**Code 501 Course Desc – React/Redux**

Overview

React and Redux's popularity among JavaScript developers is quickly growing. Created by the minds at Facebook, React has earned the industry's respect and a positive reputation in an endless world of front-end frameworks. If you’re starting a project from scratch in JavaScript, you should consider using React.

In this class, JavaScript developers will study the React framework, learning how to build a front-end project from scratch using React and Redux.

Outcomes

At the end of this course students will:

* Emerge with a thorough knowledge of the React ecosystem, including JSX for templating, Redux for state management, React-Router for front-end routing, Enzyme for testing, and Gatsby for static site generation.
* Be able to scaffold, design, build, and test several different styles of front-end websites.
* Enhance their skills with Lodash and functional programming styles.
* Implement a single-page application where data is retrieved from RESTful API services.
* Generate a static website from collections of files.

Prerequisites

So that everyone is starting at relatively the same point, students enrolling in this coures should have:

* Fluent knowledge of HTML and the browser DOM
* Intermediate to advanced knowledge of JavaScript (and JSON)
* Familiarity with the browser event cycle
* Experience with REST APIs, AJAX, and related technologies
* Experience working in a pair programming environment
* Knowledge of source control (Git)

To participate adequately and get the most out of the course, students should also be at least passingly familiar with:

* Some knowledge of Webpack basics (Read [this handy tutorial](https://blog.madewithlove.be/post/webpack-your-bags/) for a refresher)
* ES2015 (otherwise known as ES6) JavaScript, including arrow functions, object destructuring, class syntax, and Node module import and export
* Functional programming style, including map, reduce, and filter, and object immutability
* HTML5 semantic markup
* Testing frameworks for front-end code, including headless browsers (e.g. PhantomJS) and test runners (e.g. Mocha)

Topics

Lesson 1

* Context: Why React matters
* The component model
* Passing props and the component model
* Components as pure functions
* Components as ES6 classes
* The React/Webpack build process

Lesson 2

* Passing data between components
* Bound functions and callbacks
* Component lifecycle hooks (getInitialState, componentWillMount, etc.)
* Asynchronous actions (e.g. AJAX)
* Forms in React (controlled vs. uncontrolled)
* Client-side routing with React-Router

Lesson 3

* Brief functional programming review
* Introduction to Redux for state management
* Dividing components into “smart” and “presentational”
* Asynchronous actions with Redux
* Middleware for API calls

Lesson 4

* Scaffolding a testing framework with Enzyme
* Mocking
* Testing pure Redux functions
* Static website generation with Gatsby
* Stateless component architecture

Lesson 5

* Extending the React ecosystem
* React-Bootstrap
* CSS modules
* GraphQL
* Course wrap-up

About the Instructor

[**Need a specific instructor for this, probably contract?**]

## Learn with Stacked Modules

Concepts in each of our courses are taught using stacked modules, where a new concept is introduced in each class session, building upon what came before it. This is a challenging style that requires persistence, practice, and collaboration, but allows more concepts to be introduced over the length of the course. This method helps students learn and retain more information in a short period of time. [Learn more about stacked modules »](https://www.codefellows.org/blog/how-to-accelerate-your-learning-with-stacked-modules)