1. Getting Started
   1. Why React
   2. Reacts's basic concept
   3. Your first react component
   4. Your first React Hook
   5. Your first one-way data flow
   6. Components Reusability
   7. Tree reconciliation in Action
2. Modern javascript
   1. ECMAScript and TC39
   2. Variables and Block Scopes
   3. Arrow functions
   4. Object Literals
   5. Destructuring and Rest/Spread
   6. Template strings
   7. Classes
   8. Promises and Async/Await
3. Setting up development environment
   1. Create React App
   2. Installing Environment Dependencies
   3. Configuring your own environment
   4. Reactful
   5. Structuring Component Files
4. Designing better Components with React
   1. Recognizing a React App is just a Collection of Components
   2. Deciding how to divide your app into components
   3. Loading static assets into our visual studio code react project
   4. Create a nested tree of components from a single speaker component
   5. Introducing using context as a refactoring strategy
   6. Extract session and sessions components from our single component
   7. Learn multiple ways to pass data into components using props
   8. Create a nested tree of components from a single speaker component
5. Managing React State in a Component Hierarchy
   1. Adding a separate Toolbar to Control What Our Speaker List Renders
   2. Adding a header toolbar to our speakers list app
   3. Adding local state management to out toolbar
   4. Lifting state for sharing between components
   5. Keeping state close to where it is used
   6. Moving static speaker data to local component state
   7. Updating speaker state from speakers card component
   8. Reacting to events in nested components
6. Asynchronously updating React state using custom React hooks
   1. Handling Asynchronous Loading and State Updates
   2. Adding Error Handling including Rendering Placeholder Graphics
   3. Combining useState and useEffect to a Custom React Hooks
   4. Normalize State Usage with Loading Status Instead of Booleans
   5. Generalize the useRequestSpeakers Custom React Hook
   6. Learning Functional Programming Techniques to Implement Callbacks in Components
   7. Implementing a Completion Callback in Our Speaker Favorite Component
   8. Implementing UI Progress Animations While Updating Records
   9. Improve App Usability by Implementing and Optimistic UI for Updates
   10. Changing Speaker Favorite Updates to Respond Optimistically and Takeaways
7. Using React Context and the useContext React Hook for Component Data Sharing
   1. Why Use React Context?
   2. Creating a Theme Context and Sharing It with Our Header Component
   3. Introducing a Layout Component That Both Instantiates and Uses Our Theme Component
   4. Extracting Our Theme Context to Its Own Dedicated Component and File
   5. Separating State Management from Our Theme Context with a Custom React Hook
   6. Create a Speaker Filter Context Synchronize the Toolbar and Speaker List
   7. Adding More Controls to Our Speakers Toolbar
   8. React Context Takeaways
8. Use React Context to Share a Speaker Data and CRUD Functions
   1. Building and Using a Dedicated Speaker Context
   2. Adding Insert and Delete to Our Request Context’s Update Capability
   3. Implement a React UI to Create and Delete Speakers
   4. Implementing a REST Server Using API Routes from NextJS
   5. Implement REST GET Using API Routes
   6. Implement REST PUT POST and DELETE with API Routes
   7. Update useRequestDelay to Persist Updates with REST
   8. Integrate Changes including Handling Broken Images and Takeaways
9. Master Performance Monitoring, Error Reporting, and Debugging of Components
   1. Introduction to React Developer Tools
   2. Motivation for React Debug Tooling
   3. Working with the React Dev Tools Component Tab
   4. Working with the React Dev Tools Profiler Tab
   5. Fixing Our over Rendering Performance Problem
   6. Adding Error Boundaries for Catching Errors in Production
   7. Improving Our React Boundary Error UI and Takeaways
10. Deciding How and When to Declare State
    1. Target Audience and Prerequisites
    2. Agenda
    3. A Brief History of React State Management
