



# React JS

Duration 3 day(s) (REACT-03)

Develop single-page web application with React and Redux

## Description

React is a JavaScript library that was created by the Facebook and Instagram developer teams. Their goal was to solve some issues in performance, productivity, and maintainability with their existing JavaScript applications. React has now become a powerful and popular framework among JavaScript developers.

## Goals

- Understand React's architecture and mindset
- Understand the Flux model, Redux & component
- Build a Single-Page application using React

### *Public*

- Chief Technology Officer
- Creative Designer
- Development Lead
- Full Stack Developer
- Frontend Developer
- JavaScript Developer
- Software Engineer
- UI Developer
- Web developer

### *Prerequisites*

- Basic HTML & CSS3
- Working knowledge of JavaScript language
- Knowledge of web technologies

### *Structure*

40% Theory, 60% Practice

## Program

# Introduction

- Overview of the JavaScript ecosystem
- NodeJS and NPM fundamentals
- ES6 Primer

# Tooling

- JavaScript Transpiler
- Bundlers
- Webpack
- create-react-app

# React

- A brief history of React
- Core concepts behind React
- React under the hood: Virtual DOM, reconciliation
- Packaging: Addons, react-tools, JSXTransformer, React Developer Tools

# Rendering React components

- First React component
- Work with React.createElement
- DOM rendering
- JSX expressions
- Lab: Creating your first application using React components

# React component states

- Internal state of a component (React State)
- Read state and set state
- Pass data to component using props
- Validating props using propTypes
- Lab: Creating application to read state, set state and pass data to component

# Lifecycle of React components

- Main stages of the component life cycle (i.e. Mounting a Component)
- General methods in each stage of the component life cycle
- Error handling method in component life cycle

# Tests

- Test runner
- Test Driven Development
- Test a function
- Create Mock Functions
- Test a React component
- Lab: Creating the application unit tests with Jest

# Redux

- The unidirectional workflow
- Flux pattern
- The main components of Redux
- Redux in the React context
- Lab: Implementation of Redux store and unit tests

## REST architecture

- Introduction to REST Architecture
- React & REST Architecture
- Redux - redux-thunk
- Interactions between a React application and a REST Server
- Lab: Creating REST API to retrieve data

## Routing

- The react-router package
- Nested routes
- React Router props
- Link components
- URL handling
- Lab: Implementation of Single-Page Application with React-router

## Form

- Validation of a React form
- Redux-form
- Lab: Creation and validation of a Redux-form with React and unit tests

## Performance

- Reducing reconciliations with shouldComponentUpdate
- Using PureComponent
- Performance measurement addon
- Lab: Analysing and improving application performance with Chrome Dev tools

## Isomorphism

- Single Page Application
- Isomorphic React
- Isomorphic and data
- Isomorphic and redux

## Go further

- Lodash
- Internationalisation with react-intl
- React-native

## Conclusion