# Contribute to Stryker

This is the contribution guide for Stryker. Great to have you here! Here are a few ways you can help make this project better.

## Team members

\* Simon de Lang: Stryker dev. Started Stryker as part of his thesis at Info Support. [simondel](http://github.com/simondel) on github.

\* Nico Jansen: Stryker dev. Originally came up with the idea for the mutation testing framework as a thesis at Info Support.

[nicojs](http://github.com/nicojs) on github or [@\_nicojs](https://twitter.com/\_nicojs) on twitter

\* Sander Koenders: Stryker dev. Creator of the stryker-webpack-plugin and stryker-jest-runner.

## Learn & listen

Get in touch with us through twitter or via the [Stryker Slack](https://join.slack.com/t/stryker-mutator/shared\_invite/enQtOTUyMTYyNTg1NDQ0LTU4ODNmZDlmN2I3MmEyMTVhYjZlYmJkOThlNTY3NTM1M2QxYmM5YTM3ODQxYmJjY2YyYzllM2RkMmM1NjNjZjM).

\* [@stryker\_mutator](https://twitter.com/stryker\_mutator)

\* [@\_simondel](https://twitter.com/\_simondel)

\* [@\_nicojs](https://twitter.com/\_nicojs)

## Code style

Please adhere to our [editorconfig](https://editorconfig.org) and [eslint](https://eslint.org/) rules. If you're using vscode, please install the following extensions:

\* The [editorconfig extension](https://github.com/editorconfig/editorconfig-vscode#editorconfig-for-visual-studio-code)

\* The [eslint extension](https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint)

We configured the tslint extension to run on save in or [vscode workspace](#vscode-environment-configuration).

## Runner Stryker locally

We use [Lerna](https://lerna.js.org/) to manage the packages in this repository. You don't have to install it globally. The packages themselves can be found in the [packages folder](https://github.com/stryker-mutator/stryker/tree/master/packages). Commands such as `npm test` can be executed from the root of the project but executing them inside of a package folder is more time efficient. However, we suggest running `npm test` in the root of the project before a commit to ensure that everything still works. To get Stryker running locally, please follow these steps:

1. Clone the repository

1. Install dependencies using `npm install` in the root of the project

1. Run `npm run build` in the root of the project once.

1. Use `npm start` to run the TypeScript compiler in watch mode. Any changes you make to TypeScript files will automatically compile.

After that, you're ready to go. If you're using [vscode, please setup that as well](#vscode-environment-configuration).

Here are some common tasks to use. Just remember that they don't include compiling of the sources. So they all assume the latest code is compiled and ready to go:

\* Use `npm test` to run the unit tests. Using `npm test` in one of the package folders also works.

\* Use `npm run e2e` will install and execute the end to end tests (located in the e2e folder). These take a while.

\* Use `npm run perf` will install and execute the performance tests (located in the perf folder). These take a while.

## VSCode environment configuration

We've chosen to \*\*check in in our vscode configuration\*\*. This makes development unified amongst stryker developers. VSCode is an open source code editor maintained by Microsoft. For more info and the download link, please visit https://code.visualstudio.com/.

We recommend you to install the following plugins:

\* [editorconfig](https://marketplace.visualstudio.com/items?itemName=EditorConfig.EditorConfig), to adhere to our white spacing rules.

\* [eslint extension](https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint)

\* [code spell checker](https://marketplace.visualstudio.com/items?itemName=streetsidesoftware.code-spell-checker), no particular reason, just prevent common typos.

After cloning this repo, open the workspace with `$ code workspace.code-workspace` (or open code and use file -> Open Workspace...).

Some quick notes to help you get started:

1. On the left side, you can see all stryker projects and plugins. Open files from there.

1. Use `CTRL+Shift+B` (or `?+Shift+B` on OSX) to run the \*build task\*. This runs `npm start`, compiling any changes you make in the background.

1. Use `CTRL+Shift+D` (or `??D` on OSX) to open up the \*debug\* pane. Here you can select a config to run. For example: select "Unit tests (stryker-api)" to run the unit tests for the `stryker-api` package.

\* You can run the tests with `CTRL+F5` (or `?F5` on OSX).

\* You can debug the tests with `F5` (also `F5` on OSX). Setting breakpoints in your code and inspecting variables all work as expected.

Have fun!

## Running Stryker on Stryker

We support mutation testing Stryker with Stryker! After you got Stryker working locally, you can follow these steps to mutation test Stryker:

1. Make sure all Stryker packages are build: `npm run build`

1. Navigate to the package you want to mutation test, for example `cd packages/core`

1. Run `npm run stryker`

## Adding new features

New features are welcome! Either as requests or proposals.

1. Please create an issue first or let us know via the [Stryker Slack](https://join.slack.com/t/stryker-mutator/shared\_invite/enQtOTUyMTYyNTg1NDQ0LTU4ODNmZDlmN2I3MmEyMTVhYjZlYmJkOThlNTY3NTM1M2QxYmM5YTM3ODQxYmJjY2YyYzllM2RkMmM1NjNjZjM)

1. Create a fork on your github account.

1. When writing your code, please conform to the existing coding style.

See [.editorconfig](https://github.com/stryker-mutator/stryker/blob/master/.editorconfig), the [typescript guidelines](https://github.com/Microsoft/TypeScript/wiki/Coding-guidelines) and our tslint.json

\* You can check if there are lint issues using `npm run lint:log`. Output will be in root folder in `lint.log` file.

\* You can automatically fix a lot of lint issues using `npm run lint:fix`

1. Please create or edit unit tests or integration tests.

1. Run the tests using `npm test`

1. When creating commits, please conform to [the angular commit message style](https://docs.google.com/document/d/1rk04jEuGfk9kYzfqCuOlPTSJw3hEDZJTBN5E5f1SALo/edit).

Namely in the form `<type>(<scope>): <subject>\n\n[body]`

\* Type: feat, fix, docs, style, refactor, test, chore.

\* Scope can the the file or group of files (not a strict right or wrong)

\* Subject and body: present tense (~changed~\*change\*, ~added~\*add\*) and include motivation and contrasts with previous behavior

Don't get discouraged! We estimate that the response time from the

maintainers is around a day or so.

## Tools

As Stryker uses the Abstract Syntax Tree (AST) to create mutations, we recommend you to use [ASTExplorer](https://astexplorer.net/) to get familiar with the concepts involved.

# Bug triage

Found a bug? Don't worry, we'll fix it, or you can ;)

Please report a bug report on our [issues page](https://github.com/stryker-mutator/stryker/issues). In this please:

1. Label the issue as bug

2. Explain what the bug is in clear English

3. Include reproduction steps

This can be an example project, code snippit, etc

4. Include the expected behaviour.

5. Include actual behaviour.

6. Add more details if required (e.g. which browser, which test runner, which versions, etc)

# Community

Do you want to help? Great! These are a few things you can do:

\* Evangelize mutation testing

Mutation testing is still relatively new, especially in JavaScript. Please help us get the word out there!

Share your stories in blog posts an on social media. Please inform us about it!

\* Did you use Stryker? Your feedback is very valuable to us. Good and bad! Please contact us and let us know what you think!