# Contributing to Reach UI

Reach UI is built and tested with [Yarn](https://yarnpkg.com). Please follow their [install instructions](https://yarnpkg.com/docs/install) to get Yarn installed on your system.

Then, run these commands:

```

git clone git@github.com:reach/reach-ui.git

cd reach-ui

yarn install

yarn build

```

## Root Repo Scripts:

```sh

yarn build # builds all packages

yarn start # starts storybook server

yarn test # runs tests in all packages

yarn release # publishes changed packages

```

## `website` directory scripts

The website uses [Gatsby](https://gatsbyjs.org) v2 with [Gatsby MDX](https://github.com/ChristopherBiscardi/gatsby-mdx) powering most of the pages. It is deployed with [now](https://now.sh)

```

yarn build # builds the production site to "public/"

yarn start # starts the website

yarn stage # deploys the site with now.sh

yarn deploy # alias the latest deploy to production

```

## Running / Writing Examples

First do the steps in "Getting started", then start the Storybook server:

```

yarn start

```

Next, put a file in `packages/<component-dir>/examples/<name>.example.js` and make it look like this:

```jsx

import React from "react";

// The name of the example (always name the variable `name`)

let name = "Basic";

// The example to render (always name the function `Example`)

function Example() {

return <div>Cool cool cool</div>;

}

// Assign the name to the example and then export it as a named constant

Example.story = { name };

export const Comp = Example;

// Default export an object with the title matching the name of the Reach package

export default { title: "Dialog" };

```

Now you can edit the files in `packages/\*` and storybook will automatically reload your changes.

\*\*Note\*\*: If you change an internal dependency you will need to run `yarn build` again. For example, if working on `MenuButton` requires a change to `Rect` (an internal dependency of `MenuButton`), you will need to run `yarn build` for the changes to `Rect` to show up in your `MenuButton` example.

## Running / Writing Tests

First do the steps in "Getting Started", then:

```

yarn test

```

Or if you want to run the tests as you edit files:

```

yarn test --watch

```

Often you'll want to just test the component you're working on:

```

cd packages/<component-path>

yarn test --watch

```