

Developing Applications using React

Duration: 4 Days

Description

This is three-day, introductory-level course designed for students who might be newer to using modern web development skills and standards. This course provides students with the hands-on skills they require to implement the Reactive paradigm to build stateless and asynchronous apps with React.

The course guides students through a detailed ReactJS and its evolution, and then teaches them the practical skills required to build a simple React component. Then students will build the same react component with JSX syntax to demystify its usage. Students will learn how to configure the Facebook Graph API, get their 'likes' list, and render it using React.

Next, students will break the UI into components and learn how to establish communication between them and respond to users input/events in order to have the UI reflect their state. Students will also learn core ES6 syntaxes in the process. Then the course will delve into the FLUX and its architecture, which is used to build client-side web applications and complements React's composable view components by utilizing a unidirectional data flow.

Towards the end of the program, students will learn how to make their components reusable, and how to test and deploy them into a production environment.

Finally, the course will touch on other 'next-level' topics such as React on the server side, Redux and other related, useful advanced concepts.

Delegates will learn to:

- Understand the ReactJS basics through an overview
- Install and create your first React component
- Refactor the ReactJS component using JSX
- Integrate your React application with the Facebook login and Graph API, then fetch data from your liked pages in Facebook and display them in a browser
- Handle UI elements events with React, respond to users input, and create stateful components
- Use some core lifecycle events for integration and find out about ES6 syntaxes in the React world
- Understand the FLUX architecture and create an application using FLUX with React
- Make a component more reusable with mixins and validation helpers and structure

your components properly

- Explore techniques to test your ReactJS code
- Deploy your code using webpack and Gulp

Audience

This is an introductory (gentle intro-level) React development course, designed for basic-level web developers that need to further extend their skills in modern web development. This course is targeted for newer web developers with at least basic level experience with HTML5, CSS3, and JavaScript

Outline

Getting Started with ReactJS

- Introducing ReactJS
- Downloading ReactJS
- Tools
- Trying ReactJS

Exploring JSX and the ReactJS Anatomy

- What is JSX?
- Why JSX?

Working with Properties

- Component properties
- Configuring Facebook Open-Graph API
- Rendering data in a ReactJS component

Stateful Components and Events

- Properties versus states in ReactJS
- Exploring the state property
- Events

Component Life cycle and Newer ECMAScript in React

- React component lifecycle
- Other ES (ECMAScript) versions in React

Reacting with Flux

- An overview of Flux

- Flux versus the MVC architecture
- Actions
- Dispatchers
- Stores
- Controller-Views and Views
- Revisiting the code

Making Your Component Reusable

- Understanding Mixins
- Validations
- The structure of component
- Summary

Testing React Components

- Testing in JavaScript using Chai and Mocha
- Testing using ReactTestUtils
- The jestTypical example of a Testsuite with Mocha, expect, ReactTestUtils and Babel
- Testing with shallow rendering

Preparing Your Code for Deployment

- An introduction to Webpack
- Advantages of Webpack
- Introduction to Gulp

What's Next

- AJAX in React
- React Router
- Server-side rendering
- Isomorphic applications
- Hot reloading
- Redux React
- Relay and GraphQL
- React Native
- References

Prerequisites

Knowledge of JavaScript equivalent to attending the JavaScript Programming course. Attendees should also have familiarity with a Windows, Linux, or OS X command line interface as well as a basic understanding of network protocols such as HTTP.