React 101

continued

Overview

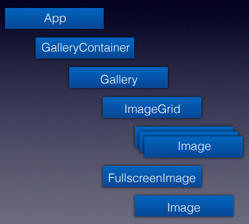
- Recap of last time
- Modern JavaScript
- Routing - react-router
- State Management - redux

Tooling

We used the package manager **yarn** and **create-react-app** to setup our app.

With zero configuration this gave us the module bundler **webpack** that gave us **live reloading** and **eslint** inside the browser, and **babel** that allowed us to use **Modern Javascript**

ImageGallery app



general layout
fetch images
show/hide fullscreen
layout grid
square image
fullscreen overlay

React

- Use **props** to pass data from parent to child components
- Local to a component **state** is used for changing data
- **Unidirectional data flow** down the tree keeps changes predictable
- Passed down **callbacks** to have for example the **Gallery** to react on a **Image** being clicked

React patterns

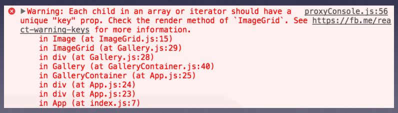
- **UI first** to iterate faster
- **Presentational** vs **container** components, or how it looks vs how it works
- I proclaimed my 💖 for having JS, CSS & HTML in a **single file**

Modern JavaScript

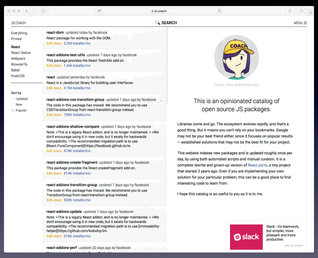
- const / let instead of var
- arrow functions
- template strings
- destructuring
- rest / spread operator
- default parameters
- object shorthand names
- classes
- importing / exporting modules

What's 🐛 me

- Low resolution full screen image
- Warning in the console



```
▶Warning: Each child in an array or iterator should have a   proxyConsole.js:56
unique "key" prop. Check the render method of `ImageGrid`. See https://fb.me/rea
ct-warning-keys for more information.
    in Image (at ImageGrid.js:15)
    in ImageGrid (at Gallery.js:29)
    in div (at Gallery.js:28)
    in Gallery (at GalleryContainer.js:40)
    in GalleryContainer (at App.js:25)
    in div (at App.js:24)
    in div (at App.js:23)
    in App (at index.js:7)
```



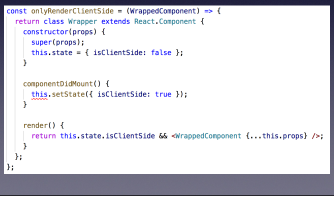
Use js.coach to find great React packages

React Router

- Declarative dynamic routing library for **React** and since **v4** also for **React Native**
- Documentation has great interactive examples
- Build your own React Router v4 is a nice blogpost where a simplified implementation is build

Higher order components

- Composition over inheritance
- A function that wraps a Component

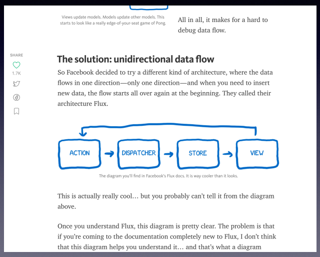


```
const onlyRenderClientSide = (WrappedComponent) => {
  return class Wrapper extends React.Component {
    constructor(props) {
      super(props);
      this.state = { isClientSide: false };
    }

    componentDidMount() {
      this.setState({ isClientSide: true });
    }

    render() {
      return this.state.isClientSide && <WrappedComponent {...this.props} />;
    }
  };
};
```

State Management



Views update models. Models update other models. This starts to look like a really edge-of-your-seat game of Pong.

All in all, it makes for a hard to debug data flow.

SHARE

1.7K

The solution: unidirectional data flow

So Facebook decided to try a different kind of architecture, where the data flows in one direction—only one direction—and when you need to insert new data, the flow starts all over again at the beginning. They called their architecture Flux.

The diagram you'll find in Facebook's Flux docs. It is way cooler than it looks.

This is actually really cool… but you probably can't tell it from the diagram above.

Once you understand Flux, this diagram is pretty clear. The problem is that if you're coming to the documentation completely new to Flux, I don't think that this diagram helps you understand it… and that's what a diagram

Redux Terminology

- Store
- Action

```
{
  type: 'INCREMENT',
  ...
}
```

- Action Creator
- Action Type
- Reducer

```
(state, action) => state
```

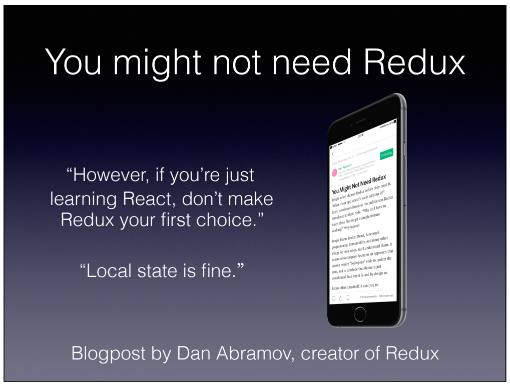
- Selector

🐥🐣

- Ducks: Redux Reducer Bundles
- Bundle Actions, Action Types and Reducers in a single file

Blogposts

- Course by Dan Abramov that also builds a small implementation
  - Getting Started with Redux
  - Building React Applications with Idiomatic Redux



You might not need Redux

"However, if you're just learning React, don't make Redux your first choice."

"Local state is fine."

Blogpost by Dan Abramov, creator of Redux