

Extracted from:

# Ruby on Rails Background Jobs with Sidekiq

Run Code Later without Complicating Your App

This PDF file contains pages extracted from *Ruby on Rails Background Jobs with Sidekiq*, 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

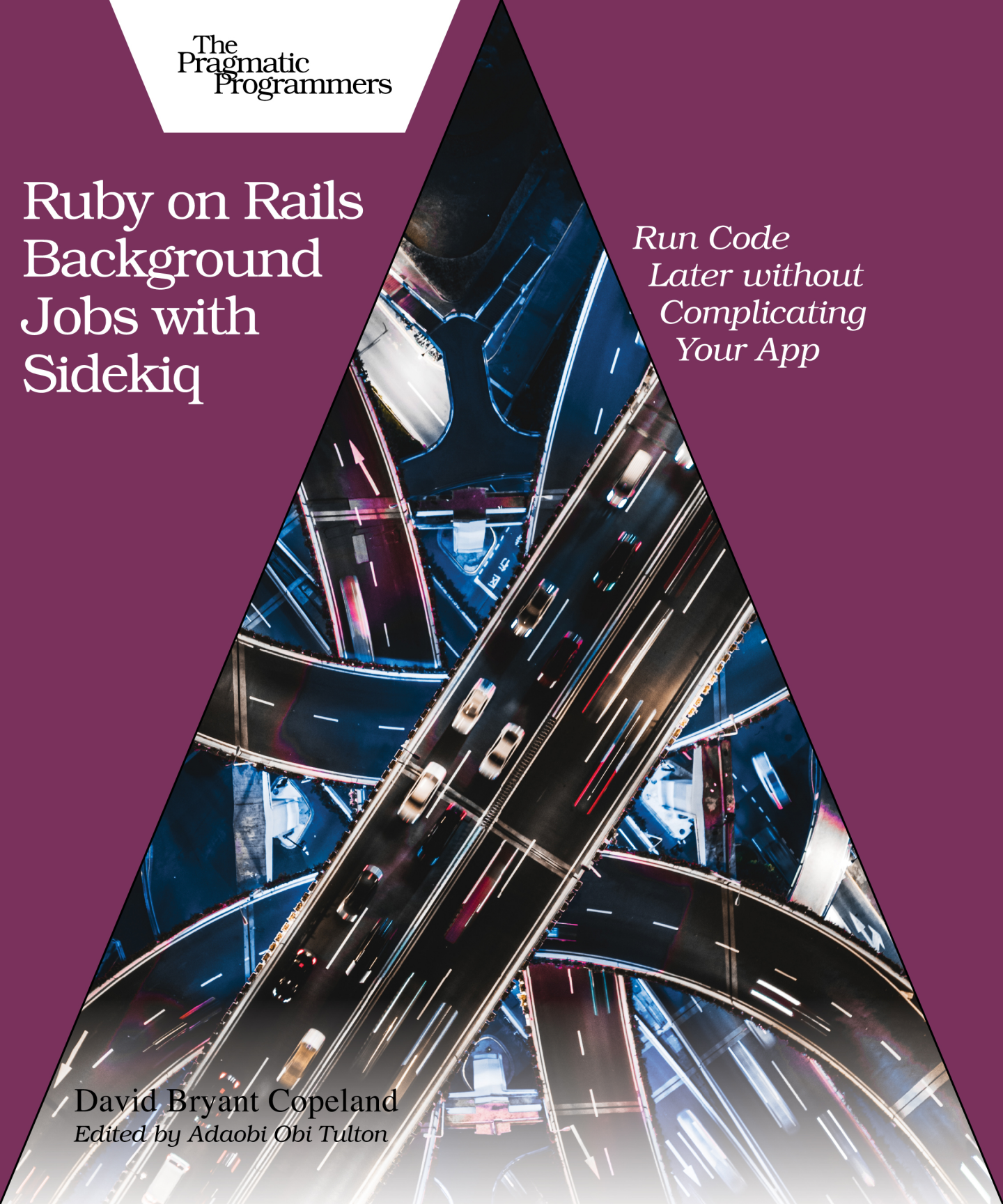
Dallas, Texas

The  
Pragmatic  
Programmers

# Ruby on Rails Background Jobs with Sidekiq

*Run Code  
Later without  
Complicating  
Your App*

David Bryant Copeland  
*Edited by Adaobi Obi Tulton*





# Ruby on Rails Background Jobs with Sidekiq

Run Code Later without Complicating Your App

David Bryant Copeland

The Pragmatic Bookshelf

Dallas, Texas



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:

Publisher: Dave Thomas

COO: Janet Furlow

Managing Editor: Tammy Coron

Development Editor: Adaobi Obi Tulton

For sales, volume licensing, and support, please contact [support@pragprog.com](mailto:support@pragprog.com).

For international rights, please contact [rights@pragprog.com](mailto: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: 979-8-88865-036-3

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

Book version: P1.0—October 2023

# Introduction

---

Sidekiq is a reliable, well-supported, high-performing way to run code in the background of any Rails app. According to Planet Argon's 2022 Ruby on Rails Survey,<sup>1</sup> 58 percent of the respondents use it for their background job needs, with Delayed Job being the next most popular at only 14 percent. This is no accident. Whether you need to offload slow code out of a controller action, or make a flaky API call more fault tolerant, Sidekiq is the way to do that.

Even though you just need a few lines of code to run a Sidekiq job, there is quite a bit of complexity hiding in the shadows. A Rails app with Sidekiq jobs is really a distributed system. It has all the same challenges as a massive microservices-based architecture. Your jobs will fail and you need to know when they do. Your code will need to evolve to allow these failures to self-heal, and this requires shedding some naivete about software design.