    4. Eight Ways to Handle React State
    5. JavaScript Data Structures
11. Managing Local and Remote State
    1. Intro
    2. Set up Demo
    3. Declaring State with useState
    4. The Rules of Hooks
    5. Implementing Event Handlers
    6. React Developer Tools
    7. Implementing Derived State and Exploring When React Renders
    8. Fetching and Storing Data Via useEffect
    9. Four Ways to Handle API Calls
    10. Handling Errors via Error Boundary
    11. Handling Loading State
    12. Implementing Async/await in useEffect
    13. Creating Custom Hooks for Handling Remote State
    14. Consuming a Custom Hook
12. Implementing Routing
    1. Intro
    2. Configuring React Router's Entry Point
    3. Creating an App Layout
    4. Declaring Routes
    5. Declaring URL Placeholders
    6. Reading URL Placeholders
    7. Handling 404s
    8. Reading URL Placeholders on the Detail Page
    9. Implementing Client-side Nav via Link and NavLink
    10. Fetching Based on a URL Parameter
    11. Handling 404s on the Detail Page
    12. Redirecting via useNavigation
13. Managing Shared, Derived, and Immutable State
    1. Intro
    2. Creating a Shoe Size Select
    3. Implementing Derived State
    4. Deciding Where to Store State
    5. Lifting State
    6. Why Immutability?
    7. Handling Immutability
    8. Implementing Immutable-Friendly Add
    9. Displaying Cart Items
    10. Implementing Immutable-Friendly Update
    11. Implementing Immutable-Friendly Delete
    12. Deriving State for Cart Header
    13. Lazy Initializing State and Persisting State to localStorage
14. Implementing Form Validation
    1. Intro
    2. Creating Checkout
    3. Centralizing Change Handlers And Persisting Events
    4. Form Validation Decisions
    5. Intro to State Enums
    6. Implementing a State Enum On Checkout
    7. Saving the Form Submission and Emptying Cart
    8. State Enums vs. Finite State Machines
    9. Implementing "On-the-fly" Form Validation
    10. Tracking Touch for Instant Validation
15. Managing State via Refs
    1. Intro
    2. When to Consider a Ref
    3. Implementing Uncontrolled Inputs
    4. Controlled vs. Uncontrolled Inputs
    5. Avoiding Setting State on Unmounted Components
    6. Storing a Previous Value
16. Managing Complex State with useReducer
    1. Intro
    2. Why useReducer?
    3. Extracting State Management to UseReducer
    4. Handling Unexpected Action Types
    5. Handling Add In Reducers
    6. Handling Updates In Reducers
    7. Consuming Reducers
    8. Dispatching Actions
    9. useState vs. useReducer
17. Sharing State and Functions via Context
    1. Intro
    2. Why Context?
    3. Declaring a Context
    4. Declaring a Context Provider
    5. Consuming a Context
    6. Creating a Provider Component
    7. Wrapping the App in a Context Provider
    8. Creating a Reusable useContext Hook
    9. Consuming Context via a Custom Hook
    10. Requiring a Context Provider
18. Managing State in Class Components
    1. Intro
    2. Class Components vs Function Components
    3. Sharing Logic and Consuming Hooks - Option 1: Pass via Props
    4. Declaring State
    5. Deriving State
    6. Setting State
    7. Destructuring and Deriving State in Render
    8. Binding to This via Arrow Functions in Class Fields
    9. Sharing Logic and Consuming Hooks - Option 2: Custom Wrapper
    10. Sharing Logic and Consuming Hooks - Option 3: Render Prop
    11. Consuming a Render Prop
    12. Sharing Logic and Consuming Hooks - Option 4: Function as Child
    13. Consuming a Single Context via contextType
    14. Consuming Multiple Contexts
19. Managing State via Third Party Libraries
    1. Intro
    2. How to Run the Demos
    3. Local State
    4. Global State
    5. Server State
    6. Immutable State
    7. Form State
    8. Summary: How to Handle State
    9. Recommended Exercises
20. What React Has Done for Styling
    1. Motivation to Style
    2. Influence of React
21. A Catalogue of Styling Techniques
    1. Inline Styles
    2. CSS-in-JS
    3. CSS Stylesheets
    4. CSS Modules
22. Comparing Styling with Examples
    1. Inline Styles
    2. CSS-in-JS
