# Contributing to CodeSandbox Client

## Table of Contents

- [Code of Conduct](#code-of-conduct)

- [Code Organization](#code-organization)

- [Setting Up the project locally](#setting-up-the-project-locally)

- [Submitting a Pull Request](#submitting-a-pull-request)

- [Add yourself as a contributor](#add-yourself-as-a-contributor)

## Code of Conduct

We have a code of conduct you can find [here](./CODE\_OF\_CONDUCT.md) and every

contributor is expected to obey the rules therein. Any issues or PRs that don't

abide by the code of conduct may be closed.

## Code Organization

The CodeSandbox client is currently divided in to 5 parts. We use `lerna` to

share dependencies between these parts.

- `app`: The editor, the search, profile page, the embed and the sandbox.

- `homepage`: The Gatsby website of the homepage.

- `common`: All common parts between these packages, reusable JS.

- `codesandbox-api`: The npm package that's responsible for communication

between the sandbox and the editor.

- `codesandbox-browserfs`: An in-browser file system that emulates the Node JS

file system API and supports storing and retrieving files from various

backends. Forked from

[https://github.com/jvilk/BrowserFS](https://github.com/jvilk/BrowserFS), with

an additional

[CodeSandbox backend](https://github.com/codesandbox/codesandbox-client/blob/master/standalone-packages/codesandbox-browserfs/src/backend/CodeSandboxFS.ts).

This version of CodeSandbox is using the production server as source of truth,

this is specified by the environment variable `LOCAL\_SERVER`. If you're working

on a feature that needs you to be logged in, you can login on

[https://codesandbox.io/](https://codesandbox.io/) and copy the contents of the

`jwt` local storage key over to your development environment on

[http://localhost:3000/](http://localhost:3000/). \*\*Be very careful with how you

handle the token\*\*, as anyone who knows it can login as you and have read/write

access to all your CodeSandbox content!

\*\*Working on your first Pull Request?\*\* You can learn how from this \_free\_

series

[How to Contribute to an Open Source Project on GitHub](https://egghead.io/series/how-to-contribute-to-an-open-source-project-on-github)

## Setting Up the project locally

To install the project you need to have `yarn` and `node`

1. [Fork](https://help.github.com/articles/fork-a-repo/) the project, clone

your fork:

```sh

# Clone your fork

git clone https://github.com/<your-username>/codesandbox-client.git

# Navigate to the newly cloned directory

cd codesandbox-client

```

2. Your environment needs to be running Node v. 10

- `.nvmrc` config exists in the repo root, specifying a v.10.x.x version

- you can use [fnm](https://github.com/Schniz/fnm) (`fnm use`) to change

your current node version to the one specified in `.nvmrc`

3. from the root of the project: `yarn` to install all dependencies

- make sure you have latest `yarn` version

4. from the root of the project: `yarn start`

- this builds the dependencies (`codesandbox-api` and

`codesandbox-browserfs`) and runs the `app` development environment,

available on [http://localhost:3000/s/new](http://localhost:3000/s/new)

- on subsequent runs you can also bypass dependencies building and use

`yarn start:fast`

- if you want to work on the homepage, start it with `yarn start:home`, it

will be available on [http://localhost:8000/](http://localhost:8000/)

> Tip: Keep your `master` branch pointing at the original repository and make

> pull requests from branches on your fork. To do this, run:

>

> ```sh

> git remote add upstream https://github.com/codesandbox/codesandbox-client.git

> git fetch upstream

> git branch --set-upstream-to=upstream/master master

> ```

>

> This will add the original repository as a "remote" called "upstream," then

> fetch the git information from that remote, then set your local `master`

> branch to use the upstream master branch whenever you run `git pull`. Then you

> can make all of your pull request branches based on this `master` branch.

> Whenever you want to update your version of `master`, do a regular `git pull`.

5. If you want to debug the state of the app, install the

[Cerebral Debugger](https://github.com/cerebral/cerebral-debugger/releases)

and connect it to the port `8383`. After that, if you refresh the app, you

should be able to see the state, the sequences executed and so on. See

[documentation](https://cerebraljs.com/docs/introduction/devtools.html) for

reference.

## Submitting a Pull Request

Please go through existing issues and pull requests to check if somebody else is

already working on it, we use `someone working on it` label to mark such issues.

Also, make sure to run the tests and lint the code before you commit your

changes.

```sh

yarn test

yarn lint

```

Before running `yarn lint`, you must have build our `common` and `notifications`

packages.

```sh

yarn build:deps

```

## Add yourself as a contributor

This project follows the

[all-contributors](https://github.com/all-contributors/all-contributors)

specification. Contributions of any kind welcome!

To add yourself to the table of contributors on the `README.md`, please use the

automated script as part of your PR:

```sh

yarn contributors:add

```

Follow the prompt and commit `.all-contributorsrc` and `README.md` in the PR.

Thank you for taking the time to contribute! ?