React Fundamentals

Duration: 3 days

Course Overview

React is a JavaScript library for building interactive user interfaces. It can be incrementally adopted, scaling easily from being used as a library to add small functionality to web pages to being used as a framework for a complete single-page-application. React also provides hooks that allow easy interaction with other JavaScript libraries and frameworks.

This course will give the student a solid and in-depth foundation for building applications that use the powerful features provided by React.

Prerequisites

Students should have a good understanding of HTML and CSS and be experienced JavaScript developers, with an advanced understanding of JavaScript objects and functions as first class citizens.

Course Outline

- Introduction
  - What is React
  - Why use React?
  - SPAs and React Web Apps
  - NextGen JavaScript Features
- React and JSX
  - Using Create React App
  - Component Basics
  - JSX Syntax
  - JSX Restrictions
  - Functional Components
  - JSX Cheatsheet
  - Props and Dynamic Content
  - State and Event Handling
  - Two-Way Binding
- Dynamic Content
  - Conditionally Rendering Content
  - Collection Content
  - Updating State Immutably
  - Collections and Keys
  - More Flexible Collections

- Styling Content
    - Dynamically Setting Styles
    - Using Class Names
    - Radium and Media Queries
    - Using CSS Modules
- Debugging
    - Understanding Error Messages
    - DevTools and Sourcemaps
    - React Developer Tools
    - Error Boundaries
- Components
    - Creating Components
    - Stateless vs. Stateful
    - Component Lifecycle
    - Pure Components
    - Virtual DOM and React Updates
    - Rendering Adjacent JSX Elements
    - Higher-Order Components
    - Validating Props and PropTypes
    - Context API
- Web Server Interactions
    - AJAX Calls
    - Using Axios
    - Rendering Fetched Data
    - Avoiding Infinite Loops
    - POSTing Data
    - Handling Errors Locally
    - Interceptors
- Routing
    - Setting Up the Router Package
    - Rendering Components for Routes
    - Using Routing-Related Props
    - Absolute vs. Relative Paths
    - Nested Routes
    - Route Guards
    - Routing and Deployment

- Forms
    - Custom Dynamic Input Components
    - Configuring a Form
    - Handling Form Submission
    - Custom Validation
    - Showing Error Messages
- Managing State with Redux
    - Complexity of State Management
    - How Redux Works
    - Reducer Functions and State Store
    - Dispatching Actions
    - Creating Subscriptions
    - Connecting React to Redux
    - Dispatching Actions from Components
- Async Redux
    - Adding Middleware
    - Redux Devtools
    - Action Creators
    - Handling Async Actions
    - Action Creators and Get State
- Testing
    - Required Testing Tools
    - What to Test?
    - Testing Components
    - Jest and Enzyme
    - Testing Containers
    - Testing Redux
- Transitions and Animations
    - Using CSS Transitions
    - Using CSS Animations
    - ReactTransitionGroup
    - Using the Transition Component
    - Wrapping the Transition Component
    - Animation Timing
    - Transition Events