# CONTRIBUTING

Contributions are always welcome, no matter how large or small. Before contributing,

please read the [code of conduct](CODE\_OF\_CONDUCT.md).

For details on contributing to documentation, see [Website Directory Readme](https://github.com/netlify/netlify-cms/blob/master/website/README.md).

## Setup

> Install Node.js (LTS) on your system: [https://nodejs.org/](https://nodejs.org/)

> Install yarn on your system: [https://yarnpkg.com/en/docs/install](https://yarnpkg.com/en/docs/install)

### Install dependencies

> Only required on the first run, subsequent runs can use `yarn start` to both

> bootstrap and run the development server.

```sh

$ git clone https://github.com/netlify/netlify-cms

$ cd netlify-cms

$ yarn

$ yarn bootstrap

```

### Run locally

```sh

yarn start

```

## Available scripts

### bootstrap

Bootstraps the monorepo.

```sh

yarn bootstrap

```

### watch

Watches all CMS packages and transpiles them on change.

```sh

yarn watch

```

### start

Starts the development server. This task runs both the `bootstrap` and `watch` scripts.

```sh

yarn start

```

### clean

Removes all of the CMS package `dist` directories.

```sh

yarn clean

```

### reset

Runs the `clean` script and removes all the `node\_modules` from the CMS packages.

```sh

yarn reset

```

### build

Runs the `clean` script and builds the CMS packages.

```sh

yarn build

```

### build-preview

Runs the `build` and `build-preview` scripts in each package and serves the resulting build locally.

```sh

yarn build-preview

```

### test

Runs linting and Jest tests.

```sh

yarn test

```

### test:all

Runs linting, Jest, and Cypress tests.

```sh

yarn test:all

```

### test:e2e

Runs Cypress e2e tests.

```sh

yarn test:e2e

```

### test:e2e:dev

Runs Cypress e2e tests on watch mode with an open instance of Chrome.

```sh

yarn test:e2e:dev

```

### format

Formats code and docs according to our style guidelines.

```sh

yarn format

```

## Pull Requests

We actively welcome your pull requests!

If you need help with Git or our workflow, please ask in our [community chat](https://netlifycms.org/chat). We want your contributions even if you're just learning Git. Our maintainers are happy to help!

Netlify CMS uses the [Forking Workflow](https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow) + [Feature Branches](https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow). Additionally, PR's should be [rebased](https://www.atlassian.com/git/tutorials/merging-vs-rebasing) on master when opened, and again before merging.

1. Fork the repo.

2. Create a branch from `master`. If you're addressing a specific issue, prefix your branch name with the issue number.

3. If you've added code that should be tested, add tests.

4. If you've changed APIs, update the documentation.

5. Run `yarn test` and ensure the test suite passes.

6. Use `yarn format` to format and lint your code.

7. PR's must be rebased before merge (feel free to ask for help).

8. PR should be reviewed by two maintainers prior to merging.

## License

By contributing to Netlify CMS, you agree that your contributions will be licensed

under its [MIT license](LICENSE).