Comprehensive React and Redux using JavaScript/ECMAScript

Course Summary

Description

In this course, students will gain hands-on experience with the latest version of React and the tools for developing React applications.

React is a JavaScript library for building web user interfaces. It is one of the most wanted and loved web frameworks today. React makes it painless to create interactive UIs on top of web standards. React is often used with the Redux library for managing the data or state in a React application.

Comprehensive React and Redux teaches developers the skills they need to immediately use React and Redux in their applications.

Objectives

At the completion of this course, Students will be able to:

- Learn to use a JavaScript package manager (either npm or Yarn)
- Quickly understand the new JavaScript language features, including classes, modules, and arrow functions
- Articulate what React is and why it is useful
- Explore the basic architecture of a React application
- Gain a deep understanding of JSX and the Virtual DOM
- Use React components to build interactive interfaces
- Create and validate forms using controlled components
- Make HTTP calls to read or change data
- Configure simple and complex routing
- Utilize Redux to manage the state of the application
- Use React and Redux together
- Learn React and Redux best practices
- Write unit tests for React using Jest and Enzyme

Topics

- Package Manager (npm/Yarn)
- Compiler Setup
- ES6/ES2015
- Project Setup (Create React App)
- Best Practices (Code Organization & Conventions)
- React Overview
- Elements
- JSX
- Virtual DOM
- Components
- Props
- Events
- State
- Lifecycle
- Conditional Rendering

- Lists
- Component Architecture
- Forms
- HTTP
- Routing (React Router)
- Hooks
- Build & Deploy
- Redux
- Using Redux with React (React Redux Library)
- Asynchronous Actions (Redux Thunk)
- Putting It All Together (React & Redux & Thunk)
- Unit Testing
- Optional Topics/FAQ (if time permits)
Comprehensive React and Redux using JavaScript/ECMAScript

Course Outline

Course Summary (cont.)

Prerequisites

All attendees must be experienced developers with an understanding of object-oriented programming and some familiarity with JavaScript.

Duration

Five Days
Comprehensive React and Redux using JavaScript/ECMAScript

Course Outline

I. Package Manager (npm/Yarn)
   A. npm
      1. What is Node.js?
      2. What is npm?
      3. Using npm
      4. npm Scripts
   B. Yarn
      1. Why Yarn?
      2. npm vs. Yarn
      3. Installation
      4. Using Yarn

II. Compiler Setup
    A. Setup (Babel)

III. ES6/ES2015
    A. Classes
    B. Scope (var, let, const)
    C. Arrow Functions
    D. Modules
    E. Template Literals
    F. Default, Rest, Spread
    G. Default
    H. Rest
    I. Spread
    J. Destructuring
    K. Optional Parameters
    L. Object.assign()
    M. Object Initializer

IV. Project Setup (Create React App)
    A. Create new Project
    B. Folder Structure
    C. Browser Support
    D. Styles and Assets
    E. Dependencies

V. Best Practices (Code Organization & Conventions)

VI. React Overview
    A. Why React?
    B. What it is?
    C. Why it is useful?
    D. Angular, React Compared
    E. Web application architectures
       1. Server-side web application architecture
       2. Single-page web application architecture
    F. React Architecture

VII. Elements
    A. Hello World in JavaScript
    B. Hello World in React

VIII. JSX
    A. Replacing createElement
    B. Embedding Expressions
    C. Specifying Attributes

IX. Virtual DOM

X. Components
    A. Creating an Element
    B. Create a Function Component
    C. Rendering a Component
    D. Creating a Class Component
    E. Composing & Reuse

XI. Props
    A. Read-only
    B. String Literals vs. Expressions
    C. Function vs. Class Components

XII. Events
    A. Listening/Subscribing/Wiring to an Event
    B. In Vanilla JavaScript
    C. In React: Function Component
    D. In React: Class Component
    E. Binding
       1. Why Binding is Necessary?
       2. Class Method
       3. Arrow Function
    F. Passing Parameters
       1. Using Arrow Functions
       2. Using Bind
    G. Handling Events
       1. Using Arrow Functions
       2. Using Bind
       3. Synthetic Events

XIII. State
    A. Defining
    B. Using State Correctly
    C. Data Flows Down
    D. Converting a Function Component to a Class Component
Comprehensive React and Redux using JavaScript/ECMAScript

Course Outline (cont.)

XIV. Lifecycle
A. What are Lifecycle Methods
B. Understanding Mounting
C. Common vs. Less Common Methods
D. Using Lifecycle Methods

XV. Conditional Rendering
A. If, else
B. Conditional Operator (?)
C. Logical (&&) Operator

XVI. Lists
A. In Vanilla JavaScript: for loop, array.forEach, array.map
B. In React: using Elements, Components
C. Why Keys are Needed

XVII. Component Architecture
A. Reuse
B. Component Communication
C. Design Patterns
   1. Container and Presentation Components
   2. Composition vs. Inheritance

XVIII. Forms
A. Controlled Components
B. Reuse of Change Logic across Multiple Inputs
C. Handling Form Submission
D. Controlling Other Form Elements: select, textarea, number
E. Validation
F. Uncontrolled Components

XIX. HTTP
A. Axios library
B. Fetch API
C. Using with React (HTTP GET)
D. Refactoring for Reuse
E. HTTP POST, PUT, DELETE

XX. Routing (React Router)
A. Installation
B. Basics
C. Handling Not Found (404)
D. Parameters (Url & Query)
E. Nesting

XXI. Hooks
A. Defined
B. Why Hooks?
C. No Breaking API Changes
D. Hooks API
E. useState
F. useEffect
G. Custom Hooks
H. Rules of Hooks

XXII. Build & Deploy
A. Building a React Application for Production
B. Deploying a React Application
C. Serving Apps with Client-Side Routing
D. Customizing Environment Variables

XXIII. Redux
A. What is Redux?
B. What is State?
C. Benefits Checklist
D. Principles of Redux
E. Core Concepts (Store, State, Reducers, Actions, Action Creators)
F. Complementary Packages
G. When do you need Redux?
H. Basic Redux Example (includes time traveling)
I. Gotchas/Tips

XXIV. Using Redux with React (React Redux Library)
A. The connect function
B. Writing mapStateToProps functions
C. Writing mapDispatchToProps Functions
D. Provider
E. Inside React Redux
F. Example

XXV. Asynchronous Actions (Redux Thunk)
A. Overview
B. Async Actions (Thunks)
C. Installation
D. Your First Thunk
E. Full CRUD Example
ProTech Professional Technical Services, Inc.

Comprehensive React and Redux using JavaScript/ECMAScript
Course Outline (cont.)

XXVI. Putting It All Together (React & Redux & Thunk)

XXVII. Unit Testing
   A. Tools (Jest, Enzyme, JSDOM)
   B. Syntax
   C. Testing Vanilla JavaScript with Jest
   D. Mocking
      1. Mocking Modules
      2. Mocking Functions
   E. Debugging Tests
   F. Component Tests
      1. Shallow
      2. Full
      3. Snapshot
   G. Testing Redux Actions & Thunks, Reducers, HTTP

XXVIII. Optional Topics/FAQ (if time permits)
   A. Styling
      1. Using Classes
      2. Using Styles
      3. CSS in JS Overview
   B. Advanced Components
      1. Higher-Order Components
      2. Render Props
      3. Context API