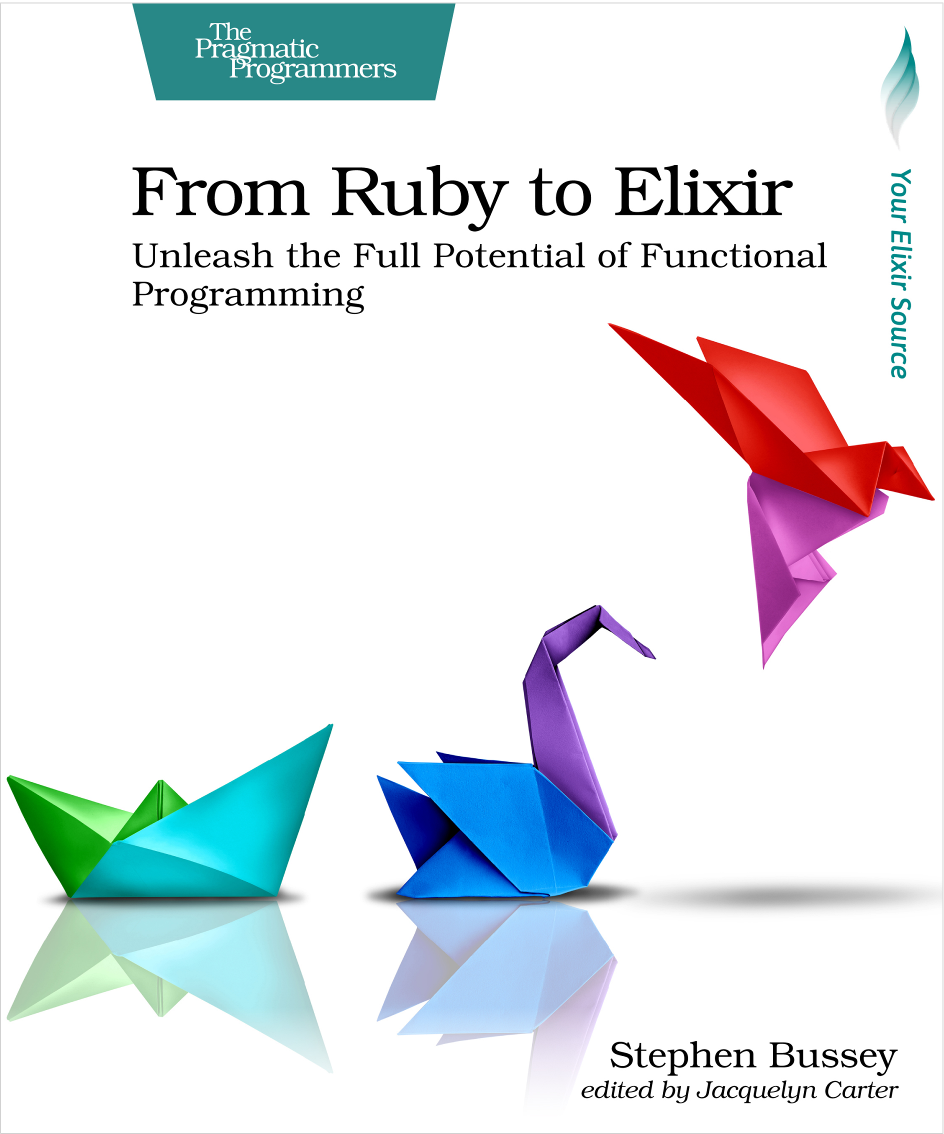

The Pragmatic Programmers

Your Elixir Source

From Ruby to Elixir

Unleash the Full Potential of Functional Programming

Stephen Bussey

edited by Jacquelyn Carter

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit https://www.pragprog.com.

Introduction

I have a confession about my journey into Elixir: it was a struggle. I could follow the guides and type out programs, but I didn't quite understand how everything fit together. Even though I was coming from Ruby, the topics were just not connecting. It wasn't until my third attempt that everything clicked.

