React and Redux

Learn via: **Classroom / Virtual Classroom / Online**

Duration: **5 Day**

https://bilginc.com/en/training/react-and-redux-5586-training/

Overview

React is a 2nd generation web framework, created by Facebook to simplify the creation of complex browser based UI's (aka. Single Page Applications). It offers a declarative, component based approach that enables the rapid creation of SPA's without resorting to the custom attributes and dirty checking using by Angular.

The React component model is at its best when combined with modern JavaScript (ES2015+). If you want to take React to the next level then you can use it with TypeScript and add safe coding practices. We can level set delegates as part of this delivery to to use the latest features of JavaScript or TypeScript effectively with React.

As React is solely a UI framework it needs to be combined with libraries for accessing remote services, routing and state management (such as Redux). We can customise this course to your needs to include the pieces that you want to focus on.

INSTIL

Prerequisites

Delegates must be proficient JavaScript or TypeScript developers. A primer in the languages can be provided to level set all delegates to the latest language features.

What You Will Learn

Be confident developing in modern JavaScript (or TypeScript)

Understand all aspects of React development for building modern SPAs

Gain experience of decomposing applications into components

Leverage a state management library to better architect React apps

Outline

Introducing React

- How Angular and React evolved from earlier frameworks
- Overview of the React architecture and the Virtual DOM
- The trade-offs between a Virtual DOM and dirty checking
- Using the JSX syntax to create and intialise React Elements
- Comparing coding in React to Angular
- Setting up a project for React development

Modern JavaScript (Optional)

- The *let* keyword and support for block scope
- FP with Arrow Functions and the Lodash library
- The class declaration syntax and inheritance
- Destructuring items from arrays and objects
- Making use of the map and set data structures
- Enhancements to functions and creating generators

- A standard model for proxies and promises
- Using async await instead of and with Promises
- The for...of loop and spread operator

Getting Started with React

- Creating basic components that use JSX
- Using properties to pass data to components
- Nesting and managing parent-child relationships
- Transpiling both JSX and ES6 using Babel
- Adding Jest to unit test your components
- Using snapshots for rapid unit test construction
- Verifying component UI structure programmatically

Building Single Page Web Applications

- Using the lifecycle callbacks for React components
- Persisting values using the *state* property
- Designing hierarchies of components for complex UI's
- Interacting with RESTful services
- Modularising your design for unit and integration testing
- Building and testing a complete SPA based Web App
- Adding Routing to your application
- Optimising performance by removing redundant renders

React Hooks

- Functional Components vs Class Components
- Reselect's evolution into React Hooks
- Managing state with useState
- Adding side effects with useEffect
- The importance of memoization to performance
- Leveraging useMemo and useCallback
- Writing custom hooks

Styling React Apps

- Identifying styling options
- Global styles and avoiding name collisions
- Encapsulating styles using CSS Modules
- Inline styles for dynamic styles
- Using *styled-components* for dynamic styles

Advanced Topics (Optional)

- Lazy loading components and code-splitting
- Accessing references to HTML elements
- Customising *create-react-app*
- Internalisation using *react-intl*
- End to End testing with Cypress

Combining React with Redux (Optional)

- Problems when managing complex state in SPA's
- Using a framework to centralise and manage state
- Creating and using Stores, Actions and Reducers
- Manipulating your data as a sequence of transitions
- Using Redux Toolkit to reduce boilerplate
- Grouping state, actions and reducers into Slices
- Asynchronous actions, thunks and Redux middleware
- Adopting Redux Thunk as your default Redux middleware
- Advantages of Redux Saga for more complex scenarios
- Redux dev tools