Extracted from:

Deliver Audacious Web Apps with Ember 2

Deliver Audacious
Web Apps with
Ember 2

Matthew White

edited by Katharine Dvorak

Deliver Audacious Web Apps with Ember 2

Matthew White

Pragmatic Bookshelf

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 courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *https://pragprog.com*.

The team that produced this book includes:

Katharine Dvorak (editor)
Potomac Indexing, LLC (index)
Liz Welch (copyedit)
Dave Thomas (layout)
Janet Furlow (producer)
Ellie Callahan (support)

For international rights, please contact *rights@pragprog.com*.

Introduction: Why Ember?

When you think about writing a web app, what excites you most? Do you look forward to writing framework code, or delivering features to your users? If you're like me, the features are the exciting stuff, and the framework exists to serve the features. Now, imagine if a group of developers had gone looking for best practices in creating a web app front end, and then codified what they found into a framework. Imagine if instead of worrying about writing glue code, you could spend your time on the exciting stuff. How would your work be different?

Ember is an open source JavaScript framework, maintained at emberjs.com and on GitHub,[1] that speeds the development of web apps. It provides much of the front-end infrastructure you need so that you can spend your time making a great app. If you use Ember, you'll write less code and you'll write simpler code. And simpler code is better code. By injecting object-oriented design principles into JavaScript to help you structure your code in a sensible way, Ember lets you think at a modular level. This makes your app easier to create and to maintain.

In this book, you'll learn how to create audacious web apps and do so efficiently by letting Ember do the heavy lifting. You'll use Ember's routing system to build single-page apps with rich interfaces and simple, shareable URLs. You'll create reusable UI components and templates and compose them into highly usable, sophisticated apps. You'll use Ember Data to read and write to RESTful services. You'll learn to use Ember addons to share code and use code others have shared. And you'll learn to use Ember CLI and QUnit to organize, build, and test your apps.

---

1. github.com/emberjs/ember.js/

What This Book Is, and Isn't

This book is a brisk introduction to the core features of Ember 2, Ember Data, and Ember CLI. These features rely on Node.js, npm, and Handlebars, but we won't go into detail on these dependencies.

This book is not an exhaustive reference on Ember, Ember Data, or Ember CLI. I hope you'll learn enough to pique your curiosity, but we won't cover every last Ember API. This book is also not an introduction to JavaScript or to web development. I assume you have some knowledge of JavaScript, DOM, Ajax, HTML/CSS, and RESTful services.

Who Should Read This Book?

This book is intended for web developers who are new to Ember. If you've used other frameworks such as React or AngularJS, or if you've used jQuery, some of the ideas behind Ember should be familiar to you, but if not, you should be fine.

How This Book Is Organized

In the first chapter, we'll begin by creating our local development environment, and then start working on a sample project. We'll briefly touch on many of the core features in this first chapter while we build a small, but complete, example feature.

In the next three chapters, we'll focus on the core aspects of building a user interface with Ember 2, and see how it uses routes, templates, and components to dynamically keep your interface up to date without a lot of clunky round-trips to the server. In Chapters 5 and 6, we concentrate on using models with Ember Data to access a RESTful service to obtain data and provide that data to the user interface. Chapter 7 discusses a few ways to promote code reuse within your Ember app, including utilities, mixins, and dependency injection. And we'll finish up in Chapters 8 and 9 by learning how to use Ember CLI for building, testing, and deploying apps, as well as sharing code through addons.

One thing that's a little different about this book is that each chapter builds on the work done in a previous chapter. We will work together to build a single app, starting on page one, and continuing throughout. For that reason, it's best if you read the book in order on your first pass through. Once you've read through the book, then you'll have better luck using it as a reference, but it works best as a progressive tutorial on your first read.

Together, we'll build an app called EmberNote. It's a simplified clone of the Evernote app, which is a web-based notetaking application. Using Evernote as a model, we'll use Evernote's hierarchical structure of notebooks and notes to learn how to use Ember's routing system and Ember Data's data access library. We'll build a Markdown-driven edit tool to help us prepare HTML-formatted notes, and learn about Ember's component feature in the process. We'll learn how to use Ember CLI to build, test, deploy, and share cool code from our app. Last, throughout the book, we'll learn how to use Ember CLI addons to add cool features to our app, such as authentication and component libraries.

Where to Go for Help

All of the coding examples in this book can be found online on the book's web page.[2] There you will also find a discussion forum where you can ask questions and provide feedback.

Now, let's get started!

---

2.   pragprog.com/book/mwjsember