# Contribution Guide

We welcome and encourage community contributions to editor.

Since the project is still unstable, there are specific priorities for development. Pull requests that do not address

these priorities will not be accepted until the ORY Editor is production ready.

Please familiarize yourself with the Contribution Guidelines and Project Roadmap before contributing.

There are many ways to help editor besides contributing code:

- Fix bugs or file issues

- Improve the documentation

## Contributing Code

Unless you are fixing a known bug, we \*\*strongly\*\* recommend discussing it with the core team via a GitHub issue before

getting started to ensure your work is consistent with the ORY Editor's roadmap and architecture.

All contributions are made via pull request. Note that \*\*all patches from all contributors get reviewed\*\*. After a pull

request is made other contributors will offer feedback, and if the patch passes review a maintainer will accept it with

a comment. When pull requests fail testing, authors are expected to update their pull requests to address the failures

until the tests pass and the pull request merges successfully.

At least one review from a maintainer is required for all patches (even patches from maintainers).

Reviewers should leave a "LGTM" comment once they are satisfied with the patch. If the patch was submitted by a maintainer

with write access, the pull request should be merged by the submitter after review.

## How to run, develop, and contribute

Do you want to run, develop or contribute to React Page? For that you need [Node](https://nodejs.org) installed on

your system. Use git to check out this repository as followed.

```bash

$ git clone https://github.com/react-page/react-page.git

$ cd editor

```

### Install dependencies

React Page is a [monorepo](https://github.com/babel/babel/blob/master/doc/design/monorepo.md) that you initialize with:

```bash

$ yarn

$ yarn bootstrap

```

we use [yarn](https://yarnpkg.com/lang/en/), but npm should work as well.

### Run the example(s)

Here are some [examples](examples/) that are a good starting point if you want to familiarize yourself with the editor.

To run the examples, use one of the following commands:

```

$ yarn start

```

### Run the toolchain

The tool chain contains tests and tslint. It is highly recommended to run this while developing.

```bash

# run the tests in watch mode

$ yarn run test:watch

# run tslint in watch mode

$ yarn run lint:watch

```

### Run the documentation

To run the guide in watch mode, do:

```bash

$ yarn run docs:guide

```

To generate API docs, run:

```bash

$ yarn run docs:api

```

### Recommended tools

Feel free to use whatever works for you, these works for us. Especially care about using "prettier" when writing code as it will avoid merge conflicts on code style.

IDE: vscode

Vscode extensions: prettier, tslint, code spell checker, beautify css/sass/scss/less

## Known issues

### Types resolution error

In case you change a lot of files, especially in core or UI, you might end up seeing old versions of these files when working on plugins. To fix this, run

```

$ yarn run build:lib

```

Which builds the library code. If this doesn't help (and you're in vscode), make sure to reload window (CTRL+SHIFT+P -> Reload Window). That forces vscode to reinitialize typescript declaration files.

### Other issues

Known issues are tracked in the [issues tab](https://github.com/react-page/react-page/issues?q=is%3Aissue+is%3Aopen+label%3Abug).