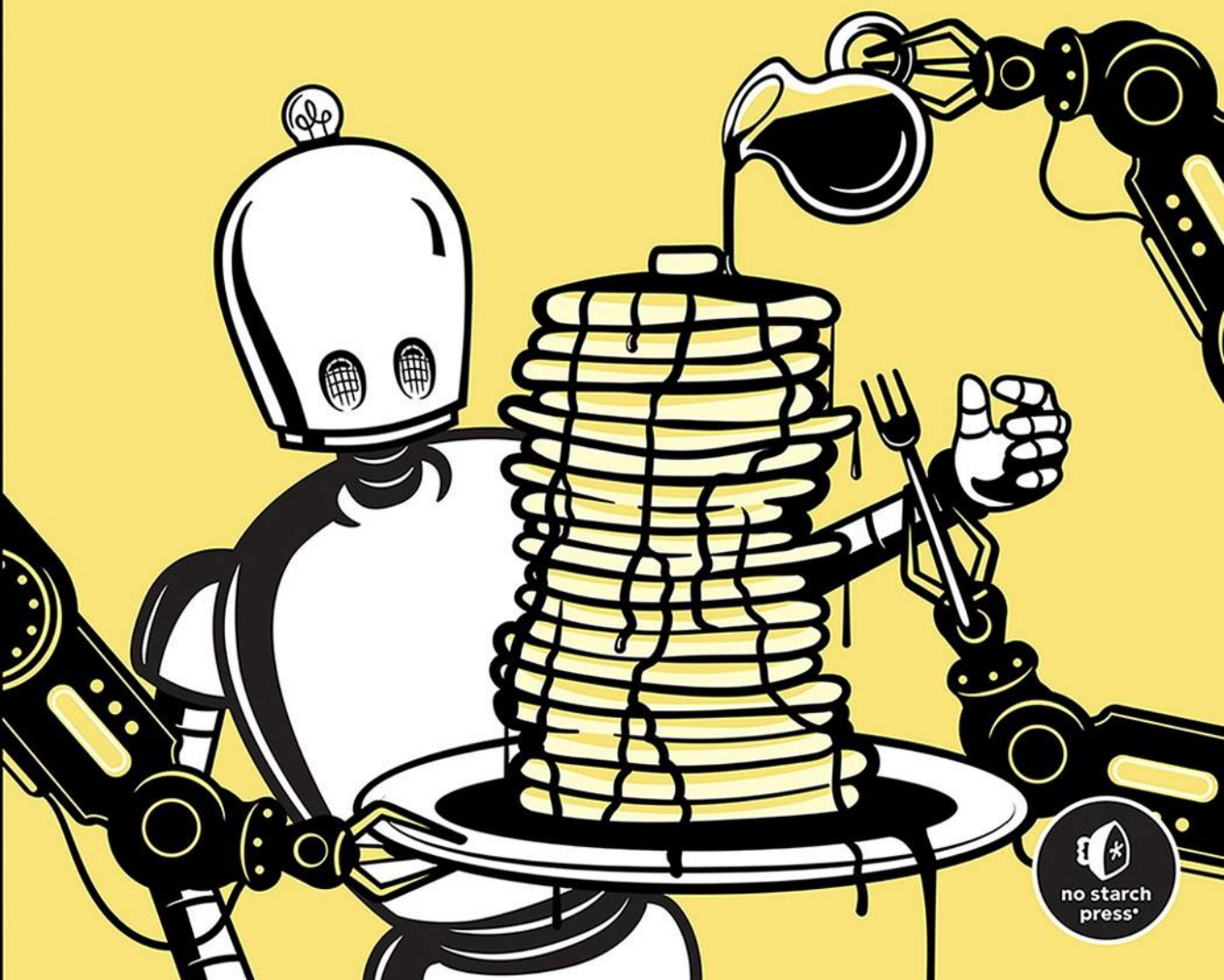


THE COMPLETE DEVELOPER

MASTER THE FULL STACK
WITH TYPESCRIPT, REACT, NEXT.JS,
MONGODB, AND DOCKER

MARTIN KRAUSE



PRAISE FOR *THE COMPLETE DEVELOPER*

“*Modern Full Stack Development* . . . takes you through the crowded JavaScript landscape and teaches you how to build a modern sample application with containerization, authentication, and tests—a great resource for anyone starting out in web development.”

—BRADLEY SMITH, AUTHOR OF *DEVOPS FOR THE DESPERATE*

“It’s really quite astounding how many different complementary technologies you’ll understand by the end of this book!”