    3. CSS Stylesheets
    4. CSS Modules
23. Potential Project Configurations for CSS in React
    1. Intro
    2. Webpack
    3. Rollup
    4. Parcel
24. Understanding Server Rendering React Components
    1. Introduction and Roadmap
    2. A Cutting Edge Startup Application Scenario
    3. Costs and Benefits of Server-rendered React Applications
    4. How Server Rendering Works
    5. Understanding Relevant Tools
    6. A Look at the Completed Application
25. Scaffolding an Environment for Server Rendering
    1. Introduction / Learning Objectives
    2. Scaffolding Decisions: Using Create React App
    3. Creating a Project and Installing Dependencies
    4. Setting up Babel
    5. Creating the Main React Component
26. Server Rendering Basic React Components
    1. Client and Server Rendering Comparison
    2. Server Rendering Workflow
    3. Rendering a React Component on the Server
    4. Understanding the Challenges of Server Rendering
    5. Adding Functionality to Our Application
27. Rehydrating Interactive React Components
    1. Limitations of Server Rendered Components
    2. Understanding Rehydration
    3. Adding Interactivity to Server Rendered Components through Rehydration
    4. Sharing Code between Client and Server
28. Creating Forms Using Vanilla React
    1. Overview
    2. Managing Form State in React
    3. How Controlled Forms Work in React
    4. Creating Your First Controlled Form in React
    5. Adding Data Validation in a Controlled Form
29. Creating Forms Using Formik
    1. Advantages of Using Formik for Creating Forms
    2. Key Components Provided by Formik
    3. Creating a Form Using Formik
    4. Handling Form Submission in Formik
    5. Rewriting a React Form Using Formik
30. Implementing Data Validation
    1. Validation Flavors Provided by Formik
    2. Controlling Validation Trigger Points in Formik
    3. Showing Field and Form Level Validation Errors in a Form
    4. Advantages of Using Yup with Formik
    5. Integrating Yup into Formik Forms
31. Creating Reusable Custom Form Elements
    1. Need for Creating Custom Form Elements
    2. Support for Creating Custom Form Elements in Formik
    3. Adding Custom Fields to a Formik Form
32. Implementing Uncontrolled Forms Using Vanilla React
    1. Implementing Uncontrolled Forms Using Vanilla React
    2. Adding File Upload Functionality to a Form
33. Using React Hook Form to Create Uncontrolled Forms
    1. Creating a Form Using React Hook Form
    2. Data Validation in React Hook Form
34. Fixing Wasted Rendering Issues
    1. Introduction to React Rendering
    2. The Virtual DOM
    3. Preventing Wasted Renders in a Simple Component
    4. Preventing Wasted Renders in Functional Components
    5. Preventing Wasted Renders When Dealing with Complex Props
    6. Using Immutable Data in Order to Allow for Comparisons
    7. Preventing Wasted Renders in Repeated Components
35. Fixing Large Bundles and Expensive Operation Issues
    1. Caching Expensive Operation Results
    2. Reducing Bundle Sizes
    3. Lazy Loading Components
36. Intro to Redux
    1. Intro
    2. Do I Need Redux?
    3. When Is Redux Helpful?
    4. Three Core Redux Principles
    5. Flux vs. Redux
    6. Redux Flow Overview
37. Actions, Stores, and Reducers
    1. Intro
    2. Actions
    3. Store
    4. What Is Immutability?
    5. Why Immutability?
    6. Handling Immutability
    7. Reducers
38. Connecting React to Redux
    1. Intro
    2. Container vs. Presentational Components
    3. React-Redux Introduction
    4. mapStateToProps
    5. mapDispatchToProps
    6. A Chat with Redux
39. Redux Flow
    1. Intro
    2. Create Simple Add Course Form
    3. Binding in Classes
    4. Handle Submit
    5. Create Course Action
    6. Create Course Reducer and Root Reducer
    7. Create Store
    8. Instantiate Store and Provider
    9. Connect Container Component
    10. Step through Redux Flow and Try Redux DevTools
    11. mapDispatchToProps: Manual Mapping
    12. mapDispatchToProps: bindActionCreators
    13. mapDispatchToProps: Object Form
    14. Action Type Constants
40. Async in Redux
    1. Intro
    2. Why a Mock API?
