

Extracted from:

Creating Software with Modern Diagramming Techniques

Build Better Software with Mermaid

This PDF file contains pages extracted from *Creating Software with Modern Diagramming Techniques*, 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 © 2023 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

Creating Software with Modern Diagramming Techniques

Build Better Software with Mermaid



Ashley Peacock
Edited by Michael Swaine

Creating Software with Modern Diagramming Techniques

Build Better Software with Mermaid

Ashley Peacock

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

Development Editor: Michael Swaine

Copy Editor: L. Sakhi MacMillan

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 © 2023 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: 978-1-68050-983-0

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

Book version: P1.0—February 2023

Preface

Welcome reader, and future diagram master!

Over many years, I learned the power of being able to turn words and thoughts into diagrams. I started out like most diagrammers, using a language called UML, which we'll learn all about in this book. Instantly, I saw the power a diagram could have. I'd go as far to say it's become a superpower. The ease with which you can explain something with a diagram is like nothing else, but it always felt a bit cumbersome to do with the tools that were on offer.

Then, along came Mermaid, a tool to write diagrams as you would code and automatically render them from that code. It wasn't the first to do this, but I'd argue it's the first to go mainstream and see widespread adoption and support. Suddenly, diagramming became so much more accessible. You didn't need a special program—all you needed was a text editor. Since then, I've advocated for adopting Mermaid with the companies and individuals I work with and never looked back.

Thank you for reading this book, I hope you enjoy reading it as much as I've enjoyed writing it, and hopefully you, too, will soon have the superpower of diagramming with Mermaid.

Who Should Read This Book?

This book is aimed primarily at engineers, of any level, who want to learn how to introduce diagrams into their development workflows. No prior knowledge is needed of any programming languages, and Mermaid's syntax is incredibly simple and easy to pick up. Technical concepts are mentioned in the book, such as domain-driven design, but they all have a brief introduction and explanation.

If you want to become a better engineer, and in particular become better at conveying your thoughts and ideas, then you're in the right place. There's no easier way to convey your ideas, in my opinion, than a diagram.

What's in This Book?

We'll start out by introducing most of the key concepts used throughout the book: Mermaid and UML. We'll cover a brief history of UML, get you set up with Mermaid, and even produce your first diagram!

We'll then delve into the many use cases for diagramming, including:

- [Chapter 1, Document Your Domain, on page ?](#), where you'll be creating a domain model and we'll introduce the concept of domain-driven design to those not familiar.
- You'll learn how to easily explain different types of sequences in [Chapter 3, Visualize Application and User Flows, on page ?](#), and [Chapter 8, Visualize Code Flows, on page ?](#).
- Visualizing your architecture using the C4 Model in [Chapter 4, Model Your Architecture, on page ?](#).
- Introducing diagramming to others, and ensuring your diagrams are easily accessible and up-to-date, in [Chapter 10, Render Diagrams Using Native Support, on page ?](#), and [Chapter 11, Create a Static Site with Mermaid Diagrams, on page ?](#).

What's Not in This Book?

While I do briefly cover key technical concepts, this book isn't going to go into deep levels of detail on anything except diagramming. For example, in [Chapter 9, Design and Refactor Your Applications, on page ?](#), we'll touch on concepts such as classes, interfaces, and inheritance, and I do provide brief introductions to them, but you'll need to find other sources for a more in-depth understanding. In some cases, I recommend additional reading if you're interested to learn more.

How to Read This Book

The book is structured in a similar way to how I approach software development, in particular when it comes to building a new service or undertaking a new project. By no means do I always follow this order, and I regularly use diagrams ad-hoc when working on individual pieces of work or explaining concepts to colleagues. It felt like the natural order for the book, but don't feel the need to follow it exactly in order when it comes to using diagrams. Once you get into the habit of diagramming, you'll learn to see opportunities to use a diagram to improve a situation.

The code samples throughout, unless otherwise stated, are usually the full syntax for the diagram at each stage. Through each chapter, we'll build the diagram up slowly, so you can always take the markup from any given sample and render a diagram from it, or even edit it to get a feel for the language.

Online Resources

The source code for each code sample is available online at the Pragmatic Bookshelf website. Please feel free to leave feedback by submitting errata¹ entries, and you can get in touch to ask any questions or give feedback via email or Twitter.

- Email: ashley@technicalbookclub.com
- Twitter: [@_ashleypeacock](https://twitter.com/_ashleypeacock)

Now, let's get diagramming with Mermaid!

1. <https://devtalk.com/books/creating-software-with-modern-diagramming-techniques/errata>