

Extracted from:

Concurrent Data Processing in Elixir

Fast, Resilient Applications with OTP,
GenState, Flow, and Broadway

This PDF file contains pages extracted from *Concurrent Data Processing in Elixir*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2021 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Raleigh, North Carolina

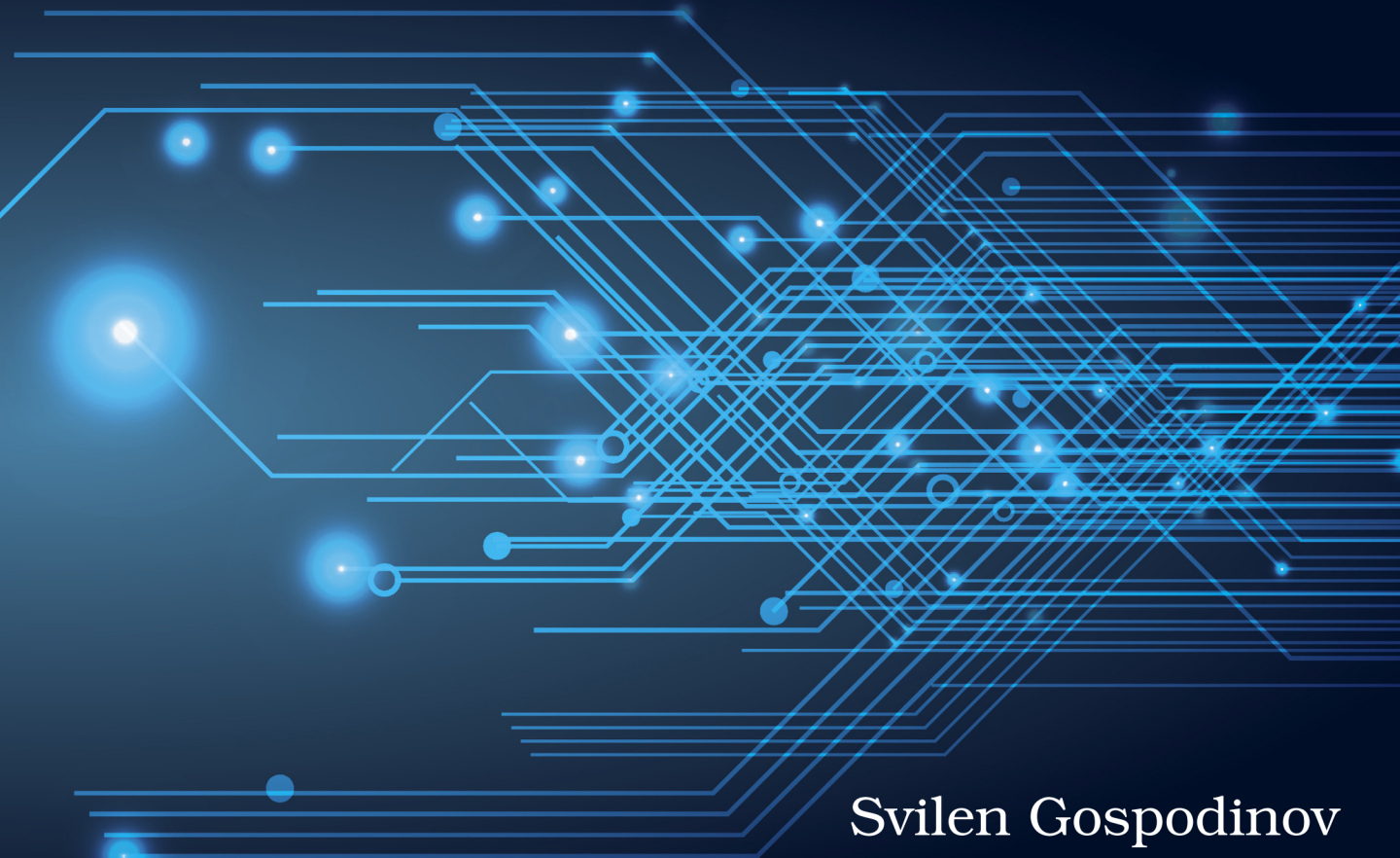
The
Pragmatic
Programmers



Your Elixir Source

Concurrent Data Processing in Elixir

Fast, Resilient Applications with
OTP, GenStage, Flow, and Broadway



Svilen Gospodinov

Foreword: *José Valim*

Series editor: *Bruce A. Tate*

Development editor: *Jacquelyn Carter*

Concurrent Data Processing in Elixir

Fast, Resilient Applications with OTP,
GenState, Flow, and Broadway

Svilen Gospodinov

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit <https://pragprog.com>.

