

A Google Slides presentation slide. The title 'React and Redux Fundamentals' is displayed in large white font. Below it, 'Day 1 of 4' is written in a smaller white font. At the bottom left, there is a URL: 'Course Portal - <http://code.walmart.com/react-redux-fundamentals/>'. The slide has a blue background with abstract geometric shapes like triangles and rectangles in orange, blue, and white.

React and Redux Fundamentals

Day 1 of 4

Course Portal - <http://code.walmart.com/react-redux-fundamentals/>

A Google Slides presentation slide. The number '1.' is displayed in large blue font. Below it, the question 'What is Electrode?' is written in white font. The slide has an orange background with abstract geometric shapes like triangles and rectangles in blue, white, and orange.

1.

What is Electrode?

A screenshot of a Google Slides presentation. The slide features a decorative header with overlapping blue and orange diagonal bars. The main content is a quote in a serif font: "The platform takes care of the grunt work, leaving the application developer free to build the best possible customer experience quickly". Below the quote is a attribution line: "- Jack Herrington". At the bottom of the slide is a navigation bar with icons for back, forward, search, and settings, and the text "Slide 4".

A screenshot of a Google Slides presentation. The slide features a decorative header with overlapping blue and orange diagonal bars. The main content is a section titled "Packages included in Electrode" in a large, bold, blue sans-serif font. Below the title is a list of packages in a smaller, blue sans-serif font, using nested bullet points:

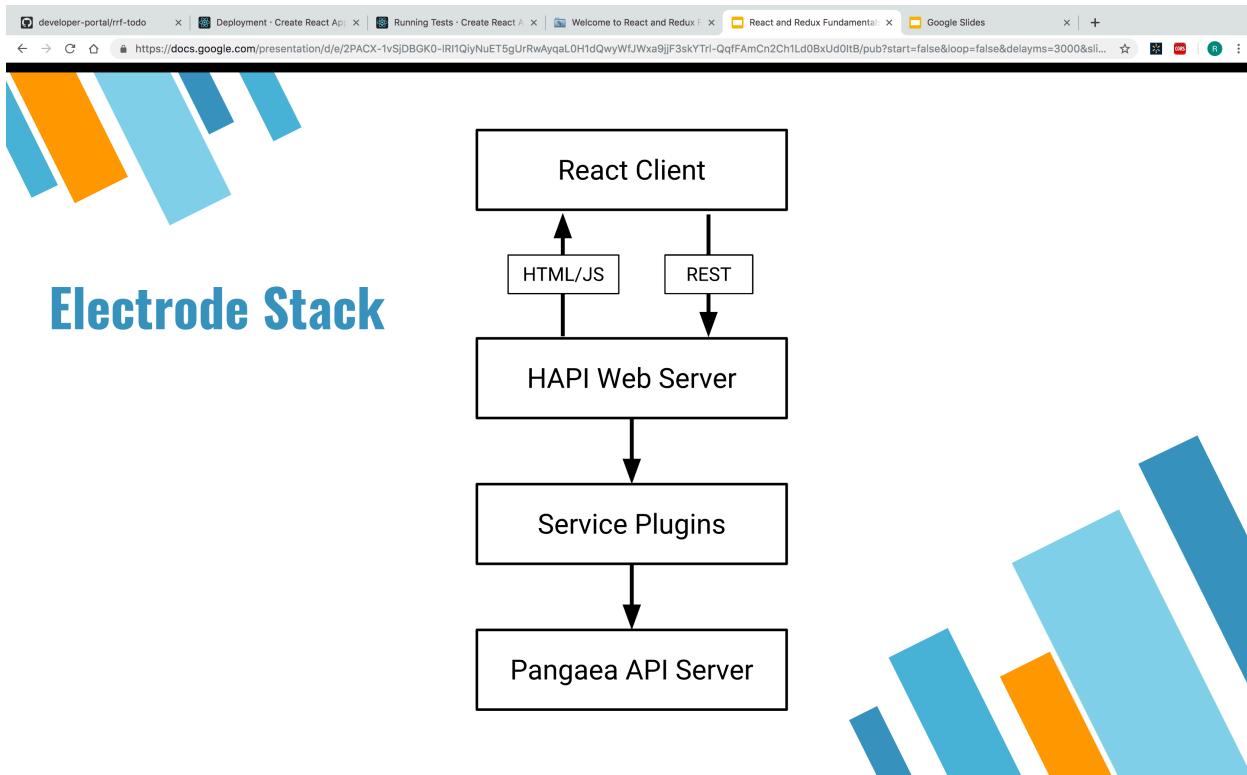
- » React
 - ◊ Redux
 - ◊ React-Router
- » Hapi.js

