

# Building Web Applications with React and Redux Eğitimi

## Eğitim Hakkında

React, HTML 5 web uygulamaları oluşturmak için kullanılan Facebook'tan açık kaynaklı bir JavaScript kitaplığıdır.

Bu eğitimde, sınıflar ve JSX bileşenleriyle yüksek performanslı Kullanıcı Arabirimleri ( UI'ler ) oluşturarak ve istemci tarafı uygulama işlevselliği oluşturmak için tamamlayıcı mimari modeller sağlayan Redux uygulamaları oluşturarak React 16 kullanarak eksiksiz bir uygulamanın nasıl geliştirileceğini öğreneceksiniz .

## Neler Öğreneceksiniz

### Building Web Applications Course Benefits

- Create an isomorphic web application using Facebook React and Redux
- Compose a UI using React JavaScript and JSX components
- Integrate React forms with Ajax and back-end REST services
- Apply the Container/Presentational architecture to create applications
- Integrate a variety of React extensions to create a modern application
- Continue learning and face new challenges with after-course one-on-one instructor coaching

### Building Web Applications Course Prerequisites

Students should have experience developing web pages with JavaScript, HTML, and CSS.

## Eğitim İçeriği

### Module 1: Introduction and Overview

- Features of React and Redux
- Why the Virtual DOM makes React run fast
- Configuring the development and build environment
- Incorporating third-party components

### Module 2: Designing Custom React Components



## Displaying React content

- Employing `React.createElement()` and `React.DOM.render()`
- Support well-designed components by leveraging imports
- Dynamically generating the UI

## Leveraging JSX for UI design

- Coding custom component classes
- Generating component output with the render method
- Composing a JSX component hierarchy
- Extending the hierarchy with the Container/Presentational design pattern
- Understanding the differences between JSX and HTML

## Module 3: Activating React Components

### Parameterizing components

- Disambiguating component properties and state
- Leveraging the children props to pass content dynamically
- Inserting properties into rendered output with `this.props`
- Improving maintainability and reliability with `defaultProps` and `propTypes`
- Centralizing logic at the container level by passing functions as props
- Integrating stateless functional components

### Manipulating component state

- Separating component state from component behavior
- Reading application data with `this.state`
- Mutating data with `this.setState()`
- Efficiently managing state to improve performance
- Best practices for sharing state with `createContext`

### Integrating into the component life cycle

- Attaching and detaching logic and event handlers in `componentDidMount` and `componentWillUnmount`
- Enhancing performance with `componentShouldUpdate`

### Routing between components

- Exploiting the react-router for semantic URLs
- Mapping URL paths and parameters to components

## Module 4: Developing React Forms

Manipulating form input components

- Working with interactive properties
- Capturing form component updates with onChange
- Exploiting controlled components
- Implementing two-way binding
- Applying the onChange / setState() convention
- Reducing onChange clutter for cleaner code

## Module 5: Centralizing State with Redux

### Modeling client-side data interactivity

- Architecting loosely-coupled applications for extensibility
- Managing state centrally
- Interacting with data using actions, reducers, and stores
- Defining reducers to correctly manage updates, deletes, and inserts

### Coding a Redux application

- Best practices when implementing Redux
- Creating a Redux data store
- Using the React-Redux Higher-Order Components
- Sharing state between containers and presentational components
- Integrating multiple stores/reducers in large applications to avoid complexity
- Reviewing alternatives: Mbox, alt, Relay, Flux, and Reflux

### Connecting React to RESTful services

- Making Ajax calls from React using thunk and promises
- Integrating isomorphic-fetch to support client and server-side access
- Mapping CRUD access to REST actions
- Reviewing alternatives

## Module 6: Analyzing React Applications

### Developer tools

- Inspecting React components with browser extensions
- Identifying and debugging rendering errors

### Unit testing

- Running tests with Jest
- Isolating code under test with Enzyme

### Performance testing

- Establishing a baseline
- Analyzing performance with Flamegraphs
- Optimizing component reconciliation with key

## Module 7: Advanced Features

- Optimizing runtime performance with shouldComponentUpdate
- Extending your components with Higher Order Components