

Atomic Design in React

How to Build React Apps that Scale for Developers

DevHub Toronto, August 9th 2018

Who is this human?

- Recovering freelance web designer of 15 years, used JavaScript to make carousels and navigation menus in WordPress
- JS developer in startups for the last 3 years
- Front-end at Tulip, Instructor at Bridge
- Feel free to interrupt me with questions, or write them down for the end
- Find social things on cwlsln.com if you want to connect!

What is Bridge?

- Bridge is this dope 11 week totally free program for women, agender, and non-binary devs, it teaches you JS at an advanced level and aims to help you go from your first job to your first awesome job
- You can follow socials at @itsbridgeschool
- if you work for a company that you think could be interested in sponsoring Bridge, which is an awesome source of talent for sponsors as well as a great diversity initiative, you can contact their Executive Director, Emily, at emily@bridgeschool.io



Spoilers

1. What is Atomic Design?
2. Pieces of the Puzzle
3. Putting it in React

What Does Atomic Mean?

- Chemistry!
- Compare user interfaces to all matter
- Atoms compose molecules
- Molecules compose organisms
- Organisms compose... the rest of the stuff

But Why?

- We're not designing pages, we're designing systems of components.—Stephen Hay
- The design systems that create our applications are getting increasingly complex
- Projects can have hundreds of components, making small fixes requires deeper and deeper project knowledge

Why not?

- Creator of React on the correct folder structure: “move files around until it feels right”
- This can feel a little overkill for smaller apps

Atoms

- Atoms are the smallest, simplest components
- Not particularly useful on their own
- An input field, a button, a label
- Ideally pure functional components

Molecules

- Start combining atoms to create something that can actually do a thing
- Combine the input, label and button components to create a simple form
- Do one thing, do it well
- Think about reuse
- If using a state manager (redux, MobX, etc...) try to keep these as PFCs as well

Organisms

- Combining molecules is when you actually start developing what can look like a page
- Think of a product page, repeating the same product molecule
- Or a header, with molecules for a search form, navigation and logo

Templates and Pages

- Templates combine your organisms into what will form your pages, but they don't have real content
- A simple blog may have templates for a home page, post page, a generic page for non-post content
- Pages provide templates real content and are inevitably what is served to the client

Pages Test the System

- By the pages part, you have context on how to test your components
- How does a content area look with a ton of words? Just one sentence? What about a button in a different language?
- How does a page work when each of the props of the components are utilized?

Atomic Design in React

- There are many ways to structure a React app
- React doesn't care how you build it, or how you provide content
- Usually, you would have a folder in your source code called "components", maybe JS files in src
- Components may be JS files, folders, could contain tests, styling
- I prefer a folder per component, with everything that component needs

Turn Basic into Atomic

- Inside your component folder, each atomic component gets its own folder
- Components folder down to 5 children
- Atomic folders will be like a reverse pyramid
- Even a 30 component atoms folder is understandable

ARc by @diegohaz

- An open source enthusiast way smarter than me has made a sweet boilerplate to use with examples, and extensions
- Can optionally come with redux and server side rendering
- Born from a failed project with over 100 components
- Almost 2k stars, comes with 100% test coverage

Magical Dynamic Export

- How do we use the components when they're deeper in the folder structure?
- Can use webpack aliases
- Or use this magical snippet and be able to refer to every component from "components"

Upcoming Event!

- Workshop for Developers Series: React-Redux
August 11th, right here at the Devhub
<https://www.eventbrite.ca/e/workshop-for-developers-series-react-redux-tickets-47854449948>

Resources

- ARc Boilerplate Repo
<https://github.com/diegoHaz/arc>
- Magic components/index.js export
<https://github.com/diegoHaz/arc/wiki/Atomic-Design#do-not-worry>
- Read the official book online
<http://atomicdesign.bradfrost.com/table-of-contents/>

Questions?