    3. Mock API Setup
    4. Middleware and Async Library Options
    5. Thunk Introduction
    6. Remove Inline Manage Course Form
    7. Add First Thunk
    8. Handle Loading Courses in Reducer
    9. Dispatch Actions on Load
    10. Create CourseList
    11. Practice Redux Flow - Load and Display Author Data
    12. Centralize Initial Redux State
41. Async Writes in Redux
    1. Intro
    2. Create Manage Course Page
    3. Implement Object Form of mapDispatchToProps
    4. Configure Routing for ManageCoursePage
    5. Convert Class Component to Function Component with Hooks
    6. Create Course Form
    7. Create Reusable TextInput and SelectInput Components
    8. Call CourseForm on ManageCoursePage
    9. Implement Centralized Change Handler
    10. Add Save Course Thunk and Action Creators
    11. Handle Creates and Updates in Reducer
    12. Dispatch Create and Update
    13. Redirect via React Router's Redirect Component
    14. Redirect via React Router's History
    15. Populate Form via mapStateToProps
42. Async Status and Error Handling
    1. Intro
    2. Create Spinner Component
    3. Create API Status Actions
    4. Call Begin API in Thunks
    5. Add Spinner to Course Pages
    6. Add Form Submission Loading Indicator
    7. Handle Server-side Validation and API Errors
    8. Implement Client-side Validation
    9. Optimistic Deletes
    10. Async/Await
43. Testing React
    1. Intro
    2. Testing Frameworks and Libraries
    3. Configure Jest
    4. Test React with Jest Snapshot Tests
    5. Test React with Enyzme
    6. Test React with React Testing Library
44. Testing Redux
    1. Intro
    2. Overview: Testing Connected React Components
    3. Testing Connected React Components
    4. Testing Action Creators
    5. Testing Thunks
    6. Testing Reducers
    7. Testing the Store
45. Production Builds
    1. Intro
    2. Production Build Plan Overview
    3. Set up Production Redux Store
    4. Set up Webpack
    5. Set up npm Scripts
    6. Run Production Build and Review Bundle Content
    7. Run Production App Locally
46. Start Using React Hooks with useState, useEffect, and useRef
    1. Learn How React Hooks Makes React Better with a Simple Example
    2. Setting up Our React Tool Chain
    3. Our First React Hook setState, and Tracking Input Field Value
    4. Using useRef to Enable Mouseover to Colorize an Image
    5. Learning How to Use the useEffect Hook
    6. Using useEffect, useRef and useState for Colorizing on Scroll
    7. Improving and Fixing Our Scrolling Colorizing React App
    8. Adding a Side Effect to Our Scroller Component
    9. Optimizing Performance with the useEffect Dependency Array
47. Using More Hooks: useContext, useReducer, useCallback, and useMemo
    1. Why Should You Use React Hooks and Does It Matter?
    2. React Hooks Basic Usage and Rules
    3. Updating Our Examples to a Real-world Conference React App
    4. Building out the Conference Speakers Page with Our Primitive React Hooks
    5. Preparing React Context for Use with useContext Hook
    6. Using useContext to Access Global Configuration from React Context API
    7. useReducer Is What useState Is Built On
    8. Using useReducer with Multiple Dispatched Actions
    9. Using useCallback to Improve React App Performance
    10. Using useMemo to Optimize Filtering and Sorting Speakers
48. Migrating Your Existing Apps to React Hooks
    1. What’s Different and What’s the Same between Hooks and No Hooks
    2. Tracking State Changes in React Hooks Functional Components vs. React Class Components
    3. What useState Represents in a Class Component
    4. What useRef Represents in a Class Component
    5. What useEffect Represents in a Class Component
49. Learn How to Combine Existing React Hooks into New Combined Hooks
    1. Benefits of Combing React Hooks into Custom Hooks
    2. Timing out and Email Validation on the Signup Form
    3. Combining Multiple React Hooks into One Custom Hook
    4. Updating Our Speakers Component to Handle Multiple State Values Using Destructuring
    5. Creating a Custom React Hook to Manage Speaker Data and Loading
