



Reusable Component Library in React

Reduce, reuse... React!



Who is this human?



- Connor Wilson
- Front-end @ Tulip, Formerly Mappedin, recovering freelancer
- Instructor @ Bridge
- Socials on cwlsn.com, me@cwlsn.com
- Cat whisperer



What is this talk?

- I built a boilerplate for creating a Component library, documented the process on GitHub and published to npm
- Discussion of advantages of using a separate library
- Personal project process on GitHub
- Clone/fork and get started right away



Examples

- **Shopify Polaris**
- Material UI
- Instacart Snacks
- Reakit
- Khan Academy React Components
- React Bootstrap
- Onsen
- Atlassian
- Fabric (Microsoft)
- Gestalt (Pinterest)
- more!

Doing Silly Extras for Motivation During Person Projects



Spoilers

- Final result: <https://rinsejs.io/> (Git repo with docz deployed by Netlify)
- GitHub: <https://github.com/cwlsn/rinse-react>
- Article:
<https://itnext.io/how-to-write-your-own-reusable-react-component-library-a57dc7c9a210>



Add Some Aesthetics

- Thinking of a name and making a basic logo for each personal project helps me get through the last 10% of the work
- Align with some Emoji for Tweets/GitHub
- People usually think I am joking



Write Requirements in Issues

- Talking to yourself on GitHub can feel strange, but having documentation for requirements and following Git flow is a huge resume booster
- Example of my first opened issue:
<https://github.com/cwlsn/rinse-react/issues/1>
- Close with GitHub keywords like “Fixes” in PR
- Demonstrate your knowledge of process in public

Doing Webpack Config But Simple

**How do we feel about
Webpack?**



Annotated Webpack Config

- <https://github.com/cwlsn/rinse-react/blob/master/webpack.config.js>
- Handles JS with Babel, Sass, and images

Create a Component



Picking a Folder Structure

```
src
  .. components
    .... Button
      ..... index.js (export file)
      ..... Button.js (actual component code)
      ..... README.md (document each component!)
```



Basic Component Code

- Using styled-components
- PR here: <https://github.com/cwlsn/rinse-react/pull/4/files>
- How you make basic Components is up to you, tools like Generact will help you quickly duplicate save overhead



Component Requirements Come from App needs

- Iterate your reusable components as your needs change
- Try to be additive and track your versions semantically
- Update docs as needed

Visualising Components



See Components Without Linking

- Using npm's link function can be handy, but sometimes you want to work ahead
- Storybook can provide a simple environment to interact with different versions of your components
- Setting up Storybook was done in this PR:
<https://github.com/cwlsn/rinse-react/pull/6/files>
- Other options include Styleguidist, MDX

Development and Publishing



Published on NPM

- You can check out how it works once published here: <https://www.npmjs.com/package/rinse-react>

```
// In your component  
import { Button } from 'rinse-react';
```

```
// In your render function  
<Button type="submit" onClick={this.submit}>Hello!</Button>
```

Things to Add



Improving Your Codebase

- TypeScript
- Unit tests (Jest, Enzyme)
- CI via GitHub (Travis, Circle)
- What else?

Questions?

