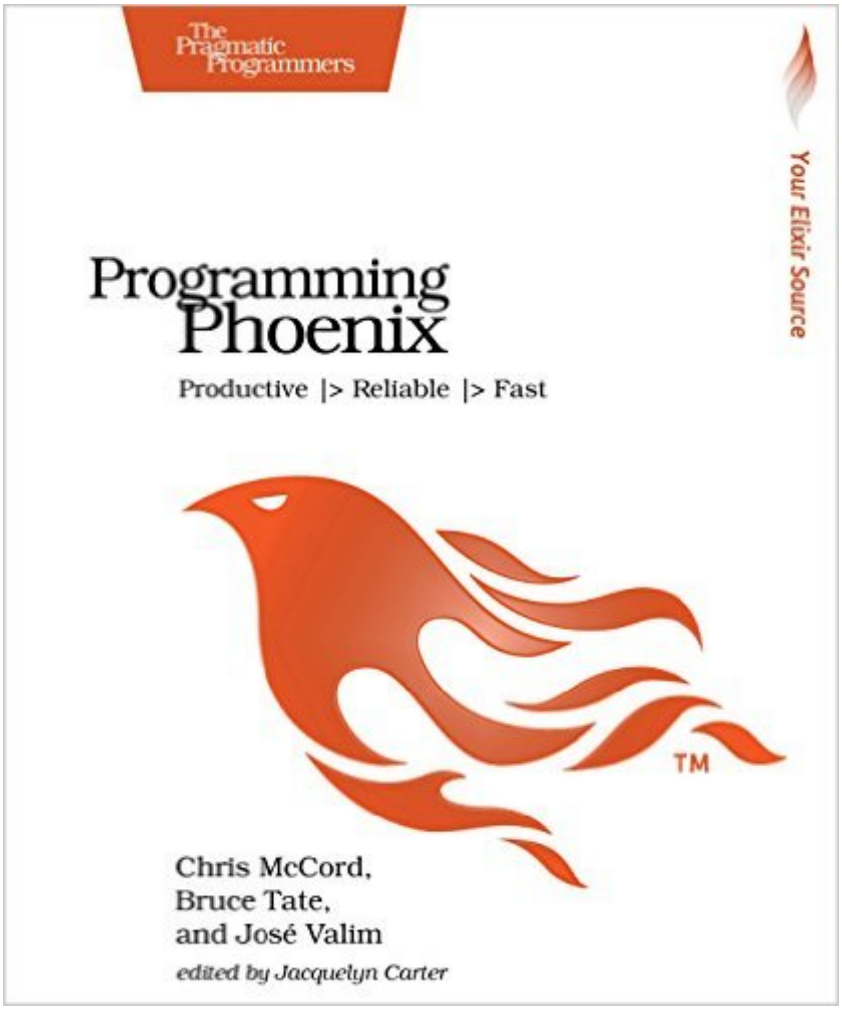
Programming Phoenix: Productive |> Reliable |> Fast



Synopsis

Don’t accept the compromise between fast and beautiful: you can have it all. Phoenix creator Chris McCord, Elixir creator Jose Valim, and award-winning author Bruce Tate walk you through building an application that’s fast and reliable. At every step, you’ll learn from the Phoenix creators not just what to do, but why. Packed with insider insights, this definitive guide will be your constant companion in your journey from Phoenix novice to expert, as you build the next generation of web applications.Phoenix is the long-awaited web framework based on Elixir, the highly concurrent language that combines a beautiful syntax with rich metaprogramming. The authors, who developed the earliest production Phoenix applications, will show you how to create code that’s easier to write, test, understand, and maintain. The best way to learn Phoenix is to code, and you’ll get to attack some interesting problems. Start working with controllers, views, and templates within the first few pages. Build an in-memory repository, and then back it with an Ecto database layer. Learn to use change sets and constraints that keep readers informed and your database integrity intact. Craft your own interactive application based on the channels API for the real-time, high-performance applications that this ecosystem made famous. Write your own authentication components called plugs, and even learn to use the OTP layer for monitored, reliable services. Organize your code with umbrella projects so you can keep your applications modular and easy to maintain.This is a book by developers and for developers, and we know how to help you ramp up quickly. Any book can tell you what to do. When you’ve finished this one, you’ll also know why to do it.What You Need:To work through this book, you will need a computer capable of running Erlang 17 or better, Elixir 1.1, or better, Phoenix 1.0 or better, and Ecto 1.0 or better. A rudimentary knowledge of Elixir is also highly recommended.

