

React.js & Redux (FT204)

40 Hours

Outline

React is a JavaScript library for building user interfaces. It simplifies complex arbitrary DOM manipulations by using a "virtual DOM" and a diff engine, allowing you to easily transition your DOM into any arbitrary state, regardless of the current state. React implements one-way reactive data flow, which reduces boilerplate and is easier to reason about than traditional data binding. Redux takes care of application architecture implementing Flux – a pattern Facebook uses for building client-side web applications. It complements React's view components by utilizing a unidirectional data flow and managing the state of your application. Learning React and Redux will put you at the very front of modern web development industry, especially with the release of React-Native. This course will get you up and running with the essentials of React components, and creating systems architectures with Redux. You will be able to build scalable apps that run in a browser and then leverage this knowledge to build native Mobile or desktop apps. Whether you are new to react, or if you have started learning it already and need a thorough understanding of React.js & Redux - this course is the leap forward you are after.

Target Audience

This course is a good fit for both junior web developers as well as experienced developers from other development environments, wanting to take their front-end expertise to the next level and master the leading framework and techniques available today.

Prerequisites

Good working knowledge and experience with HTML, CSS and JavaScript

Objectives

- Be able to create Reactive User Interfaces with React.js & Redux
- Have a Strong foundation building modular blazing fast web applications.
- Ship with confidence top tier fully testable applications and modules.

Contents

Introduction

- Overview, project orientation, React components essentials
- ES6 new language features overview
- Node.js overview for front-end developers
- React / Redux / Universal apps - Ecosystem Bird's eye overview

Project setup

- Node.js projects & dependencies
- Webpack configuration
- Using Babel to transpile ES6

React Components

- Components structure
- JSX & React Dom
- Props & prop Types
- Stateless function components
- Managing State
- Handling Events
- Using refs
- Component lifecycle events

Nested Components

- Parent child relationships
- Updating nested components
- Trigger re-render on a parent component
- Rendering collections & lists

Testing React Components

- Overview
- Tests runners like Jasmine/Mocha
- Integrating Babel
- Expect.js tests assertion
- Spies & mocks
- React Test Utils
- Testing JSX
- Enzyme.js

Routing with React Router

- Overview
- Basic example
- Router, Route and Link
- Histories
- Nested routes
- IndexRoute
- Route parameters
- Named Components
- Query String Parameters
- Redirects
- Confirming Navigation

Flux & Redux

- Flux overview
- Redux overview
- Adding redux to our project
- State is at application level
- Store API
- Reducers Pure functions
- MapStateToProps
- MapDispatchToProps
- Actions (using FSA based redux-actions)

Testing Redux

- Overview
- Testing Redux Containers
- Testing Redux Actions
- Testing redux store initial state
- Testing Reducers

Universal apps

- Overview
- Universal app demo & example code
- Server and client side rendering
- Zoom in on the React / Redux agnostic implementation
- Best practices
- What is next? Related tools resources & references