

# **React and React Native**

## ***Third Edition***

A complete hands-on guide to modern web and mobile development with React.js

**Adam Boduch**  
**Roy Derks**

**Packt**

**BIRMINGHAM - MUMBAI**

# Table of Contents

<b>Preface</b>	1
<hr/>	
<b>Section 1: React</b>	
<hr/>	
<b>Chapter 1: Why React?</b>	9
<b>What is React?</b>	9
React is just the view layer	10
Simplicity is good	11
Declarative UI structures	12
Time and data	12
Performance matters	13
The right level of abstraction	15
<b>React Features</b>	16
Revamped core architecture	17
Lifecycle methods	17
The Context API	17
Rendering fragments	18
Portals	18
Rendering lists and strings	18
Handling errors	19
Server-side rendering	19
<b>What's new in React?</b>	20
Memoizing functional components	20
Code splitting and loading	20
Hooks	21
<b>Summary</b>	21
<b>Further reading</b>	22
<b>Chapter 2: Rendering with JSX</b>	23
<b>Technical requirements</b>	23
<b>Your first JSX content</b>	23
Hello JSX	24
Declarative UI structures	24
<b>Rendering HTML</b>	25
Built-in HTML tags	25
HTML tag conventions	26
<b>Describing UI structures</b>	27
<b>Creating your own JSX elements</b>	28
Encapsulating HTML	28
Nested elements	30

Namespaced components	31
<b>Using JavaScript expressions</b>	33
Dynamic property values and text	34
Mapping collections to elements	35
<b>Fragments of JSX</b>	36
Using wrapper elements	37
Using fragments	38
<b>Summary</b>	39
<b>Further reading</b>	40
<b>Chapter 3: Component Properties, State, and Context</b>	41
<b>Technical requirements</b>	42
<b>What is component state?</b>	42
<b>What are component properties?</b>	43
<b>Setting a component state</b>	44
Setting an initial component state	44
Creating a component state	45
Merging the component state	47
<b>Passing property values</b>	49
Default property values	50
Setting property values	51
<b>Stateless components</b>	53
Pure functional components	53
Defaults in functional components	55
<b>Container components</b>	55
<b>Providing and consuming context</b>	57
<b>Summary</b>	61
<b>Further reading</b>	61
<b>Chapter 4: Getting Started with Hooks</b>	62
<b>Technical requirements</b>	62
<b>Maintaining state using Hooks</b>	62
Initial state values	63
Updating state values	64
<b>Performing initialization and cleanup actions</b>	66
Fetching component data	66
Canceling requests and resetting state	68
Optimizing side-effect actions	72
<b>Sharing data using context Hooks</b>	73
Sharing fetched data	74
Updating stateful context data	78
<b>Using reducer Hooks to scale state management</b>	81
Using reducer actions	82
Handling state dependencies	84
<b>Summary</b>	90

<b>Chapter 5: Event Handling - The React Way</b>	91
<b>Technical requirements</b>	91
<b>Declaring event handlers</b>	92
Declaring handler functions	92
Multiple event handlers	93
Importing generic handlers	93
<b>Using event handler context and parameters</b>	95
Getting component data	95
Higher-order event handlers	97
<b>Declaring inline event handlers</b>	99
<b>Binding handlers to elements</b>	99
<b>Using synthetic event objects</b>	100
<b>Understanding event pooling</b>	101
<b>Summary</b>	103
<b>Further reading</b>	103
<b>Chapter 6: Crafting Reusable Components</b>	104
<b>Technical requirements</b>	105
<b>Reusable HTML elements</b>	105
<b>The difficulty with monolithic components</b>	105
The JSX markup	106
Initial state	108
Event handler implementation	109
<b>Refactoring component structures</b>	111
Starting with the JSX	111
Implementing an article list component	113
Implementing an article item component	115
Implementing an add article component	116
Making components functional	118
<b>Render props</b>	120
<b>Refactoring class components using Hooks</b>	122
<b>Rendering component trees</b>	126
<b>Feature components and utility components</b>	127
<b>Summary</b>	128
<b>Further reading</b>	128
<b>Chapter 7: The React Component Life Cycle</b>	129
<b>Technical requirements</b>	129
<b>Why components need a life cycle</b>	130
<b>Initializing properties and state</b>	131
Fetching component data	131
Initializing state with properties	134
Updating state with properties	136
<b>Optimizing rendering efficiency</b>	138

