



React JavaScript Programming

Course ID #: WA2583-ZZ-W

Hours: 21

Delivery Method: Instructor Led Live & Group Internet Based

Course Content

Course Description:

This React Programming in 2021 Training class introduces the React JavaScript framework, sometimes referred to as ReactJS or React.JS.

This React Programming class will provide an introduction to the benefits of the React JavaScript framework, so participants in this React Training course can start to develop applications quickly using the framework.

React is a JavaScript library that makes building user interfaces effortlessly simple.

It was developed by Facebook, which maintains it along with a community of developers. Since User Interfaces are used in almost every web application, the ability to develop UIs more efficiently is crucial. React does exactly this by providing a developer-friendly way to create custom UIs.

Course Objectives:

In this React Training class, attendees will learn how to:

- Understand the programming model provided by the React framework
- Define React components
- Use the React framework to handle events and stateful data

Target Audience:

- .NET Developers with a minimum of 2 years of experience and individuals who want to become application security engineers/analysts/testers.
- Individuals involved in the role of developing, testing, managing, or protecting wide area of applications.

Prerequisites:

Minimum of two years' experience in .NET and Java



React JavaScript Programming

Course ID #: WA2583-ZZ-W

Hours: 21

Delivery Method: Instructor Led Live & Group Internet Based

Topics:

Chapter 1. React Overview

- What is React?
- What's in a Name?
- React Component Model
- What React Is Not
- What You Will Not Find in React
- Motivation for Creating React
- A React JavaScript Example
- One-Way Data Flow
- JSX
- A JSX Example
- The Virtual (Mock) DOM
- Only Sub-components that Actually Change are Re-Rendered
- React Libraries
- Summary

Chapter 2. Babel Command-Line Interface

- Babel Transpiler
- Usage Options
- Presets and Plug-ins
- Babel CLI Installation
- Babel Configuration
- Running Babel Command-Line
- A Basic ES6 Development Setup with Babel
- Test the Babel Development Setup
- Adding React to the Development Setup
- Create a Minimal React App – index.html
- Create a Minimal React App – app.js
- Summary

Chapter 3. Basic Components and JSX

- What is JSX?
- JSX Transpilation to React Code Example
- Running the Transpiled Code
- Babel
- The Babel Runtime JavaScript Library
- Script Import Skeleton Code
- Playing Around in Code Pen
- React Components
- Ways to Create UI Components
- Creating a Functional Component Example
- Component Names Must Be Capitalized
- Creating a React Class-Based Component in ES5
- The render Method
- Creating a UI Component Using ES6 Class

Notation

- Using ES6 Classes with React
- Which UI Component Creation Syntax Should I Use?
- Components vs Elements
- Elements Are Immutable
- Properties
- Property Naming Convention
- Properties Default to 'True'
- Spread Attributes (an ES6 Feature)
- Expressions
- Fragments
- Summary

Chapter 4. React Functional Component Concepts

- Functional Components
- Nesting JSX Elements
- Example of JSX Nesting
- Comments in JSX Code
- Setting CSS Styles Using Classes
- Setting CSS Styles Directly
- JSX Escapes Values
- Working with Lists of Items
- Keys in Lists
- Example List with Key
- Container vs. Presentational Components
- State
- Types of State Data
- State Hierarchy
- Lifting State Up
- Props vs. State
- Pass Down a Function
- Immutability
- Immutability – Why?
- Virtual DOM and State
- Setting state
- Updating Input fields
- Passing Props to Components
- Passing Functions to Components
- Event Handling
- Event Handler Example
- Event Binding – DOs
- Event Binding – Don'ts
- Passing Parameters to Event Handlers
- Component Life-cycle



React JavaScript Programming

Course ID #: WA2583-ZZ-W

Hours: 21

Delivery Method: Instructor Led Live & Group Internet Based

- Life-cycle in Functional Components
- App Development Workflow – 1/3
- App Development Workflow – 2/3
- App Development Workflow – 3/3
- Summary

Chapter 5. React Components with ES6 Classes

- Classes in ES6
- Functional Components
- Extending React Component
- The render Method
- Component Lifecycle
- Component Lifecycle: Overview
- Component Lifecycle – Render Phase
- Component Lifecycle – Commit Phase
- Component Lifecycle – Unmounting
- Summary

Chapter 6. React Router

- Routing and Navigation
- Creating a react-router based project
- A Basic Routed Component
- Router vs. Browser Router
- The Route component
-
- Redirect Route
- Navigating with
- Navigating with
- Route Parameters
- Retrieving Route Parameters
- Query String Parameters
- Using Router with Redux
- Summary

Chapter 7. State Management for React

- React State Basics – Props and State
- Props
- State in Class Based Components
- Managing State with Hooks in Functional Components
- The Problem with Props and State
- Redux State Library
- Redux Advantages
- Redux Disadvantages
- Basic Rules for State Management
- Types of State
- Data State

- Communication State
- Control State
- Session State
- Location State
- Location State Side Effects
- Summary

Chapter 8. Building React Apps with Redux

- Redux
- Redux Terminology
- Redux Principles
- Redux: Actions
- Redux Action Types
- Action Creators
- Dispatching Actions
- Data Flow Basics
- Redux Reducers
- Pure Functions
- Reducer Example
- Returning Default State
- Creating a Development Environment with create-react-app.
- Using Redux with React
- Initializing the Store
- Immutability
- Benefits of Immutable State
- Mutability of Standard types
- Copying Objects in JavaScript
- Copying Arrays in JavaScript
- One Store – Multiple Reducers
- Combining Reducers
- Components and Redux
- The React-Redux Package
- Wrapping App with Provider
- Map State to Props
- map Dispatch to Props
- Using Mapped Properties and Methods
- Wrapping Components with Connect
- Configure Store
- Programming Advice – Multi Tab Console
- Summary

