

Modern Full-Stack Development

Using TypeScript, React, Node.js,
Webpack, and Docker

Frank Zammetti

Apress®

Modern Full-Stack Development

Frank Zammetti
Pottstown, PA, USA

ISBN-13 (pbk): 978-1-4842-5737-1
<https://doi.org/10.1007/978-1-4842-5738-8>

ISBN-13 (electronic): 978-1-4842-5738-8

Copyright © 2020 by Frank Zammetti

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Louise Corrigan
Development Editor: James Markham
Coordinating Editor: Nancy Chen

Cover designed by eStudioCalamar

Cover image designed by Ekrulila from Pexels

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484257371. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*Dedicated to Traci, Andrew, and
Ashley – the only people I want to share
a shed in the woods with when the zombies come!*

*(Okay, maybe not the “only” ones...
but they get the good canned beans.)*

Table of Contents

About the Authorxv

About the Technical Reviewerxvii

Acknowledgmentsxix

Introductionxxi

Chapter 1: Server-Side Action: Node and NPM 1

 Of JavaScript Runtimes and Building (Mostly) Servers..... 2

 First Baby Steps with Node: Installation..... 5

 More Useful: Executing JavaScript Source Files 7

 Node’s Partner in Crime: NPM 8

 A Few More NPM Commands..... 10

 Initializing a New NPM/Node Project..... 11

 Adding Dependencies 13

 A Quick Aside: Semantic Versioning 14

 Fisher Price’s “My First Node Web Server” 15

 Bonus Example..... 18

 Summary..... 20

Chapter 2: A Few More Words: Advanced Node and NPM 21

 NPM: More on package.json..... 21

 NPM: Other Commands 25

 Auditing Package Security 26

 Deduplication and Pruning 26

 Finding/Searching for Packages sans Browser 27

TABLE OF CONTENTS

Updating Packages..... 28

Publishing/Unpublishing Packages 28

Node: Standard Modules 29

File System (fs)..... 30

HTTP and HTTPS (http and https) 32

OS (os)..... 34

Path (path)..... 35

Process..... 36

Query Strings (querystring) 37

URL (url) 38

Utilities (util) 39

The Rest of the Cast 40

Summary 42

Chapter 3: Client-Side Adventures: React 43

A Brief History of React..... 43

Yeah, Okay, History Nerd, That’s All Great, but What IS React?! 46

The Real Star of the Show: Components..... 48

Components Need Info: Props 55

Components (Sometimes) Need Memory: State..... 57

Making Them Look Good: Style 60

In the End, Why React? 62

Summary 63

Chapter 4: A Few More Words: Advanced React 65

A Better Way to Write React Code: JSX 65

Yeah, Okay, So What Does It LOOK LIKE?! 66

A Slight Detour into Babel Land 68

Compile JSX 71

And Now, Put It All Together 72

Whither Props?	76
Default Props.....	77
Typing Props.....	80
Component Lifecycle	83
Summary	86
Chapter 5: Building a Strong Foundation: TypeScript.....	87
What Is TypeScript?	87
Jumping into the Deep End	89
Beyond the Playground	91
Configuring TypeScript Compilation	92
The Nitty Gritty: Types.....	93
String.....	94
Number	94
Boolean	95
Any	95
Arrays.....	95
Tuples.....	96
Enums	97
Function	98
Object	99
Null, Void, and Undefined	99
Custom Type Aliases.....	101
Union Types	102
TypeScript == ES6 Features for “Free”!	103
The let and const Keywords	103
Block Scope	103
Arrow Functions	104
Template Literals.....	105
Default Parameters	105

TABLE OF CONTENTS

- Spread and Rest (and as an Added Bonus: Optional Arguments) 106
- Destructuring 107
- Classes 109
- Summary 116
- Chapter 6: A Few More Words: Advanced TypeScript..... 119**
 - Interfaces 119
 - Argument/Object Interfaces 120
 - Methods in Interfaces 122
 - Interfaces and Classes 123
 - Extending Interfaces 124
 - Namespaces and Modules 125
 - Namespaces..... 125
 - Modules..... 129
 - Decorators..... 131
 - Decorator Factories 134
 - Third-Party Libraries 135
 - Debugging TypeScript Apps..... 136
 - Source Maps 137
 - Summary 140
- Chapter 7: Tying It Up in a Bow: Webpack..... 141**
 - What’s a Bundle, and How Do I Make One?..... 142
 - What’s Webpack All About? 143
 - Dependency Graph 143
 - Entry 144
 - Output..... 144
 - Loaders 145
 - Plugins 146
 - Modes..... 147
 - Browser Compatibility 148