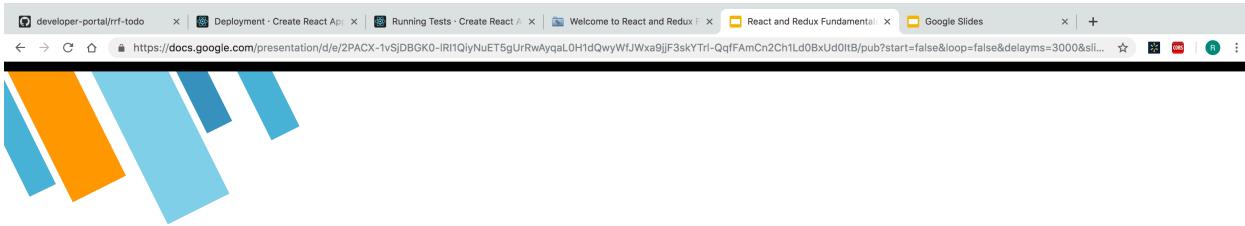
At the bottom of the slide is a navigation bar with icons for back, forward, search, and settings, and the text "Slide 5".

Packages included in Electrode

- [React](#) - an awesome JavaScript library for building user interfaces, created by Facebook.
- [Redux](#) - a predictable state container for JavaScript apps.
- [React Router](#) - a powerful routing library built on top of React.
- [CSS Modules](#) - a CSS file in which all class names and animation names are scoped locally by default. Fixes the problem of the global scope in CSS.
- [Universal rendering](#)
- [Webpack](#) - a powerful module bundler.
- [Webpack Isomorphic Loader](#) - a powerful tool that makes NodeJS require understand files such as images for SSR.
- [Babel](#) - a utility to transpile ES6 + 7.
- [ESLint](#) - a pluggable linting utility for Javascript.
- [Mocha](#) - a feature-rich Javascript testing framework.
- [Enzyme](#) - a Javascript testing utility for React, created by airbnb.
- [TravisCI](#) - a continuous integration service to build and test software projects.
- [Gulp](#) - a Javascript build tool that lets us automate tasks.
- [Yeoman](#) - a Scaffolding tool for modern webapps.
- [History](#) - a Javascript library for managing session history.
- [Bluebird](#) - a great Javascript promise library.
- [Electrode Configper](#) - a versatile and flexible utility for managing configurations of Node.js applications.
- [Electrode JWT CSRF](#) - a module to enable stateless Cross-Site Request Forgery (CSRF) protection with JWT.
- [Electrode-Redux-Router-Engine](#) - an Electrode routing and rendering engine using react-router and redux.
- [Component Caching](#) - an optimizer to improve React Server Side Rendering speed
- [Electrode-Server](#) - a configurable web server using Hapi.js on top of Node.js.
- [Electrify](#) - a tool for analyzing the module tree of webpack projects.
- [Electrode-Docgen](#) - a custom metadata extractor for the Electrode framework, automates component documentation..

