Extracted from:

Real-Time Phoenix

Build Highly Scalable Systems with Channels

This PDF file contains pages extracted from *Real-Time Phoenix*, 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 © 2020 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

Real-Time Phoenix

Build Highly Scalable Systems with Channels



Stephen Bussey Series editor: Bruce A. Tate Development editor: Jacquelyn Carter

Real-Time Phoenix

Build Highly Scalable Systems with Channels

Stephen Bussey

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.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Publisher: Andy Hunt VP of Operations: Janet Furlow Executive Editor: Dave Rankin Series Editor: Bruce A. Tate Development Editor: Jacquelyn Carter Copy Editor: Sean Dennis Indexing: Potomac Indexing, LLC Layout: Gilson Graphics

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

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

Copyright © 2020 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: 978-1-68050-719-5 Encoded using the finest acid-free high-entropy binary digits. Book version: P1.0—March 2020

Introduction

I remember working on the first production-facing Elixir application I wrote—it was a real-time application to drive an innovative new feature for our platform. Excitement, curiosity, and a good bit of nervousness led me through that project. It was a trial by fire as the application would receive more requests per minute than any other part of our platform. It still stands today without much involvement needed over the past years.

I have gotten to work on many other real-time applications since that first project. The lessons I learned were sometimes hard to come by—a critical piece of the application would fail, applications would overuse resources like CPU and RAM, or I would code something in a nonoptimal way. However, I was able to leverage the strong foundations of Elixir and Phoenix to solve any problem that appeared. This book aims to collect the experience that I have gained working with Elixir and Phoenix Channels over the last several years and distill it into the parts that matter most.

Elixir has changed 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. Throughout this book, I hope to share that enthusiasm with you.

Who Should Read This Book?

Do you work on modern web applications? Do you want to build applications that are different than the traditional web model of request-in response-out? Have you started working on Elixir or Phoenix projects and want to dive deeper into the ecosystem?

If any of these questions ring a bell for you, then you will probably have a good experience with this book. If not, you will still find an interesting approach to modern applications in these pages.

This book is targeted at intermediate to advanced developers. There will be Elixir code snippets throughout each chapter, but you will be guided through each of them in order to have working examples locally. This book will not teach you Elixir—there are other books out there that are suited for that task. However, you will quickly catch on if you have a small amount of existing Elixir or Phoenix knowledge. You will walk away with a deep understanding of the real-time Phoenix stack.

About This Book

The three parts of this book build on each other to teach you about WebSockets, Phoenix Channels, and real-time application design. Part I focuses on the most important part of the real-time stack in Elixir—Phoenix Channels, WebSockets, and GenStage. We'll cover a lot of ground in these chapters, and you will gain the foundations necessary to build real-time applications.

You will leverage the foundations from Part I when we work on a real-time application in Part II. You will add real-time features to an e-commerce application that serves many users simultaneously. You will also get to work with some of the more advanced features of Phoenix, such as Phoenix Tracker.

Part III finishes the book with guidance on running real-time applications in production. The battle is only beginning when you write an application. You have to then keep it healthy and happy in production.

About the Code

Elixir is required for this book, although setup is a bit outside of this book's focus. I recommend using a version manager like asdf¹ in order to configure both Erlang and Elixir. Make sure to use a recent version of both—I used OTP 22 and Elixir 1.9 for all examples. You will also need to have Phoenix installed for the samples in this book. You can follow the HexDocs Installation guide² in order to get Phoenix set up.

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

Part II uses an application that comes already started for you. This helps keep the focus of the book on real-time features rather than the other parts of the application. You will need to download the base for the project in Part

^{1.} https://github.com/asdf-vm/asdf

^{2.} https://hexdocs.pm/phoenix/installation.html

II before you can start it. There will be instructions on how to get the project set up 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.³ You will also find the sample application for Part II there.

Please report any errors or suggestions using the errata link that is available on the Pragmatic Bookshelf website. 4

If you like this book and it serves you well, I hope that you will 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 August 2019

^{3.} https://pragprog.com/book/sbsockets/real-time-phoenix

^{4.} https://pragprog.com/book/sbsockets/real-time-phoenix