Getting Started with Webpack.....	148
Getting More Complex.....	150
Configuration.....	153
Using Modules.....	156
Wither TypeScript?	157
Summary.....	159
Chapter 8: Delivering the Goods: MailBag, the Server.....	161
What Are We Building?	161
Basic Requirements	162
Setting Up the Project	163
Source File Rundown	164
Adding Node Modules	165
Adding Types	166
A More Convenient Development Environment.....	168
The Starting Point: main.ts.....	170
A Quick Detour: Time to Take a REST	170
Another Quick Detour: Express, for Fun and Profit.....	176
Back to the Code!	177
Gotta Know What We're Talking to: ServerInfo.ts	187
Time to Send the Mail: smtp.ts.....	189
A Quick Detour: Nodemailer	190
Another Quick Detour: Generics	190
Back to the Code!	193
Time to Get the Mail (and Other Stuff): imap.ts	195
A Quick Detour: emailjs-imap-client and mailparser	195
Back to the Code!	196
Reach Out and Touch Someone: contacts.ts	204
A Quick Detour: NoSQL.....	204
Another Quick Detour: NeDB	207

TABLE OF CONTENTS

Back to the Code! 209

Testing It All 212

 Optional Tooling 215

Suggested Exercises 216

Summary 217

Chapter 9: Delivering the Goods: MailBag, the Client 219

 What Are We Building? 219

 Basic Requirements 223

 Setting Up the Project 224

 Source File Rundown 228

 The Starting Point: index.html 230

 The Starting Point, Redux: main.tsx 230

 A Quick Detour: State'ing the Obvious 231

 Back to the Code! 234

 A Bit of Configuration: config.ts..... 237

 A Worker for All Seasons 237

 A Quick Detour: AJAX 238

 Mirroring the Server Part 1: Contacts.ts..... 242

 Mirroring the Server Part 2: IMAP.ts 244

 Mirroring the Server Part 3: SMTP.ts 247

 A Cavalcade of Components 247

 A Quick Detour: Material-UI 248

 BaseLayout.tsx 253

 A Quick Detour: Functional Components 260

 Toolbar.tsx 261

 MailboxList.tsx 264

 ContactList.tsx..... 266

 ContactView.tsx 268

 MessageList.tsx 273

MessageView.tsx.....	275
WelcomeView.tsx	280
Suggested Exercises	280
Summary	281
Chapter 10: Time for Fun: BattleJong, the Server	283
What Are We Building?	284
Basic Requirements	285
Setting Up the Project	286
Some tsconfig.json Changes.....	287
Adding Node Modules	289
Adding Types	289
Source File Rundown	289
The Starting Point (the ONLY Point, in Fact!): server.ts.....	289
A Quick Detour: WebSockets	290
Back to the Code!	293
Serving the Client: The Express Server	294
Handling Messages: The WebSocket Server and Overall Game Design	294
Finishing Up the WebSocket Server	299
Of Tiles and Board Layouts.....	300
Suggested Exercises	303
Summary	304
Chapter 11: Time for Fun: BattleJong, the Client.....	305
What Are We Building?	305
Basic Requirements	306
Setting Up the Project	307
Some tsconfig.json Changes	308
Some webpack.config.js Changes	309
Adding Libraries	311

TABLE OF CONTENTS

Adding Types 311

Source File Rundown 311

The Starting Point: index.html 312

The REAL Starting Point: main.tsx 313

The Basic Layout: BaseLayout.tsx 313

Feedback and Status: ControlArea.tsx 316

 Scores 317

 Game State Messages..... 318

Where the Action Is: PlayerBoard.tsx 319

 A Quick Detour: Custom-Type Definitions..... 320

Back to the Code! 321

 The Render Process 322

Talking to the Server: socketComm.ts..... 326

 Handling Server-Sent Messages 327

 Sending Messages to the Server 329

The Main Code: state.ts..... 329

 A Few Interface for Good Measure..... 330

 The Beginning of the State Object..... 330

 Back to the Code! 332

 Message Handler Methods..... 334

 The Big Kahuna: tileClick() 335

Suggested Exercises 344

Summary..... 345

Chapter 12: Bringing the Dev Ship into Harbor: Docker 347

 An Introduction to Containers and Containerization..... 347

 The Star of the Show: Docker..... 349

 Installing Docker 350

 Your First Container: “Hello, World!” of Course! 352

A Quick Rundown of Key Docker Commands.....	353
Listing Images	353
Listing Containers	353
Starting (and Stopping) Containers	354
Remove Containers and Images.....	355
Pulling Images.....	355
Searching for Images	355
Attaching to a Container.....	356
Viewing Container Logs.....	356
Creating Your Own Image.....	357
Deploying to Docker Hub.....	362
Wrapping Up MailBag and BattleJong.....	363
Suggested Exercises	365
Summary	365
Index.....	367

About the Author

Frank Zammetti is an application architect for a major financial firm with nearly 25 years of professional experience (plus almost 15 years of nonprofessional experience before that). He is an author of, including this one, 12 technical books for Apress. Frank has also authored over two dozen certification exams for SHL as well as several independent articles for various publications. He is also a fiction author (shameless plug: look him up on Amazon if you like sci-fi) and a musician of some renown (and here, “some” should be taken to mean very little). Frank has been married for 25 years (to the same woman even!) and they have two children together.