Many experienced developers learn how to manage background jobs and Sidekiq through the School of Hard Knocks.<sup>2</sup> Luckily for you, this short book can let you skip at least a few semesters.

## What You'll Need

This isn't an intro to Sidekiq, so you should have some understanding of what it is and how it works. That said, you'll still be able to follow along if you are experienced with other background job systems like Resque or Delayed Job. You should know the basics of Rails, though you don't need to be an expert.

You will also need a way to run Docker, which we'll use to run the examples. The book includes two applications that allow you to simulate common failure modes when using Sidekiq. Docker will allow you to run them with a single command. If you are using Windows, you will need the Windows Subsystem for Linux, Version 2.<sup>3</sup>

1. <https://rails-hosting.com/2022/#configuration>
2. [https://en.wikipedia.org/wiki/School\\_of\\_Hard\\_Knocks](https://en.wikipedia.org/wiki/School_of_Hard_Knocks).
3. <https://learn.microsoft.com/en-us/windows/wsl/install>

## What You'll Learn

First, you'll get Sidekiq configured for a reasonable production environment in [Chapter 1, From Zero To a Sustainable Sidekiq Config, on page ?](#). After that, you'll learn how to manage failed jobs in [Chapter 2, Handle Failures by Planning Ahead, on page ?](#). In that chapter, you'll quickly learn how to know when a job fails, how not to be bothered when a transient failure eventually succeeds and, of course, how to find out about transient failures that *don't* eventually succeed.

In [Chapter 3, Safely Retry Jobs by Making Them Idempotent, on page ?](#), you'll learn how to design your code so it can be safely retried many times in a Sidekiq job, all without having more effects than desired. If you've ever had to manually refund a customer who was mistakenly double charged, this is your chapter. Finally, in [Chapter 4, Sustainable Operations and Development, on page ?](#), you'll learn how to set up useful monitoring for your Sidekiq installation, manage performance via queues and concurrency, and keep all this code organized.

When you're done, you'll be ready to handle the most common challenges everyone faces when running Sidekiq in production. Let's get started!