To render or not to render	139
Using metadata to optimize rendering	142
<b>Rendering imperative components</b>	143
Rendering jQuery UI widgets	144
<b>Cleaning up after components</b>	146
Cleaning up asynchronous calls	147
<b>Containing errors with error boundaries</b>	149
<b>Summary</b>	153
<b>Further reading</b>	154
<b>Chapter 8: Validating Component Properties</b>	155
<b>Technical requirements</b>	155
<b>Knowing what to expect</b>	156
<b>Promoting portable components</b>	156
<b>Simple property validators</b>	157
Basic type validation	157
Requiring values	160
Any property value	163
<b>Type and value validators</b>	164
Things that can be rendered	165
Requiring specific types	166
Requiring specific values	168
<b>Writing custom property validators</b>	170
<b>Summary</b>	172
<b>Further reading</b>	172
<b>Chapter 9: Handling Navigation with Routes</b>	173
<b>Technical requirements</b>	173
<b>Declaring routes</b>	174
Hello route	174
Decoupling route declarations	175
Parent and child routes	177
<b>Handling route parameters</b>	179
Resource IDs in routes	179
Optional parameters	184
<b>Using link components</b>	186
Basic linking	187
URL and query parameters	188
<b>Summary</b>	190
<b>Further reading</b>	190
<b>Chapter 10: Code Splitting Using Lazy Components and Suspense</b>	191
<b>Technical requirements</b>	192
<b>Using the lazy API</b>	192
Dynamic imports and bundles	192

Making components lazy	193
<b>Using the Suspense component</b>	194
Top-level Suspense components	194
Simulating latency	196
Working with spinner fallbacks	197
<b>When to avoid lazy components</b>	198
<b>Lazy pages and routes</b>	200
<b>Summary</b>	202
<b>Chapter 11: Server-Side React Components</b>	203
<b>Technical requirements</b>	203
<b>What is isomorphic JavaScript?</b>	203
The server is a render target	204
Initial load performance	204
Sharing code between the server and the browser	205
<b>Rendering to strings</b>	206
<b>Backend routing</b>	208
<b>Frontend reconciliation</b>	211
<b>Fetching data</b>	214
<b>Summary</b>	217
<b>Further reading</b>	218
<b>Chapter 12: User Interface Framework Components</b>	219
<b>Technical requirements</b>	219
<b>Layout and organization</b>	220
Using containers	220
Building responsive grid layouts	222
<b>Using navigation components</b>	225
Navigating with drawers	225
Navigating with tabs	229
<b>Collecting user input</b>	232
Checkboxes and radio buttons	232
Text inputs and select inputs	233
Working with buttons	235
<b>Working with styles and themes</b>	237
Making styles	238
Customizing themes	239
<b>Summary</b>	241
<b>Section 2: React Native</b>	
<b>Chapter 13: Why React Native?</b>	243
<b>Technical requirements</b>	243
<b>What is React Native?</b>	243
<b>React and JSX are familiar</b>	245

<b>The mobile browser experience</b>	245
<b>Android and iOS – different yet the same</b>	246
<b>The case for mobile web apps</b>	247
<b>Summary</b>	247
<b>Further reading</b>	248
<b>Chapter 14: Kick-Starting React Native Projects</b>	249
<b>Technical requirements</b>	249
<b>Installing and using the Expo command-line tool</b>	250
<b>Viewing your app on your phone</b>	251
<b>Viewing your app on Expo Snack</b>	257
<b>Summary</b>	261
<b>Chapter 15: Building Responsive Layouts with Flexbox</b>	262
<b>Technical requirements</b>	262
<b>Flexbox is the new layout standard</b>	263
<b>Introducing React Native styles</b>	264
<b>Building Flexbox layouts</b>	266
Simple three-column layout	267
Improved three-column layout	270
Flexible rows	273
Flexible grids	275
Flexible rows and columns	278
<b>Summary</b>	281
<b>Further reading</b>	282
<b>Chapter 16: Navigating Between Screens</b>	283
<b>Technical requirements</b>	283
<b>Navigation basics</b>	284
<b>Route parameters</b>	287
<b>The navigation header</b>	290
<b>Tab and drawer navigation</b>	294
<b>Handling state</b>	298
<b>Summary</b>	305
<b>Further reading</b>	305
<b>Chapter 17: Rendering Item Lists</b>	306
<b>Technical requirements</b>	307
<b>Rendering data collections</b>	307
<b>Sorting and filtering lists</b>	309
<b>Fetching list data</b>	316
<b>Lazy list loading</b>	318
<b>Summary</b>	320
<b>Further reading</b>	320

