bump version numbers yourself.

- [`raven-js`](https://github.com/getsentry/sentry-javascript/tree/3.x/packages/raven-js) and [`raven-node`](https://github.com/getsentry/sentry-javascript/tree/3.x/packages/raven-node) are deprecated, and only bug and security fix PRs will be accepted targeting the [3.x branch](https://github.com/getsentry/sentry-javascript/tree/3.x). Any new features and improvements should be to our new SDKs (`browser` and `node`) and the packages (`core`, `hub`, `integrations`, and the like) which support them.