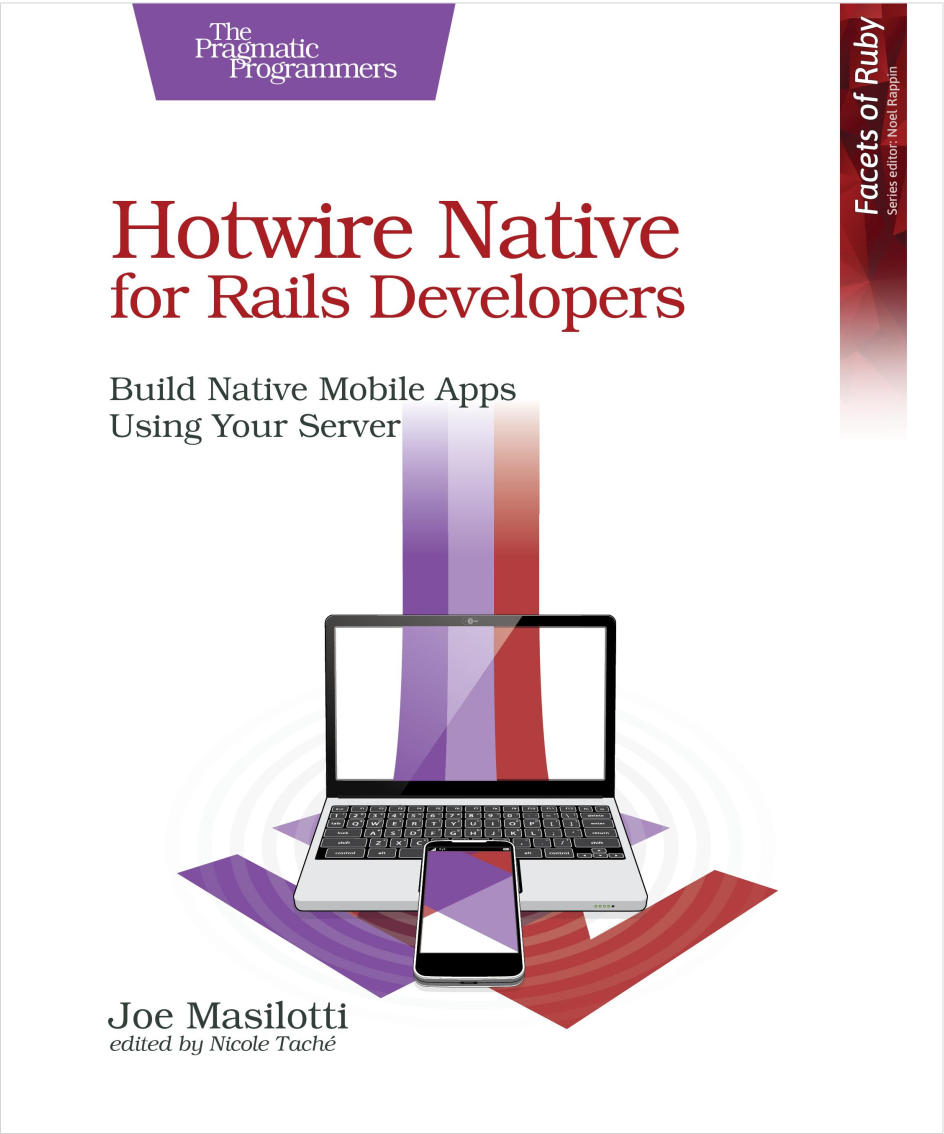

The Pragmatic Programmers

Facets of Ruby

Series editor: Noel Rappin

Hotwire Native for Rails Developers

Build Native Mobile Apps Using Your Server

Joe Masilotti

edited by Nicole Taché

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit https://www.pragprog.com.

Preface

Welcome to *Hotwire Native for Rails Developers*!

I'm Joe Masilotti, The Hotwire Native guy.[1] I help Rails developers build server-powered iOS and Android apps with Hotwire Native.

