Extracted from:

Craft GraphQL APIs in Elixir with Absinthe

Flexible, Robust Services for Queries, Mutations, and Subscriptions

This PDF file contains pages extracted from *Craft GraphQL APIs in Elixir with Absinthe*, 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.

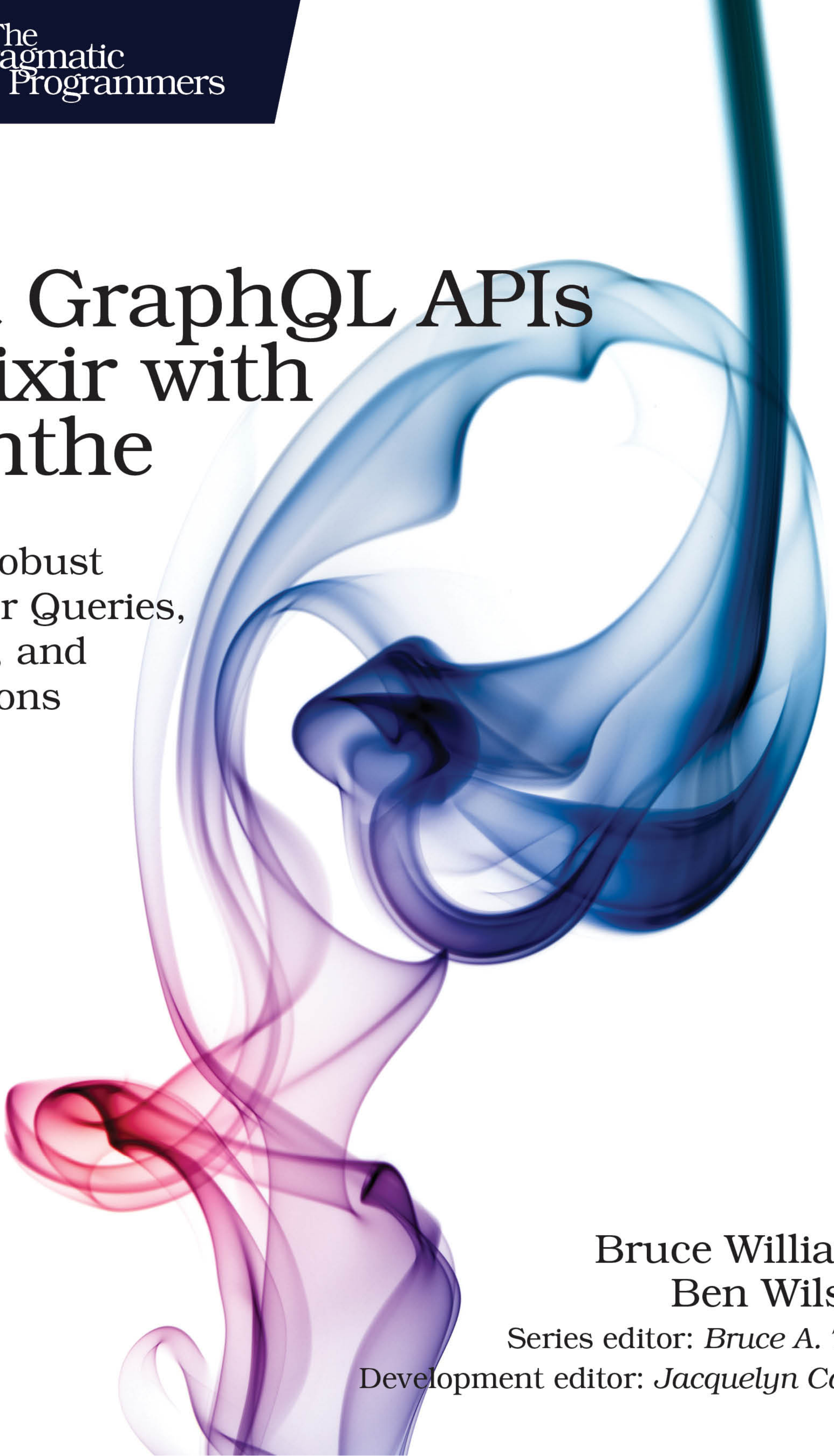
The Pragmatic Bookshelf
Raleigh, North Carolina

The Pragmatic Programmers

Your Elixir Source

Craft GraphQL APIs in Elixir with Absinthe

Flexible, Robust Services for Queries, Mutations, and Subscriptions



Bruce Williams
Ben Wilson

Series editor: *Bruce A. Tate*
Development editor: *Jacquelyn Carter*

Craft GraphQL APIs in Elixir with Absinthe

Flexible, Robust Services for Queries, Mutations, and Subscriptions

Bruce Williams

Ben Wilson

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
Managing Editor: Brian MacDonald
Supervising Editor: Jacquelyn Carter
Series Editor: Bruce A. Tate
Copy Editor: Nicole Abramowitz
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*.

ISBN-13: 978-1-68050-255-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P2.0—April 2020

Introduction

GraphQL is an exciting newcomer to the world of web service protocols. It arrived on the scene at exactly the right time for us, struggling as we were with a complicated, inflexible REST API that was wearing a little of the shine off our fancy new Elixir application in production.

We decided to give GraphQL a try.

Little did we know that along the way, we'd discover an approach to building APIs and collaborating across the backend/front-end divide that was more flexible, maintainable, and fun for our team.

