Erich de Souza Oliveira

Mastering Reactive JavaScript

Create applications empowered with real-time data without compromising performance

Packt

Mastering Reactive JavaScript

If you're struggling to handle a large amount of data and don't know how to improve your code readability, then reactive programming is the right solution for you. It lets you describe how your code behaves when changes happen and makes it easier to deal with real-time data. This book will teach you what reactive programming is, and how you can use it to write better applications.

The book starts with the basics of reactive programming, what Reactive Extensions is, and how can you use it in JavaScript along with some reactive code using Bacon. Next, you'll discover what an Observable and an Observer are, and when to use them.

Moving on, you'll explore the RxJS API, be introduced to the problem of data traffic, and see how you can mitigate it. You'll also learn about other important operators that can help improve your code readability, and you'll see how to use transducers to compose operators.

At the end of the book, you'll get hands-on experience of using RxJS, and will create a real-time web chat using RxJS on the client and server, providing you with the complete package to master RxJS.

Things you will learn:

- Get to know the basics of functional reactive programming using RxJS
- Process a continuous flow of data with linear memory consumption
- Filter, group, and react to changes in your system
- Discover how to deal with data traffic
- Compose operators to create new operators, and use them in multiple Observables to avoid code repetition
- Explore transducers and see how they can improve your code readability
- Detect and recover from errors in Observables using the Retry and Catch operators
- Create your own reactive application: a real-time web chat

Packt›
www.packtpub.com

$ **39.99** US
£ **32.99** UK

Prices do not include local sales
Tax or VAT where applicable

ISBN 978-1-78646-338-8
53999
9 781786 463388

Mastering Reactive JavaScript

Erich de Souza Oliveira

Mastering Reactive JavaScript

Create applications empowered with real-time data without compromising performance

Erich de Souza Oliveira

Packt>

BIRMINGHAM - MUMBAI

Mastering Reactive JavaScript

Copyright © 2017 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: May 2017

Production reference: 1240517

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.
ISBN 978-1-78646-338-8

www.packtpub.com

此为试读,需要完整PDF请访问: www.ertongbook.com

Credits

Author
Erich de Souza Oliveira

Reviewer
Kevin Earley

Commissioning Editor
Amarabhab Banerjee

Acquisition Editor
Shweta Pant

Content Development Editor
Jason Pereira

Technical Editor
Leena Patil

Copy Editor
Safis Editing

Project Coordinator
Sheejal Shah

Proofreader
Safis Editing

Indexer
Francy Puthiry

Graphics
Jason Monterio

Production Coordinator
Shantanu Zagade

About the Author

Erich de Souza Oliveira is a developer with 10 years of experience, mostly in JavaScript, Scala, and Java. He has already worked for big companies and start-ups and currently works as a Chief Technology Officer for a brazilian video platform, called Winnin.

He is the author of Studio.js (microservices library for JavaScript) and can be found talking and attending South American conferences. He has a special interest in functional reactive programming, micro services, information retrieval, and recommender systems.

"I would like to give special thanks to my family and my girlfriend, for being so supportive throughout the writing of this book, Packt for the opportunity and for the amazing help they gave me with Shweta, Jason, and Leena and all editors of this book, and finally all my coworkers from Winnin for all the help and enthusiasm they had in this journey from the beginning. To write this book was a dream come true for me, and I hope you enjoy reading it as much as I enjoyed writing it."

About the Reviewer

Kevin Earley has always been passionate about technology. In high school, he picked up programming as a hobby. After high school, he entered the United States Air Force, where he was an electronic technician for about 9 years.

While in the USAF, he had multiple opportunities to program several utility applications for his shop. Later, Kevin left the USAF to begin a career in software development with his best friend, Mark Bowren.

Over the last 22 years, Kevin has worked for multiple companies across dozens of industries. Kevin has worked with many different technologies, but his primary expertise is with Microsoft products.

Among Kevin's skills and talents is writing, especially technical and factual writing. Over the years, Kevin has had many opportunities to write professionally--things such as technical manuals, tests, and some religious studies. His interest in writing is what brought Kevin to Packt. One day, when he retires, Kevin hopes to write a great deal more.

Kevin currently resides in Chicago, Illinois, with his three young children, Noelani, Eliana, and Matthew "Max" Alexander.

I'd like to thank the author of this book for writing such a wonderful and useful book. I would like to thank my children for supporting my writing projects and being so patient while "daddy has to work with his words."

www.PacktPub.com

For support files and downloads related to your book, please visit `www.PacktPub.com`.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`https://www.packtpub.com/mapt`

Get the most in-demand software skills with Mapt. Mapt gives you full access to all Packt books and video courses, as well as industry-leading tools to help you plan your personal development and advance your career.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Customer Feedback

Thanks for purchasing this Packt book. At Packt, quality is at the heart of our editorial process. To help us improve, please leave us an honest review on this book's Amazon page at `www.amazon.com/dp/1786463385`.

If you'd like to join our team of regular reviewers, you can e-mail us at `customerreviews@packtpub.com`. We award our regular reviewers with free eBooks and videos in exchange for their valuable feedback. Help us be relentless in improving our products!

