Advanced Node and React: Building a Secured SPA using REST APIs

Course Summary

Description

Best practices will be emphasized, and real-world techniques employed as students create an actual end-to-end SPA (Single Page Application). React 16 with JSX is used to create a front-end using a supplied back end environment. Then the back-end is replaced by creating a REST API using Node to access Postgres. Client and server-side routing, validation of data, security, authentication, and tools to enhance productivity will be emphasized throughout. Students will learn and use testing strategies and tools for each layer of the application.

Objectives

After taking this course, students will be able to:

- Setup projects for reproducible builds and testing leveraging modern tools
- Understand and implement React Concepts and performance enhancement options
- Build a secure Rest-API using Node.js and Express
- Simple and complex queries using Knex and using migrations
- Front and Back End validation with Joi
- The Flux design pattern and handling state with Redux
- Leverage Express.js middleware for authorization and security
- Create and execute tests against React front-end and Node back-end
- Practice with various testing tools including: Jest, Enzyme, supertest, Mocha, Chai, Sinon

Topics

- A Basic App with Simple State
- Creating Rest API in Node
- Forms and Validation
- Understanding Flux and Redux
- Enhancing Security

Audience

This course is designed for experienced JavaScript developers who want to understand the inner workings of React and Node, to be able to build full applications with React and Node.

Prerequisites

Before taking this course, you should have the following:

- Knowledge of ES6 features including the concept of modules / packages
- Experience with Node and npm
- Exposure to React development

Duration

Five days

ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Advanced Node and React: Building a Secured SPA using REST APIs

Course Outline

I. A Basic App with Simple State

A. Initial Project Setup
1. Version control with Git, download course solutions and demos
2. Latest helpful extensions in VSCode
3. Creating a project from scratch using npm init
4. Managing modules with npm
5. Using npx and other newer approaches with Node
6. Prepare reproducible builds with npm scripts
7. Configuring ESLint
8. Testing Concepts with Examples
   a) Unit Testing React using Jest and Enzyme
   b) Unit Testing Node using Mocha and Chai

B. Setting up a Simple Version of our Project
1. Current state of React development
2. Using build tools and Running React projects
3. Configuring Webpack
4. Debugging in VS Code
5. Installing and using React Developer Tools
6. Preparing JSON data and Working with JSON-server
7. Testing Rest APIs using Postman

C. Fundamental React concepts
1. Exploring app created with Create React App (CRA)
2. Virtual DOM
3. React and React.DOM.*
4. Special DOM attributes
5. Basics of components
6. Why and how of JSX

D. React Development
1. Rendering Elements
2. Using built-in components
3. Nesting components
4. Hierarchies of React components
5. Communicating between components
6. Outputting lists and conditional content
7. When to use “props” and when to use “state”
8. Styling components

E. Component’s State and Lifecycle
1. Using constructors
2. Component's lifecycle methods
3. Mounting methods
4. Updating methods
5. Unmounting methods
6. Validating against expected properties
7. Setting defaults
8. Leveraging conditional rendering
9. Lists and Keys

F. Event Handling in React
1. Adjusting syntax for React events and JSX
2. The Synthetic events system
3. Attaching Event Handlers
4. Passing Arguments to Event Handlers
5. Rendering Based on State
6. Updating State

G. Asynchronous HTTP Requests & Routing
1. Creating routes
2. Navigating between “pages”
3. Nesting Routes
4. Capturing Path Params
5. Route Prefixing
6. Defining the root of your app

H. Testing with Jest and Enzyme
1. Adding and using Jest in our projects
2. Testing React components

II. Creating Rest API in Node

A. Getting Started with the Express Module
1. Modifying project created with Express App Generator
2. Leveraging middleware for improved error handling and logging
3. Creating Routing files for the apps GET/POST/PUT/DELETE
4. Automating testing of routes with Mocha and Chai
5. Using nock for HTTP mocking and expectations library
6. Using supertest for integration testing

Advanced Node and React: Building a Secured SPA using REST APIs

Course Outline (cont'd)

B. Setting up migrations, knex, and bookshelf for database access
   1. Overview and setup of course database
   2. Creating a migration for database tables and data
   3. Simple and more complex queries using of Knex
   4. Testing and mocking the database

III. Forms and Validation

A. Forms
   1. Differences between Controlled and Uncontrolled components
   2. JSX and Forms
   3. Using form events
   4. Controlling data input with Controlled form components
   5. How React.js changes the interface of some form components
   6. The importance of using the name attribute
   7. Creating custom form components
   8. Dealing with multiple Controlled form components
   9. Leveraging control of focus
   10. Best practices for creating usable forms

B. Handling User Input and Requests
   1. Considerations with body-parser module
   2. Server-side caching
   3. Adding in Validation with Joi
   4. Validate user input on backend
   5. Validate user input on frontend

IV. Understanding Flux and Redux

A. Understanding Flux and Redux
   1. Overview of the Flux design pattern
   2. Why Redux is easier, better, and endorsed by the creator of Flux
   3. What is state and what are examples where it influences the screen view
   4. Three principles of Redux: single source, read only state, only pure functions
   5. Examples of using Redux
   6. Defining actions and using action creators
   7. Specify state changes to actions with Reducers
   8. Setting up the Store
   9. Dispatching Actions
   10. Using Immutable.JS

V. Enhancing Security

A. Session Management in Node
   1. Allowing User Registration with hashed password
   2. Authenticating Login with database
   3. Forcing logins with route-level middleware (requireLogin)
   4. Logging out to end the Session
   5. Sessions + express-session usage
   6. Returning auth cookie
   7. Using passport with local strategy

B. Adding authentication to our application
   1. Understanding authentication practices
   2. Setting up the Store and login actions
   3. Creating a reducer to return states
   4. Using dispatch for the login and logout components

C. More Security
   1. Understanding the components of helmet
   2. Continuous Security Monitoring
   3. HTTPS

D. Next Steps and Considerations
   1. Deployment strategies
   2. Production practices
   3. NextJS