We wrote Absinthe, the GraphQL toolkit for Elixir, because we thought GraphQL might be a good fit for our problem, our development process, and our platform. Since that time, a community has grown up around the project. We wrote this book because we think that it might just be a good fit for yours, too.

It's certainly a great fit for Elixir!

About This Book

This book is organized into three parts.

Part I introduces you to the core concepts behind GraphQL, then guides you through building a GraphQL API in Elixir. You'll learn about queries, mutations, and subscriptions, and end up with a functional example by the time you're through.

Part II addresses the most common security, maintainability, and performance questions that people have about GraphQL in production, focusing on building real-world solutions that turn naive implementations into rock-solid, scalable APIs that are easy to maintain.

Part III is all about using your GraphQL API. You'll learn how to support client-side GraphQL frameworks that access it over HTTP or directly from within your Elixir applications.

The book wraps up by providing a useful appendix to use as a quick reference when building your GraphQL schema and interacting with your API.

What This Book Isn't

This book isn't an introductory book on Elixir or Phoenix. While the book's code should be straightforward reading for most developers, we will cover some more advanced topics that would benefit from a firm grounding in the Elixir language and the Phoenix web framework. We recommend the following supplementary books to readers who might be coming at this book from a different background:

- *Programming Elixir 1.6 [Tho18]*
- *Programming Phoenix 1.4 [TV19]*

This book isn't a lengthy diatribe about REST or JSON APIs. While we'll take time to compare GraphQL with other technologies—building on your experience—this isn't going to be a point-by-point takedown of other choices; GraphQL is a compelling choice, and while we'll often touch on why, we'll move on to actually building things with it rather quickly.

Who This Book Is For

If you're a developer who writes modern web APIs, this book is for you. You know the rewards and challenges of building a RESTful API using JSON, working with databases, and aggregating data from remote services. This book is for you because you're in a good position to experience the "aha moment" of GraphQL full-force. You already know the problem space.

If you write Elixir (or Erlang) code, this book is for you. You use the Erlang/OTP platform for a reason—it's robust, fault-tolerant, and battle-tested. This book is for you because GraphQL is the web API and (data retrieval/mutation) technology that Elixir/Erlang deserves, and you're using a platform that can take GraphQL to the next level.

If you write Node.js code, and especially if you're already supporting GraphQL APIs for JavaScript-based web frameworks, this book is for you. Maybe you're looking for an alternative backend technology, and you've heard a lot about Elixir and Phoenix. You already know a fair bit about GraphQL, but you want to take it *further*. This book is for you because it will help you take the jump to Elixir with the confidence and knowledge to make your APIs even better.

How to Read This Book

If you're brand new to GraphQL and want the full background story, or if you've played a bit with it before but would like a solid refresher, start at the beginning. In Chapter 1, Meet GraphQL, on page ?, you'll learn about its origins and walk through the core concepts that you'll need as you read the rest of the book.

If you've used GraphQL before in another language and want to jump straight into building your first Elixir GraphQL API, then Chapter 2, Building a Schema, on page ? is the chapter for you. It's where you'll be introduced to Absinthe, the GraphQL toolkit for Elixir, and start building the GraphQL schema for our example application.

Readers who have already had some experience building GraphQL APIs using Elixir and are looking for specific answers might want to start at Part II, Publish Your API, on page ?; its chapters are focused on common challenges GraphQL developers face taking their dreams to reality. If you're ready to use your Absinthe-based GraphQL API, take a look at Part III, Use Your API, on page ?, where you'll learn about supporting fancy JavaScript GraphQL-fluent, client-side frameworks over HTTP and even building some custom processing logic to use GraphQL directly from elsewhere in your Elixir applications.

Regardless of where you start reading the book, check out the "Moving On" section at the end of each chapter. You're sure to find ideas and challenges there that will help you take your understanding of GraphQL and its use in Elixir to the next level.

About the Code

The sample code and examples included in this book are written using the Elixir programming language, and will focus on how to use Absinthe,[1] the GraphQL toolkit for Elixir. As the authors of Absinthe, we can't wait to show you the ropes.

Absinthe is distributed as a number of separate, focused packages, and we'll be using several; absinthe provides the core GraphQL functionality, absinthe_plug supports handling requests over HTTP/S, and absinthe_phoenix adds support for GraphQL subscriptions over Phoenix channels. You learn about each of these, and more related packages, as we build out the examples.

[1] http://absinthe-graphql.org

System Dependencies

The Elixir example code for the book assumes that you've installed the following system dependencies. This can be done by using your favorite package manager or downloading and installing them as pre-built binaries from the associated websites.

- Erlang[2] (v20.0+)
- Elixir[3] (v1.5.0+)
- PostgreSQL[4] (v9.4+)

The final chapter of the book covers integration with JavaScript client-side frameworks, and we recommend Node.js[5] v8.9.0+.

Online Resources

You can find all the example code for this book on its Pragmatic Bookshelf website,[6] alongside a handy community forum if you'd like to reach out for help along the way.

While we've worked hard to make our code examples and explanations bug-free and clear in every way for our readers, we've written enough software to know that we're fallible. Thanks in advance for reporting any issues that you find in the book code or text via the errata form, also conveniently found on the book website.

Welcome to the world of GraphQL—using Elixir. We can't wait to see what you build!

Bruce Williams & Benjamin Wilson

March 2018

[2] https://www.erlang.org
[3] https://elixir-lang.org
[4] https://www.postgresql.org
[5] https://nodejs.org
[6] https://pragprog.com/book/wwgraphql/craft-graphql-apis-in-elixir-with-absinthe