# Contributing

[![lerna](https://img.shields.io/badge/maintained%20with-lerna-cc00ff.svg)](https://lerna.js.org/)

## Getting Started

Install required global dependencies:

```bash

npm install -g yarn

```

Check out the code and go into the pollyjs directory:

```bash

git clone https://github.com/netflix/pollyjs.git

cd pollyjs

```

Install the dependencies and bootstrap the monorepo:

```bash

yarn

```

The code for individual packages of this monorepo are in `packages/@pollyjs/\*`.

Within any of the packages in this monorepo you'll generally use the npm

package scripts to manage the project, E.g. `yarn run test` or

`yarn run lint`. Run `yarn run` for a list of available commands.

## Running Tests

### Full Suite

To run the full test suite, from the root directory run:

```bash

yarn test

```

This will perform a full bootstrap, clean and build on all of the sub-packages

and test suite, stand up the node server, run the test suite and then terminate.

### Running only changed tests

While developing, it may become cumbersome to run the entire suite after each change.

In one terminal tab, run the following:

```bash

yarn watch

```

This will build all of the sub-packages and test suite, watch for any changes, and

perform incremental builds when the suite or packages are changed.

Wait until the build settles (i.e. build output stops scrolling). Then, in another tab:

```bash

yarn test:watch

```

This will launch an interactive test runner (`testem`), which will automatically detect

and re-run changed tests. To manually re-run the suite, hit `enter`. To exit, hit `q` and

then `ctrl-c` your watch process. For more information, look at the

[testem docs](https://github.com/testem/testem).

## Running Node Tests with Chrome Inspector

To run the node test suite with node inspector support, run from the root directory:

```bash

yarn test:ci -l Node:debug

```

Next, attach Chrome to the running process by visiting [chrome://inspect/#devices](chrome://inspect/#devices)

## Running Docs

All the documentation can be found in the root level `docs` directory. Running

the following command will stand up the docs server which will watch for

changes.

```bash

yarn docs:serve

```

## Conventional Commits

Lerna depends on the use of the [Conventional Commits Specification](https://conventionalcommits.org/)

to determine the version bump and generate CHANGELOG.md files. Make sure your

commits and the title of your PRs follow the spec. A pre-commit hook and CI test

have been added to further enforce this requirement.

## Tips for Getting Your Pull Request Accepted

1. Make sure all new features are tested and the tests pass.

2. Bug fixes must include a test case demonstrating the error that it fixes.

3. Open an issue first and seek advice for your change before submitting

a pull request. Large features which have never been discussed are

unlikely to be accepted. \*\*You have been warned.\*\*