

Topics

1. Frontend Folder Structure - *Best Practices from around the web*
2. Documentation for React UI Components - `react-styleguidist`
3. Atomic Design - *What is that? Can it help us?*

1. React Folder Structure

"Code != Clothing, or How to neatly structure your code"



First stop: grouping files by “type”

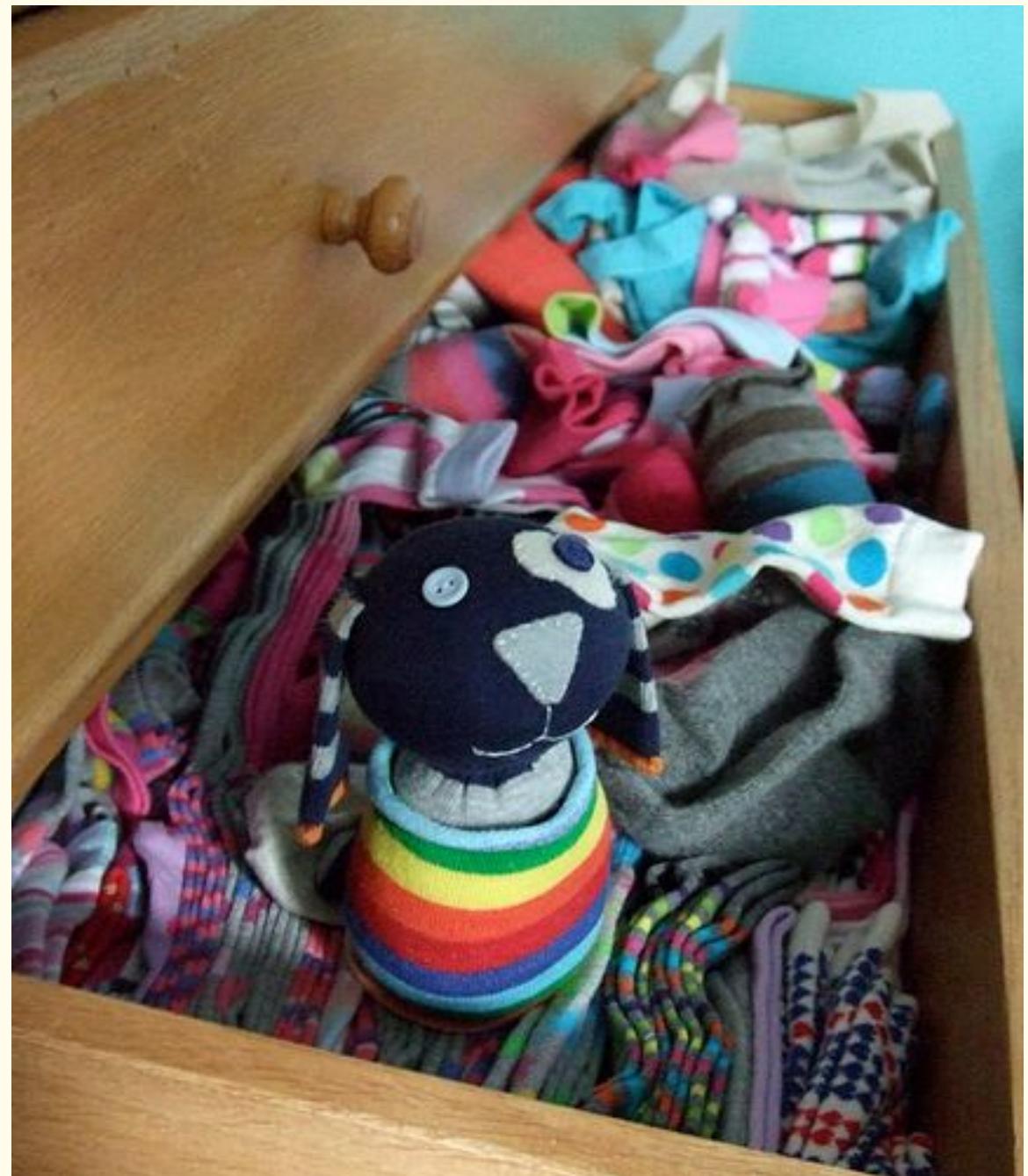
```
src
└── components
    └── ...
    
└── actions (action creators)
    ├── auth.js
    └── product.js

└── types (action types)
    ├── auth.js
    └── product.js

└── reducers
    ├── auth.js
    └── product.js
```

The Sock Drawer:

You know where your socks are but how
about your whole outfit?