<b>Chapter 18: Showing Progress</b>	321
<b>Technical requirements</b>	321
<b>Progress and usability</b>	321
<b>Indicating progress</b>	322
<b>Measuring progress</b>	325
<b>Navigation indicators</b>	330
<b>Step progress</b>	332
<b>Summary</b>	336
<b>Further reading</b>	337
<b>Chapter 19: Geolocation and Maps</b>	338
<b>Technical requirements</b>	338
<b>Where am I?</b>	338
<b>What's around me?</b>	341
<b>Annotating points of interest</b>	342
Plotting points	342
Plotting overlays	344
<b>Summary</b>	347
<b>Further reading</b>	347
<b>Chapter 20: Collecting User Input</b>	348
<b>Technical requirements</b>	348
<b>Collecting text input</b>	348
<b>Selecting from a list of options</b>	351
<b>Toggling between on and off</b>	355
<b>Collecting date/time input</b>	358
<b>Summary</b>	363
<b>Further reading</b>	363
<b>Chapter 21: Displaying Modal Screens</b>	364
<b>Technical requirements</b>	364
<b>Important information</b>	365
<b>Getting user confirmation</b>	365
Displaying a success confirmation	366
Error confirmation	371
<b>Passive notifications</b>	375
<b>Activity modals</b>	379
<b>Summary</b>	382
<b>Further reading</b>	382
<b>Chapter 22: Responding to User Gestures</b>	383
<b>Technical requirements</b>	383
<b>Scrolling with your fingers</b>	384
<b>Giving touch feedback</b>	386
<b>Swipeable and cancellable</b>	389

<b>Summary</b>	394
<b>Further reading</b>	394
<b>Chapter 23: Controlling Image Display</b>	395
<b>Technical requirements</b>	395
<b>Loading images</b>	396
<b>Resizing images</b>	398
<b>Lazy image loading</b>	402
<b>Rendering icons</b>	405
<b>Summary</b>	408
<b>Further reading</b>	408
<b>Chapter 24: Going Offline</b>	409
<b>Technical requirements</b>	409
<b>Detecting the state of the network</b>	409
<b>Storing application data</b>	413
<b>Synchronizing application data</b>	416
<b>Summary</b>	421
<b>Further reading</b>	421
<b>Section 3: React Architecture</b>	
<hr/>	
<b>Chapter 25: Native UI Components Using NativeBase</b>	423
<b>Technical requirements</b>	424
<b>Application containers</b>	424
<b>Headers, footers, and navigation</b>	427
<b>Using layout components</b>	431
<b>Collecting input using form components</b>	434
<b>Displaying data using lists</b>	437
<b>Showing user notifications</b>	440
<b>Summary</b>	442
<b>Chapter 26: Handling Application State</b>	443
<b>Technical requirements</b>	443
<b>Information architecture and Flux</b>	444
Unidirectionality	444
Synchronous update rounds	445
Predictable state transformations	445
<b>Unified information architecture</b>	446
<b>Implementing Redux</b>	446
Initial application state	447
Creating the store	448
Store provider and routes	448
The App component	449
The Home component	452

State in mobile apps	456
<b>Scaling the architecture</b>	457
<b>Summary</b>	458
<b>Further reading</b>	458
<b>Chapter 27: Why Apollo?</b>	459
Yet another approach?	460
Verbose vernacular	460
Declarative data fetching	461
Mutating application state	464
<b>Summary</b>	466
<b>Further reading</b>	466
<b>Chapter 28: Building an Apollo React App</b>	467
Technical requirements	467
Todo and Apollo Client	467
The GraphQL schema	469
Bootstrapping Apollo Client	470
Adding todo items	474
Rendering todo items	478
Completing todo items	479
<b>Summary</b>	481
<b>Other Books You May Enjoy</b>	482
<b>Index</b>	485

---