I became the Hotwire Native guy in 2016 when I was tasked with launching a 100+ screen Rails app to the App Store and Google Play. With the help of Hotwire Native, I was able to do this in a few months - *as the only developer on the team.* This would have taken me *years* if I'd gone fully native!

My world changed the day those apps went live in the app stores. And since then I've been *all in* on Hotwire Native.

Over the years, I've learned a lot about Hotwire Native. I know where it shines and where it falls short. And I'm excited to share everything I know with you.

The Problem With Native Apps

Building fully native iOS and Android apps is a *lot* of work. They are expensive to build and even more expensive to maintain. Developers have to build every screen three times: once for the web, once for iOS, and once for Android. For small teams, this just isn't viable.

Every new feature or bug fix also requires review from the app store teams. At best, this can delay a mission critical bug fix 24 hours. But it's not unheard of for reviews to take up to a week or more.

There's also the complexity of maintaining separate codebases. Fully native apps require all of your business logic to be built in Ruby, Swift, *and* Kotlin. Re-implementing the same thing three times is a recipe for inconsistencies.

So, if you're a Rails developer who wants to build mobile apps for both iOS and Android, what are you to do?

[1] https://masilotti.com

The Hybrid Solution

Enter Hotwire Native, a framework that builds hybrid mobile apps for iOS and Android. It renders HTML from your Rails server in an embedded web view, packaged inside a native app. You build your screens once, in HTML, then deploy across web, iOS, and Android *simultaneously*. Deploy to your server and you're done. No repackaging apps or resubmitting to app stores.

And Hotwire Native maximizes your skills by giving you more time to do what you do best - *writing Ruby code*. Most of your logic will remain on the server, with the apps acting as thin clients to your HTML.

While fully native apps are expensive to build and maintain, Hotwire Native apps are not. After an initial upfront cost, it's possible to not touch the native code again for years. I had the same version of a Hotwire Native app in the App Store for almost *five years*, all while receiving weekly feature updates and bug fixes via changes to the Rails codebase.

Finally, when a web experience isn't cutting it you can drop down to Swift or Kotlin. These components and screens can be upgraded to native on a case-by-case basis. It's not all or nothing. This means you can tackle that new feature when the team is ready, without having to block the launch of your initial release.

No other framework does this. *Hotwire Native gives Rails developers superpowers.*

Prerequisites

This book assumes you have a decent understanding of building applications with Ruby on Rails and Hotwire. If you can create a basic CRUD app and add a few Stimulus controllers, you will be fine. No Swift or Kotlin experience is necessary. I recommend Agile Web Development with Rails 8[2] and Modern Front-End Development for Rails[3] if you'd like to brush up on your Rails skills.

To get the most out of this book, I recommend building the apps as you follow along. To do so, you will need to run a Rails server, Xcode, and Android Studio.

[2] https://pragprog.com/titles/rails8/agile-web-development-with-rails-8/

[3] https://pragprog.com/titles/nrclient2/modern-front-end-development-for-rails-second-edition/

To run the Rails app, you'll need to have Ruby 3.3.5 and SQLite[4] installed. And you'll need Xcode[5] 16 and Android Studio[6] Ladybug or later on macOS to build the mobile apps.

How This Book is Structured

This book walks you through building a Hotwire Native app on iOS and Android. We'll start with an existing Rails codebase and quickly dive into creating new Xcode and Android Studio projects for the apps.

You'll build a small hiking tracker, which will provide a quick way to log some notes and a photo from your strolls through nature. Most of the content will be rendered from the server. We'll also progressively enhance screens to add native functionality, like maps.

Each chapter builds a new feature into the apps by introducing a new Hotwire Native concept. Most follow the same cadence: we'll first cover the Rails code, then iOS, and finally Android. If you're only interested in building for one platform, then feel free to skip the other and come back to it later.

Need Help?

If you have any questions or need help, don't hesitate to reach out. You can find me in my Discord server[7] or send me an email.[8] I'd love to hear from you.

[4] https://www.sqlite.org
[5] https://developer.apple.com/xcode/
[6] https://developer.android.com/studio
[7] https://discord.gg/t6kQrYcHfZ
[8] mailto:joe@masilotti.com