A different (better?)
approach:

The Uniform == Modularity



Grouping files by feature

```
src
  └── components <-- for basic, shared UI
      ├── Button.js
      ├── Select.js
      └── ...
  └── auth
      ├── components <-- components specific to the feature (like)
          └── Login.js
          └── PasswordReset.js
      ├── actions.js
      ├── reducers.js
      ├── types.js
      └── selectors.js
  └── product
      ├── components
      ├── actions.js
      ├── types.js
      └── ...
  ...
```

Bottom line

- What do we want as a team?
- Status quo vs "group by feature"
- Thoughts?

2. React-Styleguidist



Start documenting your components

```
/**  
 * Description of the component if needed  
 */  
const Button = props => (...);  
  
Button.propTypes = {  
  /** No user action on button is allowed */  
  disabled: PropTypes.bool,  
  
  /** Primary theme color */  
  primary: PropTypes.bool,  
  
  /** Shadow effect */  
  raised: PropTypes.bool  
};
```

The component above will produce this clear output in style guide:

It's like **Swagger** but for UI code

Button

[src/components/Button/Button.js](#) □

Description of the component if needed

PROPS & METHODS

Prop name	Type	Default	Description
disabled	bool	false	Indicates no action is allowed
primary	bool	false	Primary theme color
raised	bool	false	Shadow effect

Documentation lives next to the code!

Styleguidist will look for `README.md` or `ComponentName.md` files by default.

```
components/
  Button/
    Button.js
    Button.md  <--- here
```

Button.md

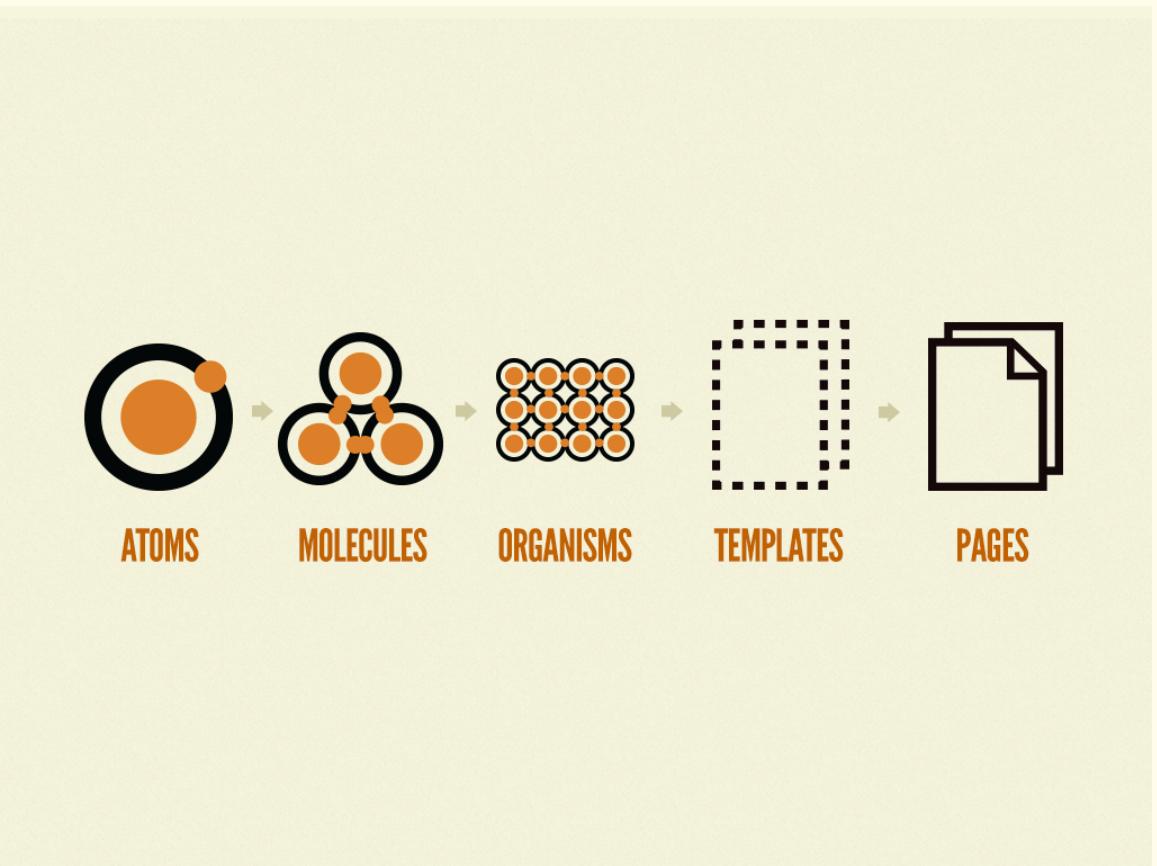
```
<Button primary>Click me!</Button>
```