Book Information

Paperback: 300 pages

Publisher: Pragmatic Bookshelf; 1 edition (April 30, 2016)

Language: English

ISBN-10: 1680501453

ISBN-13: 978-1680501452

Product Dimensions: 7.5 x 0.6 x 9.2 inches

Shipping Weight: 1.2 pounds (View shipping rates and policies)

Average Customer Review: 5.0 out of 5 starsÂ Â See all reviewsÂ (8 customer reviews)

Best Sellers Rank: #57,747 in Books (See Top 100 in Books) #4 inÂ Books > Computers &

Technology > Programming > Parallel Programming #126 inÂ Books > Computers & Technology > Networking & Cloud Computing > Internet, Groupware, & Telecommunications #161 inÂ Books > Computers & Technology > Programming > Web Programming

Customer Reviews

Q&A with JosÃƒÂ© Valim, creator of Elixir, and Chris McCord, creator of Phoenix. JosÃƒÂ© Valim, creator of Elixir, and Chris McCord, creator of Phoenix collaborated with Bruce Tate on Programming Phoenix. They took some time out of their busy days to answer a few questions for about those two important inventions. JosÃƒÂ©, why did you create Elixir? Elixir was created out of a need for writing robust and concurrent software productively. In the last decade, our CPUs are not getting any faster, instead we are getting computers with more and more cores. This change in hardware is affecting how we write software and Elixir reflects that. JosÃƒÂ©, why did you choose the Erlang VM for Elixir? The Erlang VM is one of the few runtimes widely deployed in production that was designed for running network services. The Erlang VM provides the foundation that allows Phoenix to be extremely performant while holding 2 million open connections on a single machine. Elixir adds productive and expressive tooling to this robust runtime.

JosÃƒÂ©, was Elixir designed solely for the web? Elixir was designed to be an extensible language. Throughout the book, we will see how Phoenix and Ecto effectively extend Elixir to provide fast request routing, elegant database queries and more. However, they are two of many examples. As Elixir adoption increases, we will continue to see it being brought to new domains, like data processing and embedded software. Chris, why did you write Phoenix? I wanted a Web framework that could take on the modern Web's realtime requirements of many connected devices. I found Elixir and realized it would allow a framework to be both highly productive and have world-class performance. Thus Phoenix was born. Chris, why is it better than what you used before? Web frameworks that I had used before gave me the productivity I wanted, but I had to sacrifice performance to get a system that was a joy to program in. At the same time, most languages and frameworks I had experience in scaled very poorly when handling long-running, persistent connections. With Phoenix, we can handle 'millions' of active connections on a single server in a language that is such a joy to use.

View larger Chris, what kinds of apps can benefit? Any application requiring persistent connections is an obvious benefactor, but even standard HTML5 or JSON API applications see

huge benefits when using Phoenix. Programmers often see 'microsecond' response times when using Phoenix for HTML or JSON applications, and this directly results in heightened end-user experiences. We cover both types of applications in Programming Phoenix.

This book is certainly in the top 3 best programming books I've ever read.You end the book with a fascinating little product, having learned about a beautifully structured and very powerful Ã¢Â Âœframework Ã¢Â Â• of sorts. Even if you Ã¢Â Â™ve never programmed Elixir, this could be your introductory book, and I bet for practical-minded people it's actually ideal.I followed along about 7 of all betas, and the support from the authors was superb. They generously explained concepts, fixed my mistakes and responded to errata, all in Elixir-level response times.Recommend it very highly.

Amazing book to start learning Phoenix.This book tackles the basic fundamental stuffs of modern web development such as:1. MVC serve rendering style of web programming2. Database (Ecto)3. Authentication (over session and over token for websocket)4. Real time/websocket (Channel)6. OTP (breaking down your Phoenix app into small supervised apps to avoid monolithic design)7. How to test all of the 1-6 components aboveThis is a must have book for people interested in Phoenix and Elixir.There are a lot of magic in Phoenix, like Rails, but the concepts are easy to grasp once you repeat the book few times. Coming from NodeJS world where everyone does their own thing, and there's no convention. I'd say Phoenix embraces a nice balance, having solved many things under the hood by using macros, provide great developer experience and convention, but at the same time not overly complicated to understand.Not too mention that it solves your scalability in real time applications that are self-healing, fault tolerance, great tooling because Elixir itself is super awesome! This framework is truly an amazing framework, for those that seeks to replace Rails to embrace the new web, where everything is massively connected real-time.Amazing work Chris McCord!!

Great practical book from the creators of Phoenix. While hands on a great level of detail is provided on how Phoenix works. Unlike with many other frameworks that feel like magic black box filled with unicors Phoenix is very elegant and simple and thanx to this book I have a very clear idea of how it functions.

It is a must read for those who want to build next-gen web apps using Elixir and Phoenix

Framework.Chris has the talent in articulate what he knows to other developers. If you are looking for a practical approach to productive, reliable, high-performance web development is book is for you. This is a type of book we can get our hands dirty developing while going through the content of the book.

Want to build the next $21b WhatsApp? Then read this book. Phoenix is the framework of the future, and the quicker you jump on the bandwagon, the quicker your company will start to reap the inherent rewards.

I hadn't written any Elixir before reading this book and I was worried it might be over my head. Not at all! Everything was explained in the perfect amount of detail to make it easy to follow.

A great intro book to Phoenix!

Good explanations.

Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) Greater Swiss Mountain Dog: A Complete and Reliable Handbook (Rare Breed) Miniature Bull Terrier (Complete & Reliable Handbook) From Reliable Sources: An Introduction to Historical Methods McGraw-HIll's GED : The Most Complete and Reliable Study Program for the GED Tests