此为试读,需要完整PDF请访问: www.ertongbook.com

Table of Contents

Preface 1

Chapter 1: What Does Being Reactive Mean? 7

- **The reactive paradigm** 8
- **Why do I need it?** 9
- **Installation of tools** 10
  - Node.js 10
  - bacon.js 11
  - RxJS 11
- **Your first code using reactive programming** 12
- **Summary** 17

Chapter 2: Reacting for the First Time 19

- **The bacon.js observables** 20
  - Creating our first observable 21
    - Observables from DOM events (asEventStream) 21
    - Observables from promises (fromPromise) 22
    - Observable node event emitter (fromEvent) 23
    - Observables from an array (fromArray) 24
    - Observables from an array (sequentially) 24
    - Observables from an interval (interval) 24
    - Observables from other sources 25
  - Properties 27
  - Changes in an observable 28
- **Reacting to changes** 29
  - Subscribing 29
    - Subscribing using the onValue() method 29
    - Subscribing using the log method 31
    - Subscribing using the assign method 32
  - Acting when an error occurs 33
- **Reading how an operator works** 34
- **Transforming events using bacon.js** 36
  - Reusing observables 42
  - Observables' lazy evaluation 44
- **Summary** 45

Chapter 3: A World Full of Changes - Reactive Extensions to the Rescue 47

- **RxJS observables** 48

Difference between bacon.js and RxJS observables 49
Hot and cold observables 49
Installing RxJS 50
Creating an observable 50
Creating an observable from iterable objects 51
Creating an observable from a sequence factory 53
Creating an observable using range () 55
Creating an observable using period of time 55
Creating an observable from callbacks 57
Creating an observable from a promise 58
Creating empty observables 59
Creating an observable from a single value 60
Creating an observable from a factory function 61
Creating an observable from arbitrary arguments 61
Creating an observable from an error 63
Creating observables from DOM events (or EventEmitter) 65
Creating an observable from an arbitrary source 68
Subscribing to changes (Observer) 70
RxJS Subjects 73
RxJS Disposable 76
RxJS Schedulers 79
Summary 79
Chapter 4: Transforming Data - Map, Filter, and Reduce 81
Adding operators to observables 82
The map() operator 84
The flatMap() operator 86
The filter() operator 88
The reduce() operator 90
Summary 92
Chapter 5: The World Changes Too Fast - Operators to Deal with Backpressure 95
What is backpressure? 96
Common strategies to deal with backpressure 97
Lossy strategies to deal with backpressure 97
The throttle() operator 97
The sample() operator 100
The debounce() operator 102
The pausable() observables 105
Loss-less operators to deal with backpressure 107
Buffering 107
The bufferWithCount() operator 108
The bufferWithTime() operator 112

The bufferWithTimeOrCount() operator 114
Pausable observables (with buffer) 117
Controlled observables 118
More ways to filter data 121
The first() operator 122
The take() operator 124
The takeLast() operator 125
The takeWhile() operator 126
The skip() operator 127
The skipWhile() operator 129
Summary 130
Chapter 6: Too Many Sources - Combining Observables 131
When do I need to combine observables? 132
Running observables 133
Concatenating observables 134
Using the concat() operator 134
Using the merge() operator 137
Using the concatAll() operator 141
Using the mergeAll() operator 141
Combining observables 142
Using the forkJoin() operator 142
Using the zip() operator 145
Using the zipIterable() operator 148
Summary 149
Chapter 7: Something is Wrong - Testing and Dealing with Errors 151
Dealing with errors 152
Acting when an error occurs 154
Treating an error 156
The catch() operator 156
The onErrorResumeNext() operator 159
The retry() operator 161
The mergeDelayError() operator 162
Testing our application 164
Testing in the server 164
Testing in the browser 167
Testing applications using RxJS 169
Summary 173
Chapter 8: More about Operators 175
The road so far 176
The fundamentals of RxJS 177

The map() operator versus the flatMap() operator 178
Challenge - Creating a function to filter elements using only the flatMap() operator 182
Filtering data 187
Aggregating data 189
Going beyond the basics 189
Dealing with backpressure 190
Combining observables 190
Other important operators 191
The flatMapLatest() operator 191
The flatMapFirst() operator 194
The finally() operator 195
The groupBy() operator 197
The do() operator 198
Summary 200
Chapter 9: Composition 201
What is a transducer? 202
Available transducer implementations for JavaScript 203
The transducers-js API 204
Using transducers 204
Using transducers with RxJS 207
Testing transducers 212
A performance comparison for JavaScript code using transducers 216
Summary 220
Chapter 10: A Real-Time Server 223
The web chat server 224
Creating the project 225
The architecture of the server 226
Implementing the server 228
DataSource 228
Service to send messages 229
Test implementation 230
Service implementation 233
Service to listen to new messages 234
Test implementation 235
Service implementation 237
Service to block messages from a given user 238
Test implementation 239
Service implementation 240
Service to send a command 241
Test implementation 242

Service implementation 245
Handling WebSocket connections 247
Scaling the server 250
Connecting to Redis 250
Using Redis as the DataSource for messages 252
Summary 253
Chapter 11: A Real-Time Client 255
Installing dependencies 256
The structure of our client 257
Building the application for the first time 259
Connecting to the server application 264
Using RxJS to manage user input 267
Connecting user interactions and server communication 269
Improving interaction with RxJS 275
Batching changes to the DOM 277
Testing the application 281
Summary 289
Index 291

此为试读,需要完整PDF请访问: www.ertongbook.com