# Contributing

To contribute to `webpack-bundle-analyzer`, fork the repository and clone it to your machine. [See this GitHub help page for what forking and cloning means](https://help.github.com/articles/fork-a-repo/)

## Setup packages

Next, install this package's dependencies:

```sh

npm i

```

## Develop with your own project

Run the following to build this library and watch its source files for changes:

```sh

npm run start

```

You will now have a fully functioning local build of this library ready to be used. \*\*Leave the `start` script running\*\*, and continue with a new Terminal/shell window.

Link the local package with `yarn` and/or `npm` to use it in your own projects:

```sh

# Needed if your own project uses `yarn` to handle dependencies:

yarn link

# Needed if your own project uses `npm` to handle dependencies:

npm link

```

Now go to your own project directory, and tell `npm` or `yarn` to use the local copy of `webpack-bundle-analyzer` package:

```sh

cd /path/to/my/own/project

# If you're using yarn, run this:

yarn link webpack-bundle-analyzer

# ...and if you're not, and you're using just npm in your own

# project, run this:

npm link webpack-bundle-analyzer

```

Now when you call `require('webpack-bundle-analyzer')` in your own project, you will actually be using the local copy of the `webpack-bundle-analyzer` project.

If your own project's Webpack config has `BundleAnalyzerPlugin` configured with `analyzerMode: 'server'`, the changes you do inside `client` folder within your local copy of `webpack-bundle-analyzer` should now be immediately visible after you refresh your browser page. Hack away!

## Send your changes back to us! :revolving\_hearts:

We'd love for you to contribute your changes back to `webpack-bundle-analyzer`! To do that, it would be ace if you could commit your changes to a separate feature branch and open a Pull Request for those changes.

Point your feature branch to use the `master` branch as the base of this PR. The exact commands used depends on how you've setup your local git copy, but the flow could look like this:

```sh

# Inside your own copy of `webpack-bundle-analyzer` package...

git checkout --branch feature-branch-name-here upstream/master

# Then hack away, and commit your changes:

git add -A

git commit -m "Few words about the changes I did"

# Push your local changes back to your fork

git push --set-upstream origin feature-branch-name-here

```

After these steps, you should be able to create a new Pull Request for this repository. If you hit any issues following these instructions, please open an issue and we'll see if we can improve these instructions even further.

## Add tests for your changes :tada:

It would be really great if the changes you did could be tested somehow. Our tests live inside the `test` directory, and they can be run with the following command:

```sh

npm run test-dev

```

Now whenever you change some files, the tests will be rerun immediately. If you don't want that, and want to run tests as a one-off operation, you can use:

```sh

npm test

```