—NICK MORGAN, AUTHOR OF *JAVASCRIPT CRASH COURSE*

THE COMPLETE DEVELOPER

Master the Full Stack with
TypeScript, React, Next, JS,
MongoDB, and Docker

by Martin Krause



no starch
press®

San Francisco

THE COMPLETE DEVELOPER. Copyright © 2024 by Martin Krause.

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

First printing

28 27 26 25 24 1 2 3 4 5

ISBN-13: 978-1-7185-0328-1 (print)

ISBN-13: 978-1-7185-0329-8 (ebook)



Published by No Starch Press[®], Inc.
245 8th Street, San Francisco, CA 94103
phone: +1.415.863.9900
www.nostarch.com; info@nostarch.com

Publisher: William Pollock
Managing Editor: Jill Franklin
Production Manager: Sabrina Plomitallo-González
Production Editor: Jennifer Kepler
Developmental Editor: Frances Saux
Cover Illustrator: Gina Redman
Interior Design: Octopod Studios
Technical Reviewer: Quentin Hartman
Copyeditor: Audrey Doyle
Proofreader: Sharon Wilkey
Indexer: BIM Creatives, LLC

Library of Congress Control Number: 2023033924

For customer service inquiries, please contact info@nostarch.com. For information on distribution, bulk sales, corporate sales, or translations: sales@nostarch.com. For permission to translate this work: rights@nostarch.com. To report counterfeit copies or piracy: counterfeit@nostarch.com.

No Starch Press and the No Starch Press logo are registered trademarks of No Starch Press, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of this work, neither the author nor No Starch Press, Inc. shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it.

To true friends and partners. We run on caffeine and gasoline.

About the Author

Martin Krause has been making websites professionally for more than two decades. He has served as an engineering manager at Publicis Sapient and as a senior frontend architect at Razorfish, creating cutting-edge microsites and leading frontend teams on large-scale projects for Fortune 500 companies. As a certified scuba diving professional and avid traveler, he goes on frequent adventures above and below sea level. You can find him at <https://mkrause.info>, and he is @martinkr.xyz on Bluesky.

About the Technical Reviewer

In his nearly 25-year career in technology, Quentin Hartman has managed telecom systems, data centers, and public and private clouds and has acted as a sysadmin, a database administrator, a network engineer, and an incident responder. As a leader, he has advised tiny startups and Fortune 500 companies and run DevOps, QA, and development teams. He is passionate about social-impact projects that use open source tools. He lives near Denver with his family and can often be found building things, cooking, or wandering the woods. He is @qhartman on X.

BRIEF CONTENTS

Acknowledgments	xix
Introduction	xxi
PART I: THE TECHNOLOGY STACK.	1
Chapter 1: Node.js	3
Chapter 2: Modern JavaScript.	15
Chapter 3: TypeScript.	33
Chapter 4: React	53
Chapter 5: Next.js	69
Chapter 6: REST and GraphQL APIs.	93
Chapter 7: MongoDB and Mongoose.	115
Chapter 8: Testing with the Jest Framework	129
Chapter 9: Authorization with OAuth	157
Chapter 10: Containerization with Docker	173
PART II: THE FULL-STACK APPLICATION	183
Chapter 11: Setting Up the Docker Environment	185
Chapter 12: Building the Middleware.	195
Chapter 13: Building the GraphQL API.	207
Chapter 14: Building the Frontend	215
Chapter 15: Adding OAuth	231
Chapter 16: Running Automated Tests in Docker	253

Appendix A: TypeScript Compiler Options	259
Appendix B: The Next.js app Directory.	263
Appendix C: Common Matchers	289
Index	295

CONTENTS IN DETAIL

ACKNOWLEDGMENTS	xix
------------------------	------------

INTRODUCTION	xxi
---------------------	------------

Who Should Read This Book?	xxii
What's in This Book?	xxii
The Parts of a Full-Stack Application	xxiv
The Frontend	xxiv
The Middleware	xxv
The Backend.	xxv
A Brief History of JavaScript and Full-Stack Development.	xxvi
Setting Up.	xxvi

PART I: THE TECHNOLOGY STACK **1**

1	
NODE.JS	3

Installing Node.js.	4
Working with npm	4
The package.json File.	4
Required Fields	5
Dependencies.	6
Development Dependencies	6
The package-lock.json File.	6
Creating a Project	8
Initializing a New Module or Project	8
Installing the Dependencies	8
Installing the Development Dependencies.	9
Auditing the package.json File	10
Cleaning Up the node_modules Folder	11
Updating All Packages	11
Removing a Dependency	11
Installing a Dependency.	11
Using npx to Execute a Script Only Once.	12
Exercise 1: Build a "Hello World" Express.js Server	13
Setting Up	13
Writing the Server Code	13
Summary	14

