

Learn React with TypeScript

A beginner's guide to reactive web development with React 18
and TypeScript

Carl Rippon



BIRMINGHAM—MUMBAI

Table of Contents

Part 1: Introduction

1

Introducing React	3
Technical requirements	4
Understanding the benefits of React	4
Understanding JSX	5
Creating a component	8
Creating a CodeSandbox project	8
Understanding the React entry point	9
Understanding the React component tree	10
Creating a basic alert component	11
Understanding imports and exports	14
Understanding the importance of modules	14
Using export statements	14
Using import statements	16
Adding Alert to the App component	16
Using props	18
Understanding props	18
Adding props to the alert component	19
Using state	22
Understanding state	22
Implementing a visible state in the alert component	23
Adding a close button to Alert	24
Using events	27
Understanding events	27
Implementing a close button click handler in the alert	28
Implementing an alert close event	29
Summary	32
Questions	32
Answers	34

2**Introducing TypeScript 37**

Technical requirements	37	Using the void type	52
Understanding the benefits of TypeScript	38	Using the never type	53
		Using arrays	55
Understanding TypeScript	38	Creating TypeScript types	56
Catching type errors early	39	Using object types	56
Improving developer experience and productivity with IntelliSense	41	Creating type aliases	58
		Creating interfaces	60
Understanding JavaScript types	42	Creating classes	61
Using basic TypeScript types	44	Creating enumerations	64
Using type annotations	44	Creating union types	66
Using type inference	46	Using the TypeScript compiler	68
Using the Date type	47	Summary	72
Using the any type	49	Questions	73
Using the unknown type	49	Answers	73

3**Setting Up React and TypeScript 75**

Technical requirements	75	Adding linting to Visual Studio Code	89
Creating a project with webpack	76	Adding code formatting	91
Introducing webpack	76	Starting the app in development mode	94
Creating the folder structure	76	Producing a production build	96
Creating package.json	76	Creating a React and TypeScript component	97
Adding a web page	77	Adding a props type	97
Adding TypeScript	77	Adding a state type	99
Adding React	79	Using React DevTools	100
Adding Babel	81	Summary	104
Adding webpack	83	Questions	104
Creating a project with Create React App	88	Answers	105
Using Create React App	88		

4

Using React Hooks	107
Technical requirements	107
Using the effect Hook	108
Understanding the effect Hook parameters	108
The rules of Hooks	109
Effect cleanup	111
Creating the project	112
Fetching data using the effect Hook	114
Using state Hooks	117
Using useState	118
Understanding useReducer	121
Using useReducer	122
Using the ref Hook	125
Understanding the ref Hook	125
Using the ref Hook	126
Using the memo Hook	129
Understanding the memo Hook	129
Using the memo Hook	130
Using the callback Hook	132
Understanding the callback Hook	132
Understanding when a component is re-rendered	133
Using the callback Hook	134
Summary	139
Questions	140
Answers	143

Part 2: App Fundamentals**5**

Approaches to Styling React Frontends	147
Technical requirements	147
Using plain CSS	148
Creating the project	148
Understanding how to reference CSS	148
Using plain CSS in the alert component	150
Experiencing CSS clashes	154
Using CSS modules	155
Understanding CSS modules	155
Using CSS modules in the alert component	156
Using CSS-in-JS	159
Understanding CSS-in-JS	160
Using Emotion in the alert component	161
Using Tailwind CSS	165
Understanding Tailwind CSS	165
Installing and configuring Tailwind CSS	167
Using Tailwind CSS	168
Using SVGs	172
Understanding how to use SVGs in React	172
Adding SVGs to the alert component	173
Summary	175
Questions	175
Answers	177

6**Routing with React Router** **179**

Technical requirements	180	Creating an error page	203
Introducing React Router	180	Understanding error pages	203
Creating the project	180	Adding an error page	204
Understanding React Router	180	Using index routes	207
Installing React Router	181	Understanding index routes	207
Declaring routes	181	Using an index route in the app	207
Creating the products list page	181	Using search parameters	209
Understanding React Router's router	184	Understanding search parameters	209
Declaring the products route	185	Adding search functionality to the app	210
Creating navigation	187	Navigating programmatically	214
Using the Link component	188	Using form navigation	216
Using the NavLink component	191	Implementing lazy loading	218
Using nested routes	192	Understanding React lazy loading	218
Understanding nested routes	192	Adding a lazy admin page to the app	219
Using nested routes in the app	195	Summary	222
Using route parameters	197	Questions	222
Understanding route parameters	197	Answers	225
Using route parameters in the app	198		

