# How to contribute to Prism

## Development

Yarn is a package manager for your code, similar to npm. While you can use npm to use Prism in your own project, we use yarn for development of Prism.

1. If you don't already have the yarn package manager on your machine, install [yarn](https://yarnpkg.com/lang/en/docs/install/).

2. Fork the [https://github.com/stoplightio/prism](https://github.com/stoplightio/prism) repo.

3. Git clone your fork (i.e. `git clone https://github.com/<your-username>/prism.git`) to your machine.

4. Run `yarn` to install dependencies and setup the project.

5. Because during the development we run the software directly on top of TypeScript sources, we advise you to use our script: `cd packages/cli && yarn cli mock openapi.yaml`.

6. Run `git checkout -b [name\_of\_your\_new\_branch]` to create a new branch for your work. To help build nicer changelogs, we have a convention for branch names. Please start your branch with either `feature/{branch-name}`, `chore/{branch-name}`, or `fix/{branch-name}`. For example, if I was adding a CLI, I would make my branch name: `feature/add-cli`.

7. Make changes, write code and tests, etc. The fun stuff!

8. Run `yarn test` to test your changes.

9. Commit your changes.

10. Don't forget to `git push` to your branch after you have committed changes.

Now, you are ready to make a pull request to the Stoplight repo! ?

If this is your first Pull Request on GitHub, here's some [help](https://egghead.io/lessons/javascript-how-to-create-a-pull-request-on-github).

We have a pull request template setup that you will fill out when you open your pull request.

> We try to respond to all pull requests and issues within 7 days. We welcome feedback from everyone involved in the project in open pull requests.

### Dependencies

If you are adding a new `devDependency`, add it to the root workspace `package.json`: `yarn add -D -W {packageName}`.

If you are adding a new runtime dependency, add it to the relevant `package.json` file (inside of `prism-core`, `prism-http`, etc).

### Testing

Prism has an extensive test suite. To run it, just use the regular `test` script

```bash

yarn test

# or

npm test

```

We also have an harness test that requires some more setup. In general you do not need to run this on your computer but if you really have nothing better to do, you can run it by executing the following commands:

```bash

yarn build.binary

yarn test.harness

```

There's a dedicated README.MD file in the `test-harness` directory in case you want to know what's going on.

### Debugging

The best way to debug a Prism behavior is probably to attach your debugger to the CLI and go from there. To make that happen:

```bash

cd packages/cli

yarn cli:debug mock file.oas.yml

```

The application will wait for a debugger to be attached and break on the first line; from there, you can put your breakpoint here and there and help us debug the software!

#### What is this `fp-ts` all about?

`fp-ts` is the library containing functions and data structures that help Prism lean toward a functional style. It might be annoying to step into its functions; fortunately according to your IDE, you might be able to skip the code. In case you're using Visual Studio Code, you can use the `skipFiles` section of your `launch.json` file. You can cobble something like this:

```json

{

"configurations": [

{

"type": "node",

"request": "launch",

"name": "Mock file",

"autoAttachChildProcesses": true,

"skipFiles": ["node\_modules/fp-ts/\*.js"],

"program": "${workspaceRoot}/packages/cli/src/index.ts",

"args": ["mock", "${input:oasFile}"],

"cwd": "${workspaceRoot}/packages/cli",

"runtimeArgs": ["-r", "ts-node/register/transpile-only", "-r", "tsconfig-paths/register"]

}

]

}

```

### Common issues

1. I am receiving weird errors from TypeScript, but I didn't touch any part of the build process!

Prism is using TypeScript's incremental compiler capability that sometimes does not work. The best way to fix the issue is to simply remove any compiled file as well the incremental files:

```sh

yarn build --clean

```

## Support

For support questions, please use the [Stoplight Community forum](https://community.stoplight.io/c/open-source). If you are unsure if you are experiencing a bug, the [forum](https://community.stoplight.io/c/open-source) is a great place to start.

If you have found a bug, please create an issue.

We try to respond to all pull requests and issues within 7 days.