2	
MODERN JAVASCRIPT	15

ES.Next Modules	15
Using Named and Default Exports	16
Importing Modules	17

Declaring Variables	17
Hoisted Variables	18
Scope-Abiding Variables	19
Constant-Like Data	20
Arrow Functions	20
Writing Arrow Functions	21
Understanding Lexical Scope	21
Exploring Practical Use Cases	22
Creating Strings	22
Asynchronous Scripts	24
Avoiding Traditional Callbacks	24
Using Promises	25
Simplifying Asynchronous Scripts	26
Looping Through an Array	27
Dispersing Arrays and Objects	27
Exercise 2: Extend Express.js with Modern JavaScript	29
Editing the package.json File	29
Writing an ES.Next Module with Asynchronous Code	29
Adding the Modules to the Server	30
Summary	31

3

TYPESCRIPT

33

Benefits of TypeScript	34
Setting Up TypeScript	36
Installation in Node.js	36
The tsconfig.json File	37
Dynamic Feedback with TypeScript	38
Type Annotations	38
Declaring a Variable	39
Declaring a Return Value	39
Declaring a Function's Parameters	39
Built-in Types	40
Primitive JavaScript Types	40
The union Type	41
The array Type	41
The object Type	42
The tuple Type	42
The any Type	43
The void Type	43
Custom Types and Interfaces	44
Defining Custom Types	44
Defining Interfaces	45
Using Type Declaration Files	45
Exercise 3: Extend Express.js with TypeScript	46
Setting Up	46
Creating the tsconfig.json File	47
Defining Custom Types	47
Adding Type Annotations to the routes.ts File	48
Adding Type Annotations to the index.ts File	49
Transpiling and Running the Code	50
Summary	51

4		
REACT		53
The Role of React		53
Setting Up React		55
The JavaScript Syntax Extension		56
An Example JSX Expression		56
The ReactDOM Package		57
Organizing Code into Components		57
Writing Class Components		59
Providing Reusable Behavior with Hooks		61
Working with Built-in Hooks		62
Managing the Internal State with useState		62
Handling Side Effects with useEffect		62
Sharing Global Data with useContext and Context Providers		63
Exercise 4: Create a Reactive User Interface for the Express.js Server		64
Adding React to the Server		64
Creating the Endpoint for the Static HTML File		66
Running the Server		66
Summary		67
5		
NEXT.JS		69
Setting Up Next.js		70
Project Structure		71
Development Scripts		72
Routing the Application		72
Simple Page Routes		73
Nested Page Routes		73
API Routes		75
Dynamic URLs		77
Styling the Application		78
Global Styles		79
Component Styles		79
Built-in Next.js Components		80
The next/head Component		80
The next/link Component		81
The next/image Component		82
Pre-rendering and Publishing		83
Server-Side Rendering		84
Static Site Generation		86
Incremental Static Regeneration		87
Client-Side Rendering		88
Static HTML Exporting		89
Exercise 5: Refactor Express.js and React to Next.js		89
Storing Custom Interfaces and Types		90
Creating the API Routes		90
Creating the Page Routes		90
Running the Application		91
Summary		91

6		
REST AND GRAPHQL APIS		93
REST APIs		94
The URL		95
The Specification		95
State and Authentication		97
HTTP Methods		98
Working with REST		99
Reading Data		99
Updating Data		100
GraphQL APIs		101
The Schema		101
The Resolvers		103
Comparing GraphQL to REST		106
Over-Fetching		106
Under-Fetching		107
Exercise 6: Add a GraphQL API to Next.js		108
Creating the Schema		108
Adding Data		109
Implementing Resolvers		109
Creating the API Route		110
Using the Apollo Sandbox		111
Summary		113
7		
MONGODB AND MONGOOSE		115
How Apps Use Databases and Object-Relational Mappers		116
Relational and Non-Relational Databases		116
Setting Up MongoDB and Mongoose		117
Defining a Mongoose Model		118
The Interface		118
The Schema		118
The Model		119
The Database-Connection Middleware		120
Querying the Database		121
Creating a Document		121
Reading a Document		122
Updating a Document		122
Deleting a Document		123
Creating an End-to-End Query		123
Exercise 7: Connect the GraphQL API to the Database		125
Connecting to the Database		125
Adding Services to GraphQL Resolvers		126
Summary		127
8		
TESTING WITH THE JEST FRAMEWORK		129
Test-Driven Development and Unit Testing		130
Using Jest		130
Creating an Example Module to Test		131

