# Contributing to Spectator

Contributions to Spectator - whether in the form of ? pull requests, ? bug reports, ? documentation or ? feature ideas - are more than welcome!

# Developing

## Building and testing the library

To develop Spectator locally, simply do the following:

Install dependencies:

```bash

npm install

```

Run the tests:

```bash

npm test

```

Build the library:

```bash

npm run build

```

## Using `npm link`

If you want to test your changes in an app that consumes Spectator, you can do the following:

`cd` to the library build output directory:

```bash

cd dist

```

Tell `npm` to use this package when asked to `link`:

```bash

npm link

```

Then, in the root of your consuming app,

Tell `npm` to link to the package:

```

npm link @ngneat/spectator

```

Run tests while preserving symlinks:

```

ng test --preserve-symlinks

```

# Committing changes

Before you commit, please ensure that your code passes the existing unit tests.

New features should be accompanied by new tests.

## Commit message format

When you are ready to commit, please prepare your commit message in the following 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.

Githooks are used to enforce this format and to help you see how to improve your commit message.

# Thanks for contributing! ?