Chapter 9. Using React Hooks

- Functional Component Shortcomings
- Hooks Overview
- Hook Rules



React JavaScript Programming

Course ID #: WA2583-ZZ-W

Hours: 21

Delivery Method: Instructor Led Live & Group Internet Based

- React Linter Example
- Functional Component Props
- The use State Hook
- Functional Component using the use State hook.
- Use State with Multiple Variables
- use State can also be used with Objects.
- The use Effect Hook
- Use Effect Hook Example
- Using use Effect Hook to Load Data
- Restricting when use Effect is Called.
- The use Context Hook
- Additional Hooks
- The use Reducer Hook
- An Example Reducer Function
- Calling and Using use Reducer
- The use Memo Hook
- Use Memo Example
- The use Callback Hook
- Use Callback Example
- The use Ref Hook
- Using use Ref to Hold Values
- The use Imperative Handle Hook
- Use Imperative Handle Hook Example
- The use Layout Effect Hook
- Summary

Chapter 10. Creating Custom React Hooks

- Custom Hooks
- Custom Message Hook
- Using the Custom Message Hook
- A Custom use List Hook
- Using the use List Custom Hook
- The built-in use Debug Value Hook
- Viewing the Effect of the use Debug Value Hook
- Summary

Chapter 11. Unit Testing React with React

Testing Library

- React Testing Framework
- Features
- Snapshot Testing
- Code Coverage
- Interactive Mode
- Projects created with create-react-app

- Default App Component Test
- Unit Tests
- Anatomy of a Unit Test
- Common Matchers
- Combining Tests
- Running Tests
- Testing Promise based async code with 'done'
- Setup and Teardown
- A Simple Component Test
- A Simple Snapshot Test
- Running and Updating Snapshot Tests
- Building Component Tests
- Calling Render
- Render Properties
- Simulating Events
- Testing Results
- Using Query Functions
- Text Matching
- Counter Component
- Summary

Chapter 12. [OPTIONAL] Exception Handling in JavaScript

- Exception Handling
- Try Syntax
- The Final Block
- The Nested Try Blocks
- Exception Types in JavaScript
- The Throw Statement
- Using the Error Object
- Summary

Chapter 13. [OPTIONAL] Web Storage, Web SQL, and IndexedDB

- Data Storage
- Data Storage Options
- Web Storage
- Web Storage Programming Interface
- Web Storage Examples
- Storing JavaScript Objects
- IndexedDB
- Indexed DB Terminology
- Getting indexed DB Objects
- Opening a Database
- Creating an Object Store



React JavaScript Programming

Course ID #: WA2583-ZZ-W

Hours: 21

Delivery Method: Instructor Led Live & Group Internet Based

- Inserting a Record
- Retrieving a Record
- Summary

Chapter 14. [OPTIONAL] Asynchronous Programming with Promises

- The Problems with Callbacks
- Introduction to Promises.
- Requirements for Using Promises
- Creating Promises Manually
- Calling the Promise-based Function
- Making APIs that support both callbacks and promises.
- Using APIs that support both callbacks and promises.
- Chaining then Method / Returning a Value or a Promise from then Method.
- Promisifying Callbacks with Bluebird
- Using Bluebird
- Bluebird – List of Useful Functions
- Benefit of using Bluebird over ES6 for Promisification

- Error Handling in Promise-based asynchronous functions
- Summary

Lab Exercises

- Lab 1. Setting Up a React Development Environment
- Lab 2. Basic React Components
- Lab 3. More React Component Concepts
- Lab 4. ES6 React Components
- Lab 5. React Router Application
- Lab 6. React Redux Application
- Lab 7. React Hooks Application
- Lab 8. React Custom Hooks
- Lab 9. Unit Testing with Jest
- Lab 10. Course Project – React App Using Hooks
- Lab 11. [OPTIONAL] Exception Handling in JavaScript
- Lab 12. [OPTIONAL] Asynchronous Programming with Promises



React JavaScript Programming

Course ID #: WA2583-ZZ-W

Hours: 21

Delivery Method: Instructor Led Live & Group Internet Based

Register for this class by visiting us
at:
www.tcworkshop.com or calling
us at 800-639-3535

normal course content with the NASBA
Information added along with links to our
policy page on the web. We will add our
name to the Official National Registry
Statement after we are approved.

Policies: Course Registration, Cancellation, Refund and Complaint Resolution

: For more information regarding
administrative policies such as complaint
and refund, please contact our offices at 800-
639-3535 or visit us at:
www.tcworkshop.com

NASBA Information:

Level: Intermediate

Attendance Requirement: To be awarded the full credit
hours, you must sign in for the course and attend the
entire course.

Fields: Computer software & Applications

CPEs: 21

Official National Registry Statement:

The Computer Workshop is registered with
the National Association of State Boards of
Accountancy (NASBA) as a sponsor of
continuing professional education on the
National Registry of CPE Sponsors. State
boards of accountancy have final authority
on the acceptance of individual courses for
CPE credits. Complaints regarding registered
sponsors may be submitted to the National
Registry of CPE Sponsors through its
website: www.nasbaregistry.org

NOTE: Since our information is in multiple
places on our web site or in PDF format that
is sent to clients, we have provided our