# How to contribute

Thank you for reading this, we always welcome help make this project even better! Without the help from our contributors this project would not be possible.

## Code of Conduct

Please read and follow our [Code of Conduct][coc].

## Setup

Run `npm install` to install dependencies.

Run `npm run build:lib` to build.

Run `npm run start:example-app` to start the example app.

### Testing

Run `npm install` to install dependencies.

Run `npm run build:lib` to build.

Run `npm run test:ci:local` to test the same way the build validation will test.

## Submission Guidelines

### Submitting a feature request

You can open an issue and request a new feature. Please add some information on what you would like, how it should work. Small features can be added directly as a Pull Request without opening an issue first.

### Submitting a bug report

Search through other issues to see if there is already a previous issue for the bug. Provide a small reproduction using a StackBlitz project or a GitHub repo.

### Submitting a Pull Request

- Please test your code by running the tests.

- Rebase your branch against the current master. We might ask you to rebase again once your branch is out of date.

- If you add new functionality please also add tests to validate this.

- Reference issues as a PR comment.

- Follow our code guidelines.

## Commit Message Guidelines

We have very precise rules over how our git commit messages can be formatted. This leads to \*\*more

readable messages\*\* that are easy to follow when looking through the \*\*project history\*\*. But also,

we use the git commit messages to \*\*generate the Angular Tree Component change log\*\*.

### Commit Message Format

Each commit message consists of a \*\*header\*\*, a \*\*body\*\* and a \*\*footer\*\*. The header has a special

format that includes a \*\*type\*\*, a \*\*scope\*\*, and a \*\*subject\*\*:

```

<type>(<scope>): <subject>

<BLANK LINE>

<body>

<BLANK LINE>

<footer>

```

The \*\*header\*\* is mandatory and the \*\*scope\*\* of the header is optional.

Any line of the commit message cannot be longer 100 characters! This allows the message to be easier

to read on GitHub as well as in various git tools.

The footer should contain a [closing reference to an issue](https://help.github.com/articles/closing-issues-via-commit-messages/) if any.

Samples: (even more [samples](https://github.com/CirclonGroup/angular-tree-component/commits/master))

```

docs(changelog): update changelog to beta.5

```

```

fix(release): need to depend on latest rxjs and zone.js

The version in our package.json gets copied to the one we publish, and users need the latest of these.

```

### Revert

If the commit reverts a previous commit, it should begin with `revert:`, followed by the header of the reverted commit. In the body it should say: `This reverts commit <hash>.`, where the hash is the SHA of the commit being reverted.

### Type

Must be one of the following:

- \*\*build\*\*: Changes that affect the build system or external dependencies (example scopes: gulp, broccoli, npm)

- \*\*ci\*\*: Changes to our CI configuration files and scripts (example scopes: Travis, Circle, BrowserStack, SauceLabs)

- \*\*docs\*\*: Documentation only changes

- \*\*feat\*\*: A new feature

- \*\*fix\*\*: A bug fix

- \*\*perf\*\*: A code change that improves performance

- \*\*refactor\*\*: A code change that neither fixes a bug nor adds a feature

- \*\*style\*\*: Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)

- \*\*test\*\*: Adding missing tests or correcting existing tests

### Scope

The scope should be the name of the npm package affected (as perceived by the person reading the changelog generated from commit messages).

The following is the list of supported scopes:

- \*\*component\*\*

- \*\*directive\*\*

- \*\*model\*\*

- \*\*mobx\*\*

### Subject

The subject contains a succinct description of the change:

- use the imperative, present tense: "change" not "changed" nor "changes"

- don't capitalize the first letter

- no dot (.) at the end

### Body

Just as in the \*\*subject\*\*, use the imperative, present tense: "change" not "changed" nor "changes".

The body should include the motivation for the change and contrast this with previous behavior.

### Footer

The footer should contain any information about \*\*Breaking Changes\*\* and is also the place to

reference GitHub issues that this commit \*\*Closes\*\*.

\*\*Breaking Changes\*\* should start with the word `BREAKING CHANGE:` with a space or two newlines. The rest of the commit message is then used for this.

Example:

```

feat(scope): commit message

BREAKING CHANGES:

Describe breaking changes here

BEFORE:

Previous code example here

AFTER:

New code example here

```

[coc]: https://github.com/CirclonGroup/angular-tree-component/blob/master/CODE\_OF\_CONDUCT.md