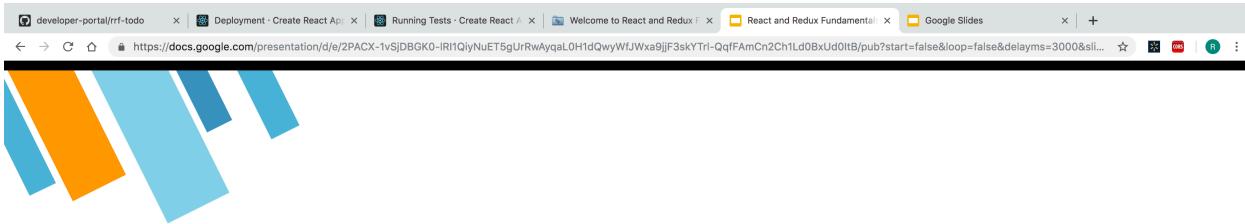
◀ ▶ ⟲ ⟳ 🔍 ⚙️





Electrode Public vs Electrode Walmart

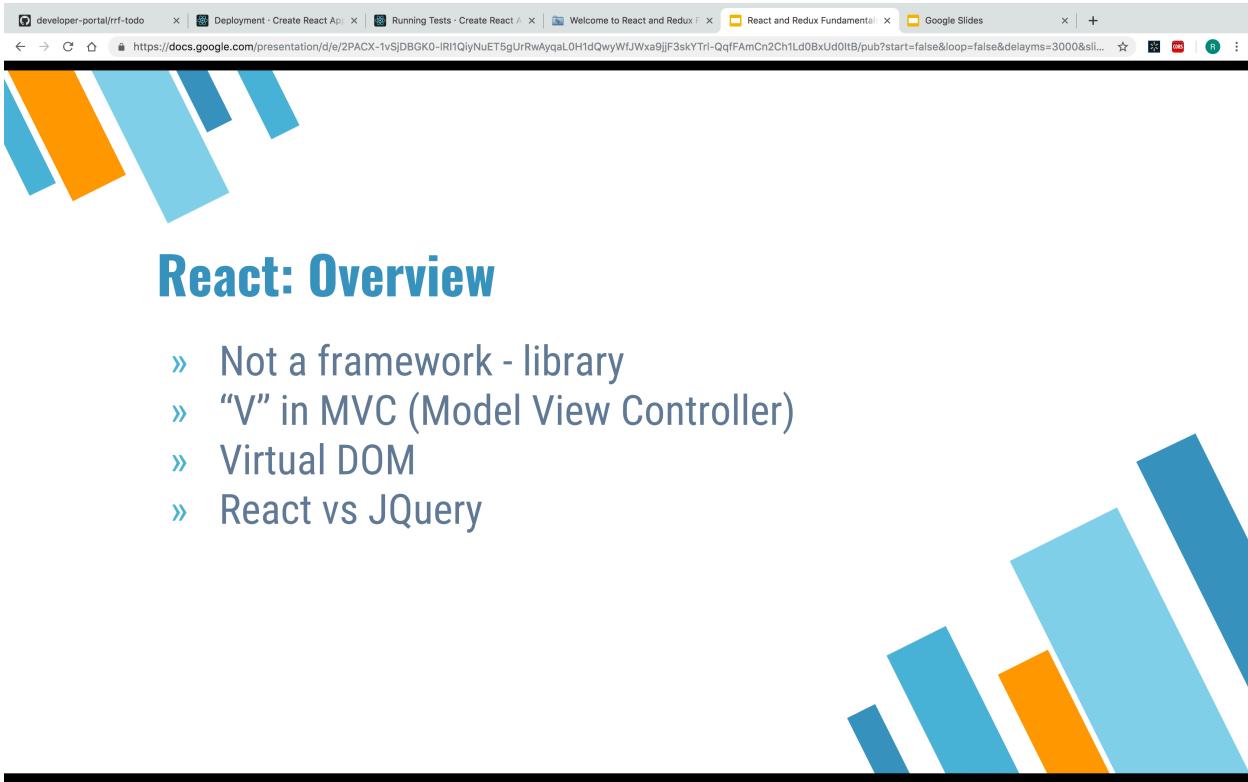
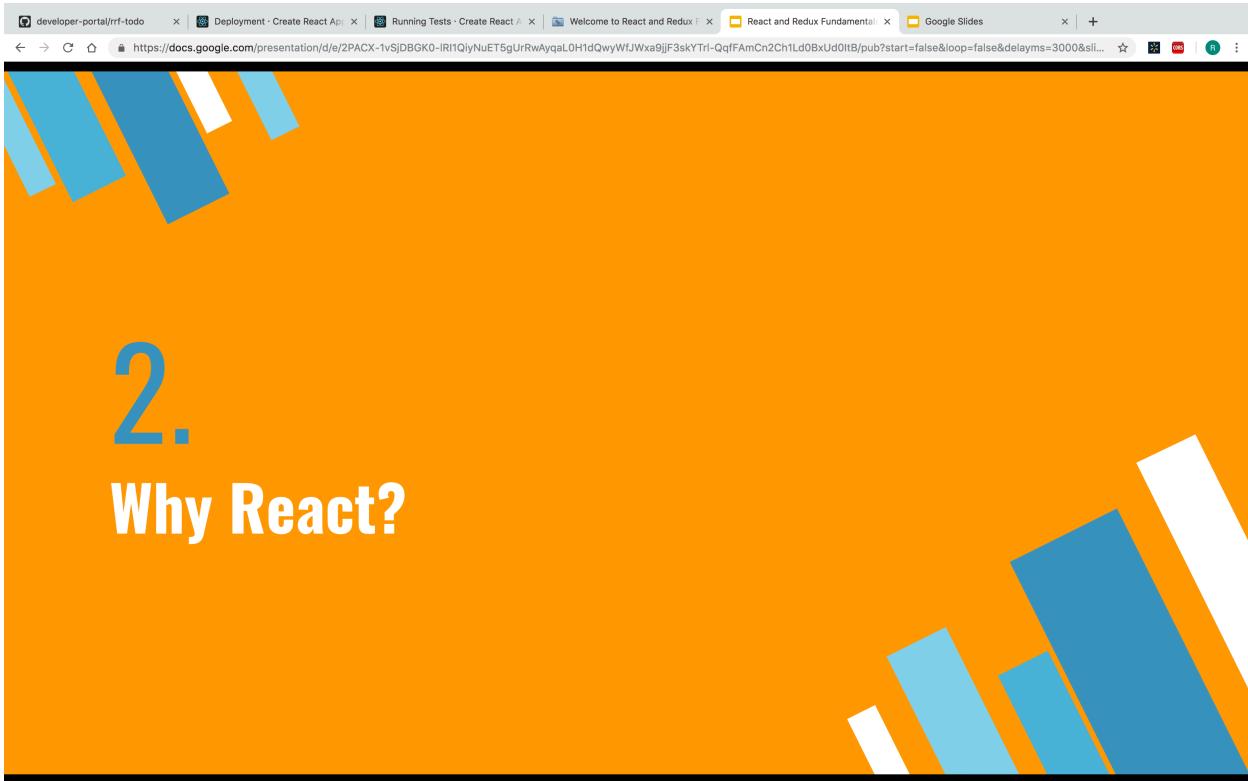
- » Most is open source
- » Internally, includes Walmart eCommerce platform integrations
- » You'll investigate the slightly different "Walmart Way" in class and assignment



