Rumos valorizamos profissionais!

Web Development with React.js

- **Formato do curso:** Presencial e Live Training
- **Localidade:** Lisboa
- **Data:** 20 Jun. 2022 a 22 Jun. 2022
- **Preço:** 1050€
- **Horário:** Laboral - das 09h00 às 17h00
- **Duração:** 21 horas

This course covers the basics of React.js and prepares the student to start developing web applications with the library.

Destinatários

This course targets Web Developers who are looking for ways to improve their productivity with React.

Pré-requisitos

This course targets professional web developers who are familiar with JavaScript (ES5, ES2015+) and HTML.

Programa

Introduction to React.js

- What is React.js?
- Why React.js?
- DOM Deep Dive
- Real DOM
- Virtual DOM
- React Architecture
- Comments
- React.js Applications
- React Browser apps
- React Native apps
- React set up

- Writing React Hello World Program

Break the UI into a components hierarchy

- What is UI?
- Compose UI
- UI as Component
- What is Component?
- Why Component?
- Understand Component hierarchy.

React and imperative Programming

- What is Imperative Programming?
- Writing Imperative DOM Programming
- Writing React Imperative DOM programming using api

Basic REACT API

- React class
- React class is as entry point to React lib
- React.createClass Overview
- React.createClass(Object Speciation) detailed understanding
- React.createElement class Overview and detailed discussion
- React.cloneElement class
- React.createFactory class
- React.isValidElement
- React.DOM
- React.PropTypes
- React.children
- React.Children.forEach
- React.Children.count
- React.Children.toArray

REACT DOM package and its API

- ReactDOM.render detailed understanding
- ReactDOM.unmountComponentAtNode
- ReactDOM.findDOMNode
- React.Component
- setState
- replaceState
- forceUpdate
- isMounted
- setProps
- replaceProps

React & Declarative Programming using JSX

- What is JSX?
- Why JSX?
- Declarative Programming and imperative programming
- JSX and Browsers
- Transforming JSX into native JavaScript code using “Transpilers”
- JSX and Various Transpilers
- JSX vs Template literals
- JSX to express UI Components
- JSX Syntax and grammar
- HTML Tags Vs React Components

Components and Expression

- Component Namespace
- JavaScript Expression
- Attribute Expressions
- Boolean Attributes
- Child Expression

Components and State, Properties, Events

- What is State?
- How State Works
- What Components Should Have State?
- What Should Go in State?
- What Shouldn’t Go in State? Using State
- Data is State Container
- Transferring Pros
- Manual Transfer
- Transferring with JSX
- Consuming and Transferring the Same Prop
- Rest and Spread Properties
- Handling Events
- Events in depth
- Event Handling and Synthetic Events
- Under the Hood: Autobinding and Event Delegation

Parent/Child Relationships

- Multiple Components
- Motivation: Separation of Concerns
- Composition Example
- Ownership
- Children
- Child Reconciliation

- Stateful Children
- Dynamic Children
- Using refs
- Prop Types
- Adding multiple child elements
- A deeper look at keys

Reusable Components

- How to develop reusable Components
- Prop Validation
- Single Child
- Default Prop Values
- Mixins

Forms

- Forms and its elements in React
- Why Controlled Components?
- Why Textarea Value?
- Imperative operations

Component Lifecycle

- Mounting
- Updating
- Unmounting

Add-ons

- Animation
- Two way data binding helpers
- Cloning Elements
- Keyed Elements
- Immutability Helpers
- Shallow Compare

Server-side Integration with Ajax

- AJAX and React.js
- React and its Ajax libs
- superagent: A lightweight “isomorphic” library for AJAX requests.
- Request: AJAX all over again. Includes support for xml HttpRequest, JSONP, CORS, and CommonJS Promises A. Browser support stretches back to IE6.
- react-ajax: Ajax Request Component for React.
- axios: Promise based HTTP client for the browser and node.js.
- request: Simplified HTTP request client.
- Load Initial Data via AJAX

- CURD operations

Managing App State with Redux

- What is Redux?
- Reducers
- Containers – Connecting Redux to React
- Implementation of a Container Class
- Containers and Reducers Overview
- Actions and Action Creators
- Binding Action Creators
- Creating an Action
- Consuming Actions in Reducers

Intermediate Redux Middleware

- App Overview and Planning
- Component Setup
- Controlled Components and Binding Context
- Form Elements in React
- Working with API’s
- Introduction to Middleware
- Ajax Requests with Axios
- Redux-Promise in Practice
- Avoiding State Mutations in Reducers
- Building a List Container
- Mapping Props to a Render Helper
- Adding Sparkline Charts

React Router + Redux Form

- App Overview and Goals
- Exploring the Posts Api
- Installing React Router
- React Router – What is It?
- Setting Up React Router
- Route Configuration
- Nesting of Routes
- IndexRoutes with React Router
- Back to Redux – Index Action
- Catching Data with Posts Reducer
- Fetching Data with Lifecycle Methods
- Creating New Posts
- Navigation with the Link Component
- Forms and Form Submit