About the Technical Reviewer

Kenneth Fukizi is a software engineer, architect, and consultant with experience in coding on different platforms internationally. Prior to dedicated software development, he worked as a lecturer for a year and was then head of IT in different organizations. He has domain experience working with technology for companies in a wide variety of sectors. When he's not working, he likes reading up on emerging technologies and strives to be an active member of the software community.

Acknowledgments

I'd like to acknowledge the exceptional team at Apress for allowing me to write not one but twelve books for them over the last decade or so. I've worked with so many great people, and it's virtually impossible not to forget someone in a list like this, but among the crew for sure are Ami Knox, Arockia Rajan Dhurai, Beth Christmas, Dulcy Nirmala Chellappa, Chris Mills, Christine Ricketts, Dominic Shakeshaft, Douglas Pundick, Frank Parnell, Frank Pohlmann, Gary Cornell, Jill Balzano, Julie Miller, Katie Stence, Kelly Gunther, Kelly Winkvist, Kevin Shea, Kim Wimpsett, Kimberly van der Elst, Krishnan Sathyamurthy, Laura Cheu, Laura Esterman, Leah Weissburg, Leonard Cuellar, Liz Welch, Louise Corrigan, Marilyn Smith, Michelle Lowman, Nancy Chen, Nicole Faraclas, Nirmal Selvaraj, Richard Dal Porto, Sharon Wilkey, Sofia Marchant, Stephanie Parker, Steve Anglin, Tina Nielsen, and Tracy Brown Collins.

As I said, I'm sure I've forgotten someone, but rest assured it was not on purpose! Thank you all for giving me a shot and allowing me to continue this journey. I most definitely could not have done it alone and I thank you all, unreservedly!

Introduction

You know, when I started learning how to program, it was a piece of cake!

You'd turn on the computer and be greeted by a nice little "Ready" prompt. You'd start typing in some code (BASIC), and eventually, you'd type run, hit Enter, and watch whatever it was you put in there spit back something (my first program was a man drawn with various keyboard characters doing jumping jacks). You might save that program to a cassette – yes, kids, a *cassette*! – and hand it to your friends if you wanted to share.

But that was it. It was just that easy.

Nowadays, though, the story is *very* different.

Writing even a trivial application now involves layers upon layers of abstractions and complexities that you must mix together, like baking the world's most complicated cake, hoping it all works in the end. Then, should you want to distribute the technological terror you've constructed (sorry, Aldearan), you've got even more challenges to overcome.

How *anyone* learns to program from scratch these days, I'm not sure!

But I'm hoping to help there!

With this book, I'm going to look at the ingredients that go into baking a cake – err, building an application – these days. To be sure, it won't cover everything. And no one recipe is necessarily the same anyway – there are lots of choices available to a developer now. But I believe I've chosen the ones most commonly used to build modern full-stack applications.

What exactly is a full-stack application anyway? Well, simply put, it's an application that includes both a front-end "client," like a web site, and a back-end "server," like, well, a *server*! We're talking about building an application that combines those two halves into a coherent whole. Most application development these days is web-based in some way (where "web" doesn't have to mean something available on the public Internet, but something built with web technologies like HTML, JavaScript, and CSS), so that's what we're going to be doing in this book.

To do this, we're going to use React, which is one of the most popular libraries for building clients out there today. And we'll use Node.js, which is a popular choice for back-end development. We're also going to use TypeScript, a language that enhances

INTRODUCTION

JavaScript on both sides of the fence to make our coding lives better. We're going to touch on several other tools that relate to all of this including Babel and Webpack. We'll talk about some strategies for connecting the client to the server including REST and WebSockets. Finally, you'll learn about packaging up applications using the very popular Docker.

All this will be combined to build two full, real applications. This way, it's not just simple, contrived examples. No, it'll be real code, practical solutions to real problems encountered in building them, and real techniques for putting all these pieces together and making sense of all this complexity.

In the end, you'll have a solid foundation for building modern full-stack applications that you can go forward with on your own to create greatness.

I mean it'll never be as great as my guy doing jumping jacks written in BASIC and loaded off a cassette, but you gotta have goals. 😊

So let's get to it. There's work to be done, learning to be accomplished, and, I hope, fun to be had!