# Contributing

Contributions welcome! Please follow the [code of conduct](./CODE\_OF\_CONDUCT.md).

## Overview

[Yarn workspaces](https://yarnpkg.com/lang/en/docs/workspaces/) are used to manage dependencies and

build config across packages in the umbrella `vx` monorepo, and

[lerna](https://github.com/lerna/lerna/) is used to manage versioning.

## Project structure

```

vx/

lerna.json

package.json

packages/

vx-package-1/

src/

test/

build/

package.json

...

vx-package-2/

...

...

```

## Local development

Run the following to setup your local dev environment:

```sh

# Install `yarn`, alternatives at https://yarnpkg.com/en/docs/install

curl -o- -L https://yarnpkg.com/install.sh | bash

# Clone or fork `vx`

git clone git@github.com:hshoff/vx.git # or your fork

cd vx

# install dependencies, and have `yarn` symlink within-`vx` dependencies

yarn

# build packages and generate types for local development

yarn build

```

#### Rebuild one package

Upon modification of a single `package` you can run

```sh

# build the package as cjs version

yarn build-one --workspaces=@vx/package

# build the esm version (the @vx/demo next server sources these files)

yarn build-one --workspaces=@vx/package --esm

# generate d.ts(definition files) for a lib

yarn type-one --workspaces=@vx/package --esm

```

from the `vx` monorepo root to re-build the package with your changes.

#### Running demo pages

You can use the local [`next.js`](https://nextjs.org) dev server within `packages/vx-demo` to view

and iterate on your changes in the gallery. From the `packages/vx-demo` folder run `yarn dev` to

start the next server which (if correctly sym-linked) will also watch for changes you make to other

packages (upon re-building them).

#### Config generation

`vx` uses [`@airbnb/nimbus`](https://github.com/airbnb/nimbus) to generate build configuration for

`eslint`, `prettier`, `jest`, `babel`, and `typescript`.