# ReDoc Contributing Guide

Hi! We're really excited that you are interested in contributing to ReDoc. Before submitting your contribution though, please make sure to take a moment and read through the following guidelines.

- [Issue Reporting Guidelines](#issue-reporting-guidelines)

- [Pull Request Guidelines](#pull-request-guidelines)

- [Development Setup](#development-setup)

- [Project Structure](#project-structure)

## Issue Reporting Guidelines

- Before filing a new issue, try to make sure your problem doesn’t already exist.

- The best way to get your bug fixed is to provide a reduced test case.

## Pull Request Guidelines

Before submitting a pull request, please make sure the following is done:

1. Fork the repository and create your branch from master.

2. Run `npm install` in the repository root.

3. If you’ve fixed a bug or added code that should be tested, add tests!

4. Ensure the test suite passes (`npm test`). Tip: `npm test -- --watch TestName` is helpful in development.

5. Format your code with prettier (`npm run prettier`).

## Development Setup

You will need [Node.js](http://nodejs.org) at `v12.0.0+`.

After cloning the repo, run:

```bash

$ npm install # or npm

```

### Commonly used NPM scripts

``` bash

# dev-server, watch and auto reload playground

$ npm start

# start playground app in production environment

$ npm run start:prod

# runt tslint

$ npm run lint

# try autofix tslint issues

$ npm run lint -- --fix

# run unit tests

$ npm run unit

# run e2e tests

$ npm run e2e

# Make sure you have created bundle before running e2e test

# E.g. run `npm run bundle` and wait for the finishing process.

# open cypress UI to debug e2e test

$ npm run cy:open

# run the unit tests (includes linting and license checks)

$ npm test

# prepare bundles

$ npm run bundle

# format the code using prettier

$ npm run prettier

# auto-generate changelog

$ npm run changelog

```

There are some other scripts available in the `scripts` section of the `package.json` file.

## Project Structure

- \*\*`benchmark`\*\*: contains basic perf benchmark. Not fully ready yet

- \*\*`demo`\*\*: contains project demo with demo specs and HMR playground used in development

- `demo/playground`: HMR Playground used in development

- \*\*`docs`\*\*: contains extra docs (linked from README.md)

- \*\*`e2e`\*\*: contains e2e tests. The e2e tests are written and run with [Cypress](https://www.cypress.io/).

- \*\*`src`\*\*: contains the source code. The codebase is written in Typescript. CSS styles are managed with [Styled components](https://www.styled-components.com/). State is managed by [MobX](https://github.com/mobxjs/mobx)

- \*\*`src/common-elements`\*\*: contains common Styled elements or components used in multiple places

- \*\*`src/components`\*\*: contains main visual components

- \*\*`src/services`\*\*: contains different services used by ReDoc including MobX stores

- \*\*`src/services/models`\*\*: contains classes for OpenAPI entities (e.g. Response, Operations, etc)

- \*\*`src/types`\*\*: contains extra typescript typings including OpenAPI doc typings

- \*\*`src/utils`\*\*: utility functions

- \*\*`src/styled-components.ts`\*\*: - reexports styled-components with proper typescript annotations using theme

- \*\*`src/theme.ts`\*\*: - default theme (colors, fonts, etc) used by all the components