GEMRAIN
CONSULTING

ReactJS with Redux
Duration: 3 days; Instructor-led3

WHAT YOU WILL LEARN

This course shows how to use Facebook's ReactJS framework to build fast, modern, scalable single page web applications (SPAs). The course covers the principles of SPA development, and then takes the students through the process of building a ReactJS user interface. The course introduces JSX, the pseudo-HTML used in building ReactJS components, and moves on creating composite components and controller views using props and state. The students will build and validate Web forms and use Facebook's unidirectional design pattern Flux, combining stores and actions to communicate with Web services and manage state. Students will gain a deeper understanding of JSX, work with form components, implement React styles, and learn how to integrate unit tests for React components.

AUDIENCE

- Programmers looking to learn React
- Developers who want to grow out of just using jQuery
- Engineers who have researched React but have had trouble mastering some concepts

PREREQUISITES

Before attending this course, students should have general programming experience and knowledge of HTML, CSS and JavaScript.

METHODOLOGY

This program will be conducted with interactive lectures, PowerPoint presentation, discussion and practical exercise.

COURSE OBJECTIVES

- Build amazing single page applications with React JS and Redux
- Master fundamental concepts behind structuring Redux applications
- Realize the power of building composable components
- Be the engineer who explains how Redux works to everyone else, because you know the fundamentals so well

- Become fluent in the toolchain supporting React, including NPM, Webpack, Babel, and ES6/ES2015 Javascript syntax

OUTLINES

DAY 1

Module 1 - Modeling Application State
- Foreword on Redux
- What is Redux?
- Managing App State with Redux
- Reducers
- Containers - Connecting Redux to React
- Implementation of a Container Class
- Containers and Reducers Review

Module 2 - Actions
- Actions and Action Creators
- Binding Action Creators
- Creating an Action
- Consuming Actions in Reducers
- Consuming Actions in Reducers Continued
- Conditional Rendering
- Reducers and Actions Review

Module 3 - Redux: Middleware
- App Overview and Planning
- Component Setup
- Controlled Components and Binding Context
- Form Elements in React
- Working with API's
- Introduction to Middleware
- Ajax Requests with Axios
- Redux-Promise in Practice
- Redux-Promise Continued
- Avoiding State Mutations in Reducers
- Building a List Container
- Mapping Props to a Render Helper

Module 4 - Charts & Maps
- Adding Sparkline Charts
- Making a Reusable Chart Component
- Labelling of Units
- Google Maps Integration
- Google Maps Integration Continued
- Project Review

DAY 2

Module 5 - React Router + Redux Form
- Overview
- Exploring the Posts API
- Installing React Router
- React Router - What is It?
- Setting Up React Router
- Route Configuration
- Nesting Of Routes
- IndexRoutes with React Router

Module 6 - Catching Data
- Introduction
- Catching Data with Posts Reducer
- Catching Data with Posts Reducer Continued
- Fetching Data with Lifecycle Methods
- Fetching Data with Lifecycle Methods Continued

Module 7 - Project Blog Post
- Creating New Posts
- Navigation with the Link Component
- Forms and Form Submission
- More on Forms and Form Submission
- Passing Control to Redux Form
- CreatePost Action Creator
- Form Validation
- Form Validation Continued
- Navigating on Submit
- Posts Index
- Dynamic Route Params
- Loading Data on Render
- Handling Null Props
- Delete Action Creator
- Navigate on Delete

Module 8 - React Router
- App Overview and Goals
- Posts API
- Installing React Router
- What React Router Does
- The Basics of React Router
- Route Design
- Our First Route Definition

- State as an Object
- Implementing Posts Reducer
- Action Creator Shortcuts
- Rendering a List of Posts
- Creating New Posts
- A React Router Gotcha
- Navigation with the Link Component

Module 9 - Redux Form
- Setting Up Redux Form
- The Field Component
- Generalizing Fields
- Validating Forms
- Showing Errors to Users
- Handling Form Submittal
- Form and Field States
- Conditional Styling
- More on Navigation
- Create Post Action Creator
- Navigation Through Callbacks
- The Posts Show Component
- Receiving New Posts
- Selecting from OwnProps
- Data Dependencies
- Caching Records
- Deleting a Post