7**Working with Forms** **227**

Technical requirements	227	Using React Hook Form	246
Using controlled fields	228	Understanding React Hook Form	246
Creating the project	228	Using React Hook Form	249
Creating a contact form	229	Adding validation	252
Using uncontrolled fields	236	Summary	258
Using React Router Form	238	Questions	258
Using native validation	243	Answers	261

Part 3: Data

8

State Management	265
Technical requirements	265
Creating the project	266
Using prop drilling	270
Using React context	274
Understanding React context	274
Using React context	276
Using Redux	282
Understanding Redux	282
Installing Redux	285
Using Redux	285
Summary	291
Questions	292
Answers	293

9

Interacting with RESTful APIs	295
Technical requirements	295
Creating the project	296
Setting up the project	296
Understanding the component structure	296
Creating a REST API	297
Using React Router	316
Understanding React Router data loading	316
Using React Router for data loading	317
Deferred React Router data fetching	319
Using React Query	322
Installing React Query	322
Adding the React Query provider	322
Getting data using React Query	323
Updating data using React Query	327
Using React Router with React Query	330
Summary	333
Questions	333
Answers	334
Using the effect hook with fetch	299
Getting blog posts using fetch	299
Strongly typing response data	300
Creating a blog posts list component	303
Creating a blog posts page component	304
Posting data with fetch	308
Creating new blog posts using fetch	308
Creating a blog post form component	310

10

Interacting with GraphQL APIs 337

Technical requirements	337	Creating the repository page	354
Understanding the GraphQL syntax	338	Using Apollo Client	366
Returning flat data	338	Understanding Apollo Client	366
Returning hierarchical data	341	Installing Apollo Client	366
Specifying query parameters	343	Refactoring the App component	366
GraphQL mutations	344	Refactoring the Header component	367
Setting up the project	346	Refactoring the repository page	368
Creating the project	346	Summary	371
Creating a PAT for the GitHub GraphQL API	346	Questions	371
Creating environment variables	347	Answers	373
Using React Query with fetch	348		
Creating the header	348		

Part 4: Advanced React

11

Reusable Components 377

Technical requirements	377	Creating custom hooks	394
Creating the project	378	Understanding custom hooks	394
Using generic props	378	Extracting checked logic into a custom hook	395
Understanding generics	378	Allowing the internal state to be controlled	398
Creating a basic list component	380	Understanding how the internal state can be controlled	398
Using props spreading	384	Allowing checkedIds to be controlled	400
Using render props	387	Summary	405
Understanding the render props pattern	387	Questions	406
Adding a renderItem prop	388	Answers	409
Adding checked functionality	391		

12

Unit Testing with Jest and React Testing Library	413
<hr/>	
Technical requirements	414
Testing pure functions	414
Understanding a Jest test	414
Testing isChecked	416
Testing exceptions	417
Running tests	419
Testing components	422
Understanding React Testing Library	423
Implementing checklist component tests	425
Using test IDs	427
Simulating user interactions	429
Understanding fireEvent and user-event	430
Implementing checklist tests for checking items	430
Getting code coverage	434
Running code coverage	435
Understanding the code coverage report	435
Gaining full coverage on the checklist component	438
Ignoring files in the coverage report	439
Summary	440
Questions	441
Answers	442
Index	443
<hr/>	
Other Books You May Enjoy	452

Preface

React was built by Meta to provide more structure to its code base and allow it to scale much better. React worked so well for Facebook that they eventually made it open source. Today, React is the dominant technology for building the frontend of an application; it allows us to build small, isolated, and highly reusable components that can be composed together to create complex frontends.