The team that produced this book includes:

CEO: Dave Rankin

COO: Janet Furlow

Managing Editor: Tammy Coron

Series Editor: Bruce A. Tate

Development Editor: Jacquelyn Carter

Copy Editor: Karen Galle

Indexing: Potomac Indexing, LLC

Layout: Gilson Graphics

Founders: Andy Hunt and Dave Thomas

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2021 The Pragmatic Programmers, LLC.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

ISBN-13: 978-1-68050-819-2

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—August 2021

Introduction

Data processing is an essential part of many software applications. In fact, most engineers don't even think about it as something separate from programming. But if you're transforming information in some way, for example, when doing reporting, data aggregation, or analytics, then you're doing data processing.

Thanks to the Erlang Virtual Machine (also known as the BEAM), everyone who uses Elixir benefits from its amazing concurrency model, which is particularly well suited for long-running, concurrent tasks. As a result, you will find that Elixir offers more ways for performing concurrent work than other languages.

While this is a good thing, it could also be challenging to find the right tool for the job. Some tasks are excellent fit for Flow, while others are perfect for Broadway. Sometimes it's easier to just use GenStage, or even the Task module. Making the right choice means going through dozens of pages of documentation, and that would be just the beginning.

This book is here to help you navigate the world of concurrency tools available in the Elixir ecosystem. You will learn about the most popular modules and libraries and start using them in no time. You will also discover a range of new techniques that will help you simplify your product, improve the performance of your code, and make your application more resilient to errors and increased workloads.

Who Should Read This Book?

Most applications have to process data in some way and run on machines with multi-core CPUs. If your application is one of them, and data processing plays an important part in what you do, then this book is for you. Web development is very popular with Elixir, but the techniques in this book work equally well if you're using Nerves and working on embedded software.

This book is targeted at intermediate developers, who are comfortable with writing Elixir code. If you're not familiar with the language, then you should pick another book, such as [Programming Elixir 1.6 \[Tho18\]](#) by Dave Thomas, before attempting this one.

However, you don't have to be an expert in Elixir—this book will guide you through each topic with plenty of examples and helpful figures along the way. In every chapter, you will gain valuable knowledge which you can then apply to your personal or business projects.

About This Book

There are five chapters, each dedicated to a specific module or a library.

In [Chapter 1, Easy Concurrency with the Task Module, on page ?](#), you'll get started on the journey of learning how concurrency works in Elixir. It introduces the Task module, processes, timeouts, and other topics that lay the foundation for the following chapters to build upon.

In [Chapter 2, Long-Running Processes Using GenServer, on page ?](#) you'll learn about GenServer and supervisors. You will see how to create and configure GenServer processes by building a simple job-processing system. We'll introduce the Supervisor behaviour and talk about how Elixir achieves fault tolerance.

In [Chapter 3, Data-Processing Pipelines with GenStage, on page ?](#) we'll move on to data-processing pipelines. You will learn about back-pressure, and the building blocks of GenStage—producer, consumer, and producer-consumer. You will also start building your very own web scraper by putting what you have learned into practice.

In [Chapter 4, Processing Collections with Flow, on page ?](#), you'll see how you can use Flow instead of GenStage for operations like map, filter, reduce, and more. You will learn how to use it when working with large datasets and even plug it into existing GenStage data-processing pipelines.

The last chapter is [Chapter 5, Data-Ingestion Pipelines with Broadway, on page ?](#). We're going to set up a data-ingestion pipeline using RabbitMQ, but the techniques apply to other message brokers, such as Amazon SQS, Apache Kafka, and Google Cloud Pub/Sub. We will cover the various options and benefits that come with Broadway.

About the Code

You can't apply all the techniques in this book without having data to process or services to integrate with. At the same time, downloading large data sets

or signing up to third-party services is too cumbersome and not practical for the purposes of this book. That's why all projects attempt to simulate real-world cases, so you can focus on the implementation details. It also makes them easy to reproduce.

To run the code examples and snippets from the book on your computer, make sure you have Erlang and Elixir installed first. The versions used in this book are Erlang 23.0 and Elixir 1.11. If you need help installing them, check Elixir's official website¹ to see what installation options you have for your operating system.

For Chapter 5, you will also need RabbitMQ installed. You can find installation instructions on RabbitMQ's website.²

Online Resources

All examples and source code can be downloaded from the book page on the Pragmatic Bookshelf website.³ Please report any errors or suggestions you may have using the errata link.⁴

If you enjoyed reading this book, please let others know about it and spread the word. If you use Twitter, you can find me at @svileng⁵ and let me know which part of the book you liked the most. You can also tweet to @pragprog.⁶

Svilen Gospodinov

London, United Kingdom, August 2021

-
1. <https://elixir-lang.org/install.html>
 2. <https://www.rabbitmq.com/download.html>
 3. <https://pragprog.com/book/sgdpelixir>
 4. <https://pragprog.com/book/sgdpelixir/errata>
 5. <https://twitter.com/svileng>
 6. <https://twitter.com/pragprog>