KOENIG
step forward

Modern React with Redux

Let's Dive In!

- Installing Node JS
- Important Update About React App Generation
- Generating a React Project
- Why Create React App?
- Exploring a Create-React-App Project
- Starting and Stopping a React App
- JavaScript Module Systems
- Important Note about Live Reloading
- Displaying Content with Functional Components

Building Content with JSX

- What is JSX?
- Converting HTML to JSX
- Inline Styling with JSX
- Converting Styling to JSX Format
- Class vs ClassName
- Referencing JS Variables in JSX
- Values JSX Can't Show
- Finding Forbidden Property Names

Communicating with Props

- Three Tenets of Components
- Application Overview
- Semantic UI CDN Link
- Getting Some Free Styling
- Naive Component Approach
- Specifying Images in JSX
- Duplicating a Single Component
- Extracting JSX to New Components
- Component Nesting
- React's Props System
- Passing and Receiving Props
- Passing Multiple Props
- Passing Props - Solutions
- Component Reuse
- Implementing an Approval Card
- Showing Custom Children
- Component Reuse

Structuring Apps with Class-Based Components

- Class-Based Components
- Application Overview
- Scaffolding the App
- Getting a Users Physical Location
- Resetting Geolocation Preferences
- Handling Async Operations with Functional Components
- Refactoring from Functional to Class Components

State in React Components

- The Rules of State
- Important Note About super(props) Deprecation
- Initializing State Through Constructors
- Updating State Properties
- App Lifecycle Walkthrough
- Handling Errors Gracefully

Understanding Lifecycle Methods

- Introducing Lifecycle Methods
- Why Lifecycle Methods?
- Refactoring Data Loading to Lifecycle Methods
- Alternate State Initialization
- Passing State as Props
- Determining Season
- Ternary Expressions in JSX
- Icons Not Loading and CORS errors
- Showing Icons
- Extracting Options to Config Objects
- Adding Some Styling
- Showing a Loading Spinner
- Specifying Default Props
- Avoiding Conditionals in Render
- Class-Based Components
- Updating Components with State
- Updating Components with State
- Conditionally Rendering Content

Handling User Input with Forms and Events

- App Overview
- Component Design
- Adding Some Project Structure
- Showing Forms to the User
- Adding a Touch of Style

- Creating Event Handlers
- Alternate Event Handler Syntax
- Uncontrolled vs Controlled Elements
- More on Controlled Elements
- Exercise Overview - Receiving Values
- Receiving Values From Controlled Elements
- Exercise Solution
- Handling Form Submittal
- Understanding 'this' In Javascript
- Solving Context Issues
- Communicating Child to Parent
- Invoking Callbacks in Children

Making API Requests with React

- Fetching Data
- Axios vs Fetch
- Viewing Request Results
- Handling Requests with Async Await
- Setting State After Async Requests
- Binding Callbacks
- Creating Custom Clients

Building Lists of Records

- Rendering Lists
- Review of Map Statements
- Rendering Lists of Components
- The Purpose of Keys in Lists
- Implementing Keys in Lists
- Exercise Overview - List Building
- Practicing List Building
- Exercise Solution

Using Ref's for DOM Access

- Grid CSS
- Issues with Grid CSS
- Creating an Image Card Component
- Accessing the DOM with Refs
- Accessing Image Height
- Callbacks on Image Load
- Dynamic Spans
- App Review

Let's Test Your React Mastery!

- App Overview
- Component Design
- Scaffolding the App
- Reminder on Event Handlers
- Handling Form Submittal
- Accessing the Youtube API
- Searching for Videos
- Adding a Video Type
- Putting it All Together
- Updating State with Fetched Data
- Passing State as Props
- Rendering a List of Videos
- Rendering Video Thumbnails
- Styling a List
- Communicating from Child to Parent
- Deeply Nested Callbacks
- Conditional Rendering
- Styling the VideoDetail
- Displaying a Video Player
- Fixing a Few Warnings
- Defaulting Video Selection

Understanding Hooks in React

- React Hooks
- Important Note
- App Architecture
- Communicating the Items Prop
- Building and Styling the Accordion
- Helper Functions in Function Components
- Introducing useState
- Understanding useState
- Setter Functions
- Expanding the Accordion
- Exercise Overview
- UseState Exercise
- Exercise Solution
- Creating Additional Widgets
- The Search Widget Architecture
- Scaffolding the Widget
- Text Inputs with Hooks
- When do we Search?
- The useEffect Hook
- Testing Execution