TypeScript was built by Microsoft to help developers more easily develop large JavaScript-based programs. It is a superset of JavaScript that brings a rich type system to it. This type system helps developers to catch bugs early and allows tools to be created to robustly navigate and refactor code.

This book will teach you how to use both of these technologies to create large, sophisticated frontends that are easy to maintain.

Who this book is for

If you are a developer who wants to create large and complex frontends with React and TypeScript, this book is for you. The book doesn't assume you have any previous knowledge of React or TypeScript – however, basic knowledge of JavaScript, HTML, and CSS will help you get to grips with the concepts covered.

What this book covers

Chapter 1, Introducing React, covers the fundamentals of building React components. This includes defining component output using JSX, making a component configurable using props, and making a component interactive using states.

Chapter 2, Introducing TypeScript, is all about the fundamentals of TypeScript and its type system. This includes using inbuilt types, as well as creating new types.

Chapter 3, Setting Up React and TypeScript, explains how to create a project for React and TypeScript development. The chapter then moves on to how to create React components that use TypeScript to make props and states type-safe.

Chapter 4, Using React Hooks, details the common React hooks and their typical use cases. The chapter also covers how to use the hooks with TypeScript to make them type-safe.

Chapter 5, Approaches to Styling React Frontends, walks through how to style React components using several different approaches. The benefits of each approach are also explored.

Chapter 6, Routing with React Router, introduces a popular library that provides client-side routing for applications with multiple pages. It covers how to declare the paths for the pages and how to create links between them. It also covers how to implement page parameters for highly dynamic pages.

Chapter 7, Working with Forms, explores how forms can be implemented using several different approaches, including the use of a popular library. The benefits of each approach are also included.

Chapter 8, State Management, walks through how states can be shared between different components. Several approaches are explored, along with their benefits.

Chapter 9, Interacting with RESTful APIs, demonstrates how React components can interact with a REST API. The chapter steps through an approach using core React and then an alternative approach using a popular library.

Chapter 10, Interacting with GraphQL APIs, shows how React components can interact with a GraphQL API. The chapter details how this can be done using two different popular libraries.

Chapter 11, Reusable Components, brings in several patterns for making React components highly reusable but still type-safe.

Chapter 12, Unit Testing with Jest and React Testing Library, first delves into how functions can be tested with Jest. The chapter then moves on to how React components can be tested with the help of the React Testing Library.

To get the most out of this book

To get the most out of this book, you need to know the basics of JavaScript, including the following:

- An understanding of some of the primitive JavaScript types, such as `string`, `number`, `boolean`, `null`, and `undefined`
- An understanding of how to create variables and reference them, including arrays and objects
- An understanding of how to create functions and call them
- An understanding of how to create conditional statements with the `if` and `else` keywords

You also need to know the basics of HTML, including the following:

- An understanding of basic HTML elements such as `div`, `ul`, `a`, and `h1`
- An understanding of how to reference a CSS class to style an HTML element

An understanding of basic CSS is also helpful, including the following:

- How to size elements and include margins and padding
- How to position elements
- How to color elements

To follow along with this book, you will need the following technologies installed on your computer:

- A modern browser such as **Google Chrome**, which can be installed from <https://www.google.com/chrome/>
- **Node.js** and **npm**: You can install them from <https://nodejs.org/en/download/>
- **Visual Studio Code**: You can install it from: <https://code.visualstudio.com/>

Software/hardware covered in the book
React 18.0 or later
TypeScript 4.7 or later

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from GitHub at <https://github.com/PacktPublishing/Learn-React-with-TypeScript-2nd-Edition>. If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: <https://packt.link/5CvU5>.

Conventions used

There are a number of text conventions used throughout this book.

Code in text: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: “Here, `null` is passed because there are no properties.”

A block of code is set as follows:

```
<div className="title">
  <span>Oh no!</span>
</div>
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
React.createElement(  
  'span',  
  null,  
  title ? title : 'Something important'  
);
```

Any command-line input or output is written as follows:

```
$ mkdir css  
$ cd css
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: “Select **System info** from the **Administration** panel.”

Tips or important notes

Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at customercare@packtpub.com and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit www.packtpub.com/support/errata and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit authors.packtpub.com.