Course Outline

React and Redux

Course Summary

Description

React is a JavaScript library that makes creation of beautiful and capable web applications possible with much less work. It allows software developers to write ultra-modern web-based programs as part of large teams. Redux is also a JavaScript library. Redux makes the management of web data extremely controllable. The combination of the two is powerful!

In this course you'll learn both topics from scratch. We'll lay a foundation of the modern use of ECMAScript and the new tooling like webpack, npm, and Babel. Then we launch into an understanding of Redux from the ground-up, moving all the way into advanced topics like middleware and asynchronous Ajax calls through dispatchers. Finally, we'll learn how React works and how to integrate Redux into it.

We will get you up and running rapidly, preparing you for real-world application development with a deep understanding of React components and well-structured applications. If you are new to React and Redux, or if you've been working with it but not quite getting how it really works, this is the course for you!

Objectives

After taking this course, students will be able to:

- Use modern ES2015+ JavaScript features with or without React and Redux
- Simplify complex data management with the Redux library
- Handle asynchronous operations with Redux -- including Ajax
- Understand React's custom markup language, JSX
- Design and create React components
- Work with React props and state
- Use forms with React
- Configure and use the React Router

Topics

- React Introduction
- Redux Intro
- Creating the Store
- State and Subscriptions
- Actions and Reducers 101
- Actions
- Advanced Actions
- Reducer Composition
- Redux Middleware
- Ajax with Redux
- Redux-thunk
- Course Overview
- Create-React-App
- Stateless Functional Components
- Styling React components
- Events in React
- Composition with React
- React Router
- Expressions
- Managing State
- The React Lifecycle
- Prop Types

Audience

Experienced web developers who want to be equipped to handle large-scale web applications.

Prerequisites

A very strong grasp of advanced JavaScript. Please ask about our 5-day JavaScript course which will prepare you for the high-level of JavaScript fluency needed for this course. Very solid HTML5 and CSS knowledge is helpful

Duration

Five Days

Course Outline

React and Redux

Course Outline

I. Course Overview

II. React Introduction
- A. What is React?
- B. Its origins WRT Angular
- C. React's 3 design principles
- D. Composition of components
- E. How is React so darned fast?
- F. The virtual DOM
- G. One-way data flow
- H. How a React app works - a peak under the covers
- I. Transpiling, bundling, minifying

III. Redux intro
- A. It's a library
- B. The single responsibility principle
- C. Why Redux?
- D. Why Redux with React?
- E. Its history and growth
- F. The 4 concepts of Redux
- G. State
- H. Pure functions
- I. Compoosition
- J. Immutability
- K. The parts of Redux
- L. The big picture of Redux

IV. Creating the store
- A. Installing and including Redux
- B. The createStore() method
- C. The simplest possible store
- D. Redux ducks

V. State and Subscriptions
- A. Single source of truth
- B. Initializing state
- C. What goes in state and what should not
- D. Why subscriptions
- E. How to subscribe in Redux

VI. Actions and Reducers 101
- A. Actions are objects
- B. The shape of actions
- C. Type and payload
- D. Reducers are functions
- E. Why we have them
- F. The shape of a reducer
- G. Avoiding the worst Redux rookie mistakes

VII. Actions
- A. Action Creators in a React Application
- B. Dispatching Actions
- C. Mapping Actions to prop Names

VIII. Advanced Actions
- A. Action constants
- B. Action type enumerations
- C. Action creators
- D. Action creator enumerations

IX. Reducer composition
- A. The problem: complex state => complex reducers
- B. The solution: Create state slices
- C. Creating reducers to handle slices
- D. Combining reducers with Redux's built-in combineReducers
- E. Doing it manually
- F. Why manually is the right way

X. Redux Middleware
- A. The Open-Closed Principle
- B. Introduction to middleware
- C. The next() function
- D. The required shape of middleware
- E. Middleware's super-powers
- F. Recipes and examples
- G. Why you must register middleware and how to do it

XI. Ajax with Redux
- A. The problem with async calls in Redux
- B. The trick to making a good middleware function
- C. Registering the middleware
- D. Dispatching an Ajax call
- E. Making RESTful API calls
- F. How to process them into redux and then into React

XII. Redux-thunk (Time permitting)
- A. The need for a thunk
- B. How redux-thunk fits that need
- C. Installing and using redux-thunk properly
- D. Dispatching Ajax calls with redux-thunk

Course Outline (cont'd)

XIII. create-react-app
- A. The need for CRA - webpack, npm, babel, JSX, eslint, es2015, jest, etc
- B. Using npx
- C. The only build dependency needed
- D. Watch mode
- E. Linting code
- F. Ejecting
- G. Running unit tests

XIV. Stateless Functional Components
- A. Why less is more
- B. A gentle intro to JSX
- C. The 7 rules of JSX
- D. When JSX deviates from the W3C standards
- E. 3 simple steps to creating a component
- F. Hosting a component

XV. Styling React components
- A. Styling with CSS
- B. Specifying Inline Styles
- C. Importing styles
- D. Using npm libraries
- E. The trick to importing images

XVI. Events in React
- A. React's synthetic events
- B. Why do they need to add events?
- C. The quick way to tell the difference
- D. The event object is reused!
- E. Passing values to the handler
- F. Creating your own custom events

XVII. Composition with React
- A. How to compose
- B. Data flowing down - props
- C. Why not 2-way binding?
- D. Data flowing up by emitting an event
- E. Passing data between sibling and cousin components
- F. All the ways to communicate and the the cleanest solution

XVIII. React router
- A. How to create a SPA in React
- B. What React router is and where it came from
- C. 4 easy steps to routing
- D. How to define the domain of a router
- E. How to create routes
- F. How to route users via a link
- G. How to route users via a url
- H. How to route users via JavaScript
- I. How to read route parameters

XIX. Expressions
- A. Demystifying expressions - how to think about them
- B. Nesting JSX in expressions and vice-versa
- C. Conditional rendering
- D. Iterating collections
- E. What does key do for me?
- F. Calling functions that return JSX

XX. Managing state
- A. Defining state in React
- B. React state != Redux state
- C. Stateful components are classes
- D. How to initialize state in the constructor and when not to
- E. Updating state with this.setState()
- F. Asynchronous updates
- G. The secrets of upserting state
- H. Forms in React
- I. Controlled vs. Uncontrolled forms
- J. How to solve the update problem
- K. Insider tips for state in React

XXI. The React Lifecycle (Time permitting)
- A. Birth and death of a component
- B. constructor
- C. getDerivedStateFromProps
- D. componentDidMount
- E. componentWillUnmount
- F. Updating a component
- G. shouldComponentUpdate
- H. getSnapshotBeforeUpdate
- I. componentDidUpdate

XXII. PropTypes (Time permitting)
- A. Why propTypes?
- B. All the types that can be used
- C. Using in class-based components
- D. Using in functional components
- E. How to handle extra props