- When Does It Run?
- Async Code in useEffect
- Executing the Request from useEffect
- Default Search Terms
- List Building!
- XSS Attacks in React
- XSS Server Code
- Linking to a Wikipedia Page
- Only Search with a Term
- Throttling API Requests
- Reminder on setTimeout
- useEffect's Cleanup Function
- Implementing a Delayed Request
- Searching on Initial Render
- Edge Case When Clearing Out Input Form
- Optional Video - Fixing a Warning
- Dropdown Architecture
- Scaffolding the Dropdown
- A Lot of JSX
- Selection State
- Filtering the Option List
- Hiding and Showing the Option List
- Err... Why is this Hard?
- Reminder on Event Bubbling
- Applying What We've Learned
- React v17 Update - capture: true
- Binding an Event Handler
- Why Stay Open!?
- Which Element Was Clicked?
- Making use of useRef
- Important Update for Event Listeners
- Body Event Listener Cleanup
- The Translate Widget
- Scaffolding the Translate Component
- Adding the Language Input
- Understanding the Convert Component
- Google Translate API Key
- Building the Convert Component
- Using the Google Translate API
- Displaying Translated Text
- Debouncing Translation Updates
- Reviewing UseState and UseEffect
- Practicing With UseState and UseEffect
- Exercise Solution

Navigation from Scratch

- Navigation in React
- Basic Component Routing
- Building a Reusable Route Component
- Implementing a Header for Navigation
- Handling Navigation
- Building a Link
- Changing the URL
- Detecting Navigation
- Updating the Route
- Handling Command Clicks

Hooks in Practice

- Project Overview
- Refactoring the SearchBar
- Refactoring the App
- Removing a Callback
- Overview on Custom Hooks
- Process for Building Custom Hooks
- Extracting Video Logic
- Using the Custom Hook
- Exercise Overview - Custom Hooks
- Another Use of Custom Hooks
- Exercise Solution

Deploying a React App

- Deployment Overview
- Deployment with Vercel
- Deployment with Netlify

Redux

- Introduction to Redux
- Redux by Analogy
- A Bit More Analogy
- Finishing the Analogy
- Mapping the Analogy to Redux
- Modeling with Redux
- Creating Reducers
- Rules of Reducers
- Testing Our Example
- Important Redux Notes
- Finished Insurance Policy Code

Integrating React with Redux

- React Cooperating with Redux
- React, Redux, and...React-Redux!?
- Design of the Redux App
- How React-Redux Works
- Redux Project Structure
- Named vs Default Exports
- Building Reducers
- Wiring Up the Provider
- The Connect Function
- Configuring Connect with MapStateToProps
- Building a List with Redux Data
- Extracting More Data From Redux
- Solution to Extracting More Data
- Calling Action Creators from Components
- Redux is Not Magic!
- Functional Components with Connect
- Conditional Rendering
- Connecting Components to Redux

Async Actions with Redux Thunk

- App Overview and Goals
- Initial App Setup
- Tricking Redux with Dummy Reducers
- A Touch More Setup
- How to Fetch Data in a Redux App
- Wiring Up an Action Creator
- Making a Request From an Action Creator
- Understanding Async Action Creators
- More on Async Action Creators
- Middlewares in Redux
- Behind the Scenes of Redux Thunk
- Shortened Syntax with Redux Thunk
- Exercise Solution - Connecting Components to Redux

Redux Store Design

- Rules of Reducers
- Return Values from Reducers
- Argument Values
- Pure Reducers
- Mutations in Javascript
- Equality of Arrays and Objects
- A Misleading Rule
- Safe State Updates in Reducers

- Switch Statements in Reducers
- Adding a Reducer Case
- Reducer Case Solution
- Dispatching Correct Values
- List Building!
- Displaying Users
- Fetching Singular Records
- Displaying the User Header
- Finding Relevant Users
- Extracting Logic to MapStateToProps
- That's the Issue!
- Memoizing Functions
- Memoization Issues
- One Time Memoization
- Alternate Overfetching Solution
- Action Creators in Action Creators!
- Finding Unique User Ids
- Quick Refactor with Chain
- App Wrapup