    6. Extending Our Custom React Hook with Update Functionality
    7. The Basics of Using a REST Service as Our Speaker Data Source
    8. Converting Our Custom React Hook to Use REST Instead of a Static Array
    9. Replace Array with REST Service Calls in Custom React Hook
    10. Eliminate Client Loading Message with Server-side Rendering of HTML
    11. Coding Server-side Rendering with No useEffect Lifecycle Event
    12. Integrating Server-side Generated Code with useContext on the Client
    13. Static Site Rendering and Near Infinite Site Scaling
50. Using useContext and useReducer to Make a Redux-like Global State
    1. Why Manage React State Globally?
    2. Create a Global Context and Include in Our App
    3. Replace Our Custom Hook Usage with a Reference to Global Context
    4. Create a New Custom Hook for Counting Clicks Per Speaker
    5. Consolidate and Totaling Favorite Click Counts to the Page Header
    6. Creating Nested Global Contexts to Reduce Re-renders of Components
    7. Handling Errors in Your React App
    8. Using Context to Solve a Corner Case
    9. Implementing Our Colorizing on Checkbox Change Bug Fix
51. Factors to Consider When Choosing a Framework
    1. Criteria 1: Ecosystem
    2. Criteria 2: Performance
    3. Criteria 3: Business Models and Pricing
    4. Criteria 4: Scalability
52. Framework Basics and “create-react-app”
    1. How Does Create React App Work?
    2. Reviewing Create React App
53. Third Party Framework Options
    1. Reviewing “Next.js”
    2. Reviewing “Gatsby”
    3. Reviewing “Blitz”
    4. Reviewing “Redwood”
    5. Reviewing “Remix”
54. Comparing Frameworks
    1. Comparing Ecosystems
    2. Comparing Performance
    3. Comparing Business Models and Pricing
    4. Comparing Scalability
55. Making HTTP Requests
    1. Introduction
    2. API Workflow
    3. Prerequisites
    4. Globoticket Site
    5. Touring the Site's Code
    6. Touring the API Code
    7. React Lifecycle
    8. JavaScript HTTP Request Options
    9. Fetch API
    10. Using the Events API
    11. Using the Shopping Cart API
56. More HTTP Request Options
    1. Introduction
    2. Going Async
    3. Using Redux Thunk - Part 1
    4. Using Redux Thunk - Part 2
    5. The Axios HTTP Client
    6. Using Axios
57. Advanced Data Retrieval
    1. Introduction
    2. Pain Points
    3. React Query
    4. Using React Query
    5. Adding Pagination
    6. Mutations
    7. Using SWR
    8. SWR Mutations
    9. Code Cleanup
58. Testing and Error Handling
    1. Introduction
    2. Faking APIs
    3. Mocking with Mirage JS
    4. Stubbing with json-server
    5. Centralized Error Handling
    6. Error Boundaries
    7. Using Suspense
59. Analyzing Performance with React Developer Tools
    1. Overview
    2. Renders and Commits
    3. The Globotickets Application
    4. Profiling with React Developer Tools
    5. Demo: Profiling with React Developer Tools
60. Improving Performance through Pagination
    1. What Is Pagination?
    2. Demo: Implementing Pagination
    3. How Pagination Improves Performance
    4. Demo: How Pagination Improves Performance
61. Optimizing Renders with Infinite Scroll and Windowing
    1. Infinite Scroll: An Improved Version of Pagination
    2. Demo: Implementing Infinite Scroll
    3. Introduction to Windowing
    4. Demo: Adding Windowing to Our Application
62. Eliminating Needless Renders
    1. Introduction
    2. React Component Keys
    3. React Rendering and Lifecycle Methods
    4. Demo: Improving Performance Using Keys and Lifecycle Methods
63. Preventing Cross-site Scripting Attacks
    1. The Globomantics Bug Tracker Project
    2. Stealing Data with Cross-site Scripting Attacks
    3. Cross-site Scripting Sinks
    4. Automatic Escaping in React
    5. Safely Rendering URLs
64. Safe Dynamic Content Rendering
    1. Dynamic Content Rendering in React
    2. Sanitization with DOMPurify
    3. Risks of Direct DOM Manipulation Using Refs
    4. Dynamic Parsing of React Components
65. Defending against Code Injection through JSON Data
    1. Server-side Rendering in React Applications
    2. Stored and Reflected Cross-site Scripting
    3. Sanitization of JSON Data