I was missing the complete picture—the development of a real application combined with an understanding of the underlying concepts. Once I got it—it clicked quickly. I had an excellent mentor, Ben, who guided me throughout my journey.

Elixir has transformed the way that I think about, design, and code applications. The creators, community, and libraries empower me to think about code with a fresh perspective. My time with Elixir has been filled with enthusiasm, to say the least. My continued use of Elixir isn't only because it has one of the most high-tech runtimes, but also because it's simply more fun to use than other languages.

The goal of this book is to be your guide as you learn Elixir. The use of theory and practical examples will provide you with the full picture needed to succeed in your own journey—From Ruby to Elixir.

Who Should Read This Book?

This book is written for those coming from the Ruby programming language. If you're an intermediate-level Ruby programmer, then this book is perfect for you!

But this isn't the full picture! The Ruby examples in this book are likely to be relevant to any object-oriented programming language. There are some Ruby-specific comparisons, but the general approach still applies even if you're coming from Java, Python, C#, or others.

This book is ideal for those who have no Elixir knowledge or are at a beginner level. It's almost guaranteed that an Elixirist of any level will learn something new, but the majority of the content is about first steps with Elixir. In this book, you'll find Elixir code listings that walk you through theoretical examples and a practical application. We'll cover every example and project from start to finish, so you'll always know what to do next.

And if you're an experienced Elixirist reading this, maybe get a copy for your teammates.

About This Book

This book is organized into two parts. Part I is focused entirely on the fundamentals of the Elixir language. You'll learn how to read and write Elixir code during this part of the book, which will be necessary in Part II. We'll also cover slightly more advanced topics such as GenServer and the full power of pattern matching. Each section of Part I is designed to be completely standalone. The code examples are all chapter-specific, so you won't be in the dark if you decide to start out of order.

Part II is where theory meets practice. You'll write a real application that uses an API to send and receive text messages. The purpose of Part II is to introduce you to the best libraries in Elixir that are used to build robust applications. We'll cover database access, Phoenix web framework, a database-powered job system, and more. The examples in this part all build on the previous chapters, but you can start out of order with the provided code snapshots.

About the Code

Elixir is required for this book. Setup will depend on your operating system, but it's important that you're set up for success. I strongly recommend using the asdf[1] version manager to configure both Erlang and Elixir.

You'll learn about why you need different libraries in the first chapter, but it's important that you pick compatible versions of Elixir and Erlang. I recommend the versions found in Install Elixir on Your Computer, on page ?. There will be newer versions of these libraries by the time you're reading this book, so you may be able to substitute other versions. But always make sure that your Erlang version matches the OTP version of Elixir.

You'll also need to have Phoenix installed for the samples in this book. You can follow the HexDocs Installation guide[2] to get Phoenix set up. We'll cover this further at the appropriate time.

[1] https://github.com/asdf-vm/asdf

[2] https://hexdocs.pm/phoenix/installation.html

The Elixir snippets in this book aren't formatted according to the Mix formatter due to book formatting needs. You can use `mix format` to make sure that all snippets that you copy or hand-type are formatted properly.

You'll need to have PostgreSQL[3] installed for Part II. If you use Mac OS, then I recommend the postgres.app[4] installer to manage PostgreSQL.

Part II provides a mock SMS API application. This application simulates the Twilio SMS API, so you can use this book's code without setting up a real Twilio account. There will be instructions on how to set up the mock SMS API when it's time to do so.

Online Resources

The examples and source code shown in this book can be found under the source code link on the Pragmatic Bookshelf website.[5] You'll also find the sample application for Part II there.

Please report any errors or suggestions using the errata link that's available on the Pragmatic Bookshelf website.[6]

If you like this book and it serves you well, I hope that you'll let others know about it—your reviews really do help. Tweets and posts are a great way to help spread the word. You can find me on Twitter at @yoooodaaaa, or you can tweet @pragprog directly.

Stephen Bussey

May 2024

[3] https://www.postgresql.org/download/
[4] https://postgres.app
[5] https://pragprog.com/titles/sbelixir/from-ruby-to-elixir
[6] https://pragprog.com/titles/sbelixir/from-ruby-to-elixir