Anatomy of a Test Case	132
Arrange	132
Act	133
Assert	133
Using TDD.	135
Refactoring Code	136
Evaluating Test Coverage	138
Replacing Dependencies with Fakes, Stubs, and Mocks	139
Creating a Module with Dependencies	140
Creating a Doubles Folder	141
Using a Stub.	142
Using a Fake	142
Using a Mock	143
Additional Types of Tests	144
Functional Tests	144
Integration Tests	144
End-to-End Tests	145
Snapshot Tests	145
Exercise 8: Add Test Cases to the Weather App	146
Testing the Middleware with Spies	146
Creating Mocks to Test the Services	148
Performing an End-to-End Test of the REST API	151
Evaluating the User Interface with a Snapshot Test	153
Summary	156

9 AUTHORIZATION WITH OAUTH 157

How OAuth Works.	158
Authentication vs. Authorization	158
The Role of OAuth.	159
Grant Types	159
Bearer Tokens	160
The Authorization Code Flow	161
Creating a JWT Token	163
The Header	163
The Payload	163
The Signature	166
Exercise 9: Access a Protected Resource	168
Setting Up the Client	168
Logging In to Receive the Authorization Grant	170
Using the Authorization Grant to Get the Access Token	171
Using the Access Token to Get the Protected Resource	172
Summary	172

10 CONTAINERIZATION WITH DOCKER 173

The Containerization Architecture	174
Installing Docker	174

Creating a Docker Container	174
Writing the Dockerfile	175
Building the Docker Image	176
Serving the Application from the Docker Container	177
Locating the Exposed Docker Port	177
Interacting with the Container	178
Creating Microservices with Docker Compose	178
Writing the docker-compose.yml File	179
Running the Containers	180
Rerunning the Tests	181
Interacting with Docker Compose	182
Summary	182

PART II: THE FULL-STACK APPLICATION 183

11 SETTING UP THE DOCKER ENVIRONMENT 185

The Food Finder Application	186
Building the Local Environment with Docker	186
The Backend Container	186
The Frontend Container	189
Summary	193

12 BUILDING THE MIDDLEWARE 195

Configuring Next.js to Use Absolute Imports	196
Connecting Mongoose	196
Writing the Database Connection	197
Fixing the TypeScript Warning	198
The Mongoose Model	199
Creating the Schema	199
Creating the Location Model	201
The Model's Services	202
Creating the Location Service's Custom Types	203
Creating the Location Services	203
Testing the Services	206
Summary	206

13 BUILDING THE GRAPHQL API 207

Setting Up	208
The Schemas	208
The Custom Types and Directives	208
The Query Schema	209
The Mutation Schema	209
Merging the Typedefs into the Final Schema	209
The GraphQL Resolvers	210
Adding the API Endpoint to Next.js	212
Summary	214

14	BUILDING THE FRONTEND	215
Overview of the User Interface		215
The Start Page		216
The List Item		216
The Locations List		218
The Page		219
The Global Layout Components		222
The Logo		222
The Header		223
The Layout		224
The Location Details Page		227
The Component		227
The Page		228
Summary		230
15	ADDING OAUTH	231
Adding OAuth with next-auth		231
Creating a GitHub OAuth App		232
Adding the Client Credentials		232
Installing next-auth		233
Creating the Authentication Callback		233
Sharing the Session Across Pages and Components		235
The Generic Button Component		235
The AuthElement Component		238
Adding the AuthElement Component to the Header		241
The Wish List Next.js Page		243
Adding the Button to the Location Detail Component		244
Securing the GraphQL Mutations		247
Summary		252
16	RUNNING AUTOMATED TESTS IN DOCKER	253
Adding Jest to the Project		254
Setting Up Docker		254
Writing Snapshot Tests for the Header Element		256
Summary		257
A	TYPESCRIPT COMPILER OPTIONS	259
B	THE NEXT.JS APP DIRECTORY	263
Server Components vs. Client Components		264
Server Components		264
Client Components		265

Rendering Components	266
Fetching Data	266
Static Rendering	267
Dynamic Rendering	268
Exploring the Project Structure	269
Updating the CSS	271
Defining a Layout	273
Adding the Content and Route	275
Catching Errors	277
Showing an Optional Loading Interface	279
Adding a Server Component That Fetches Remote Data	281
Completing the Application with the Navigation	284
Replacing API Routes with Route Handlers	285

C
COMMON MATCHERS **289**

Built-in Matchers	289
The JEST-DOM Matchers	292

INDEX **295**