Navigation with React Router

- App Outline
- Mockups in Detail
- App Challenges
- Initial Setup
- Introducing React Router
- How React Router Works
- How Paths Get Matched
- How to *Not* Navigate with React Router
- Navigating with React Router
- [Optional] - Different Router Types
- Component Scaffolding
- Wiring Up Routes
- Always Visible Components
- Connecting the Header
- Links Inside Routers
- Exercise Overview - Additional Routes
- Adding Another Route
- Exercise Solution

Handling Authentication with React

- OAuth-Based Authentication
- OAuth for Servers vs Browser Apps
- Creating OAuth Credentials
- Wiring Up the Google API Library

- Sending a User Into the OAuth Flow
- Rendering Authentication Status
- Updating Auth State
- Displaying Sign In and Sign Out Buttons
- On-Demand Sign In and Sign Out
- Redux Architecture Design
- Redux Setup
- Connecting Auth with Action Creators
- Building the Auth Reducer
- Handling Auth Status Through Redux
- Fixed Action Types

Redux Dev Tools

- Using Redux Dev Tools to Inspect the Store
- Debug Sessions with Redux Dev Tools
- Recording the User's ID

Handling Forms with Redux Form

- Important Note about Redux Form Installation
- Forms with Redux Form
- Useful Redux Form Examples
- Connecting Redux Form
- Creating Forms
- Automatically Handling Events
- Customizing Form Fields
- Handling Form Submission
- Validation of Form Inputs
- Displaying Validation Messages
- Showing Errors on Touch
- Highlighting Errored Fields

REST-Based React Apps

- Creating Streams
- REST-ful Conventions
- Setting Up an API Server
- Creating Streams Through Action Creators
- Creating a Stream with REST Conventions
- Dispatching Actions After Stream Creation
- Bulk Action Creators
- Object-Based Reducers
- Key Interpolation Syntax
- Handling Fetching, Creating, and Updating
- Deleting Properties with Omit
- Merging Lists of Records

- Fetching a List of All Streams
- Rendering All Streams
- Associating Streams with Users
- Conditionally Showing Edit and Delete
- Linking to Stream Creation
- When to Navigate Users
- History References
- History Object Deprecation Warning
- Creating a Browser History Object
- Implementing Programmatic Navigation
- Manually Changing API Records
- URL-Based Selection
- Wildcard Navigation
- More on Route Params
- Selecting Records from State
- Component Isolation with React Router
- Fetching a Stream for Edit Stream
- Real Code Reuse!
- Refactoring Stream Creation
- Setting Initial Values
- Avoiding Changes to Properties
- Edit Form Submission
- PUT vs PATCH Requests

Using React Portals

- Why Use Portals?
- More on Using Portals
- Creating a Portal
- Hiding a Modal
- Making the Modal Reusable
- React Fragments
- OnDismiss From the Parent
- Reminder on Path Params
- Fetching the Deletion Stream
- Conditionally Showing Stream Details
- Deleting a Stream
- Closing the Modal

Implementing Streaming Video

- Viewing a Stream
- Switches with React-Router
- Showing a Stream
- RTMP NodeMediaServer is not a constructor error fix
- RTMP Server Setup
- OBS Installation

- OBS Scene Setup
- Video Player Setup
- Implementing FLV JS
- Creating a FLV Player
- Optional Player Building
- It Works!
- Cleaning Up with Component Will Unmount
- Exercise Overview - Adding Some Filtering
- Filtering the Stream List
- Exercise Solution

The Context System with React

- The Context System
- An App with Context
- App Generation
- Selecting a Language
- A Touch More Setup
- Getting Data Out of Context
- Creating Context Objects
- Consuming the Context Value
- The Context Provider
- Gotchas Around Providers
- Accessing Data with Consumers
- Pulling From Multiple Contexts
- Exercise Overview - Consuming Context Values From Providers
- Creating and Consuming Context
- Exercise Solution
- Refactor to use React Final Form instead of Redux Form

Replacing Redux with Context

- Replacing Redux with Context?
- Creating a Store Component
- Implementing a Language Store
- Rendering the Language Store
- Connecting the Selector to the Store
- Connecting the Field and Button to the Store
- Context vs Redux Recap