Electrode Debugging

- » **Chrome debugger**
 - ◊ Mac (⌘ + ⌥ + J) / PC (CTRL + SHIFT + J)
 - ◊ Add to your code:
 - ◊ `console.log("your_log");`
 - ◊ `debugger;`
- » **Electrode logs**
 - ◊ Open in a separate window -
<http://localhost:2992/reporter>







React: Reusable Components

- » Increase code reuse
- » Standardize things like logging, analytics, and testing!
- » Stateful/Class vs Presentational/Function
- » Props to pass data
- » JSX



Presentational / Function

```
1 import React from 'react';
2
3 const FunctionComponent = (props) => {
4   const functionScopeVar = true;
5
6   return (
7     <main>
8       <h1>props.title</h1>
9       <button onClick={props.handleClick}>Click Me!</button>
10      </main>
11    );
12  }
13
14 export default FunctionComponent;
15
```



```
1 import React, { Component } from 'react';
2 import TitleEditor from './title-editor';
3
4 class StatefulComponent extends Component {
5   constructor(props) {
6     this.state = {
7       title: "Electrode Rocks!"
8     }
9
10   this.updateTitle = this.updateTitle.bind(this)
11 }
12
13 updateTitle(title) {
14   this.setState({ title });
15 }
16
17 render() {
18   return (
19     <main>
20       <h1>{this.state.title}</h1>
21       <TitleEditor handleUpdate={this.updateTitle} />
22     </main>
23   )
24 }
25 }
26
27 export default StatefulComponent;
```

Stateful / Class Component

3. Why Redux?