And the result looks like this:



3. Atomic Design

<https://atomicdesign.bradfrost.com/>



Atoms

SEARCH THE SITE

LABEL

ENTER KEYWORD

INPUT

SEARCH

BUTTON

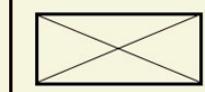
Molecules

SEARCH THE SITE

ENTER KEYWORD

SEARCH

Organisms



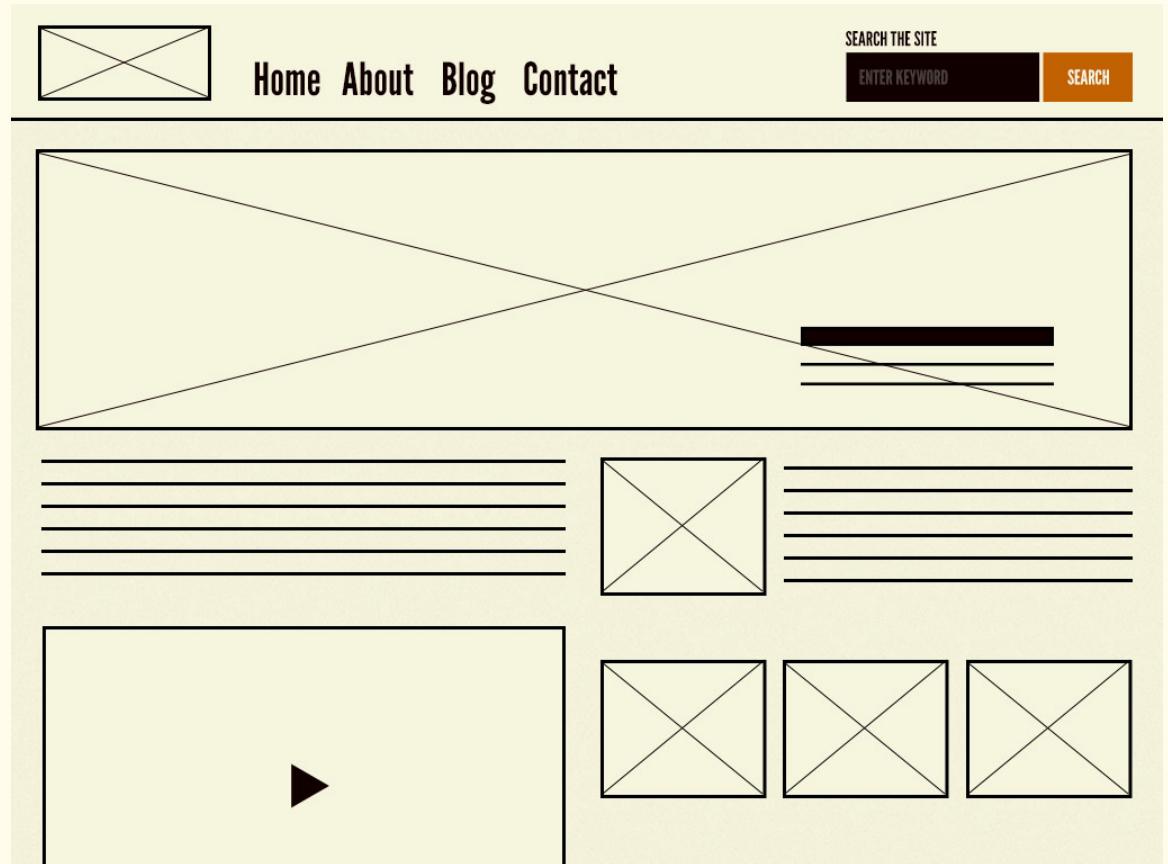
[Home](#) [About](#) [Blog](#) [Contact](#)

SEARCH THE SITE

ENTER KEYWORD

SEARCH

Pages



Links

- React Folder Structure
 - <https://medium.com/hackernoon/my-journey-toward-a-maintainable-project-structure-for-react-redux-b05dfd999b5>
 - <https://www.robinwieruch.de/react-folder-structure>
 - <https://aelia.co/how-to-neatly-structure-your-code/>
 - <https://www.codementor.io/@zmitry/react-app-structure-by-feature-guidelines-10wnfwp9eh>
 - <https://softwareengineering.stackexchange.com/questions/338597/folder-by-type-or-folder-by-feature>
- React-Styleguidist
 - <https://react-styleguidist.js.org/>
- Atomic Design
 - <https://atomicdesign.bradfrost.com/>