

Introduction to React

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 popular, high-performance JavaScript library for building rapidly responsive user interfaces.

React is more than just a library for building user interfaces. It's a new way to think about building web applications. Once you understand the principles of it, it's surprisingly simple. One of the great things about react, and about ES6 too, is that you don't have to buy into it completely. You can start out using it quickly by integrating it with your existing code, and then you can do more with it as you become more proficient. React's architecture makes it easy to reason about and easy to gradually refactor.

Introduction to React teaches experienced JavaScript developers the skills they need to immediately implement React in their applications.

Objectives

After taking this course, students will be able to:

- Use Create-React-App to get started quickly with React
- Learn to write unit tests for React, using Jest and Enzyme
- Understand what React is and what problem it solves
- Explore the basic architecture of a React application
- Gain a deep knowledge of React components and JSX
- Build a working application that uses React components
- Use Redux for maintaining state in a React.js application
- Use Redux middleware
- Make AJAX requests with React
- Use server-side rendering
- Learn React best practices

Topics

- Introduction
- Advanced JavaScript
- Introduction to React.js
- React Development Process
- React Component Best Practices
- Forms
- Component Life-Cycle
- Testing React Components
- Flux and Redux
- Advanced Topics

Audience

This course is designed for experienced JavaScript developers.

Prerequisites

There are no prerequisites for this course.

Duration

Three days

Introduction to React

Course Outline

I. Introduction

- A. What is React.js?
- B. What is React NOT?
- C. When can you use React?
- D. Who Uses React?
- E. React Quick Start
- F. What is Create React App

Lab: Get Started with Create React App

II. Advanced JavaScript

- A. Variable Scoping with const and let
- B. let vs. var
- C. Block-scoped Functions
- D. Arrow Functions
- E. Default Parameter Handling
- F. Rest Parameter
- G. Spread Operator
- H. Template Literals
- I. Enhanced Object Properties
- J. Array Matching
- K. Object Matching
- L. Symbol Primitive
- M. User-defined Iterators
- N. For-Of Operator
- O. Creating and Consuming Generator Functions
- P. Class Definition
 - 1. Class Declaration
 - 2. Class Expressions
 - 3. Class Inheritance
- Q. Understanding this
 - 1. What is this?
 - 2. Implicit Binding
 - 3. Explicit Binding
 - 4. new Binding
 - 5. window Binding
- R. Array.map()
- S. Promises
 - 1. What Are Promises?
 - 2. Promises vs. Event Listeners
 - 3. Why Use Promises?
 - 4. Demo: Callback vs. Promise
 - 5. Using Promises

III. Introduction to React.js

- A. Imperative API vs. Declarative API
- B. Imperative vs. Declarative Screen Updates

Lab: Your First Component

- C. One-way Data Flow
- D. Virtual DOM
- E. Virtual DOM vs. HTML DOM
- F. State Machines
- G. React.render()

Lab: Create More Components

- H. ReactDOM
 - 1. ReactDOM.findDOMNode
 - 2. ReactDOM.unmountComponentAtNode
 - 3. ReactDOM.render

Lab: Testing React

IV. React Development Process

- A. What is JSX?
- B. Using JSX
- C. Using React with JSX
- D. React.createElement
- E. Using React without JSX
- F. Expressions in JSX

V. React Component Best Practices

- A. Single Responsibility
- B. Pure Functions
- C. Benefits of Pure Functions
- D. Function Comparison
- E. Stateless Functional Components

Lab: Static Version

- F. Styles in React

Lab: Styling React

- G. Composition
 - 1. Reusable Components
 - 2. Container Components
 - 3. Presentational Components
 - 4. props

Lab: Props and Containers

- H. Props vs. State
- I. Setting Initial State
- J. super()

Lab: Adding State

Introduction to React

Course Outline (cont'd)

VI. Forms

- A. Form Events
- B. Events
- C. Controlled Components
- D. Uncontrolled Components

Lab: Interactions, Events, Callbacks

VII. Component Life-Cycle

- A. Life-Cycle Methods
 - 1. Mount/Unmount Life-Cycle Methods
 - 2. Data Life-Cycle Methods
 - 3. Component Life Cycle

Lab: Component Life-Cycle

VIII. Testing React Components

- A. What to Test in a React Component
- B. PropTypes
 - 1. Using PropTypes
- Lab: PropTypes**
- C. Jest
 - 1. Mocking
 - 2. Mock Function
 - 3. Manual Mock
 - 4. Automocking
 - 5. Snapshot Testing
- D. TestUtils
- E. Enzyme
- F. Shallow Rendering

Lab: Testing with Jest and Enzyme

IX. Flux and Redux

- A. Flux
 - 1. Flux Flow
 - 2. Flux Action
 - 3. Flux Dispatcher
 - 4. Flux Stores
 - 5. EventEmitter

B. Redux

- 1. Stores & Immutable State Tree
- 2. Redux Actions
- 3. Reducers
- 4. Things You Should Never do in a Reducer
- 5. Reducer Composition
- 6. Redux Store
- 7. Redux Pros and Cons

Lab: Implementing Redux

- C. Redux Middleware
 - 1. What is Redux Middleware?
 - 2. What is Middleware Good For?
 - 3. React AJAX Best Practices
 - 4. Using React with Other Libraries
 - 5. Redux Thunk
 - a) How is Thunk Useful?
 - b) How does Thunk work?
 - c) Lab 14: Thunk
 - 6. Redux Saga
 - a) Using Sagas

X. Advanced Topics

- A. React Router
- B. Server-side React
- C. Relay and GraphQL
 - 1. What is Relay?
 - 2. GraphQL
 - 3. GraphQL Example
 - 4. Relay Pros and Cons
- D. Performance Optimization
 - 1. Development vs. Production
 - 2. Perf Object
 - 3. Optimization Techniques