Ember.js IN ACTION

Joachim Haagen Skeie

MANNING

Ember.js in Action

Ember.js in Action

JOACHIM HAAGEN SKEIE

MANNING
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964
Email: orders@manning.com

Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964

Development editor: Susanna Kline
Copyeditor: Lianna Wlasiuk
Proofreader: Melody Dolab
Typesetter: Marija Tudor
Cover designer: Marija Tudor

ISBN: 9781617291456
Printed in the United States of America
1 2 3 4 5 6 7 8 9 10 – MAL – 19 18 17 16 15 14

brief contents

contents

preface

Since 2006, I've worked with the development of web applications in one form or another. I started out writing a web application for Norway's largest retailer, which was based on JavaServer Pages (JSP) and later migrated toward JavaServer Faces (JSF). At the time, these technologies were great, and they served their intended purpose. Back then (before Ajax became widely used), the request-response cycle of HTTP demanded that most of the logic be put on the server side and that the server deliver complete markup, scripts, and style sheets to the browser on every request.

Even though these server-side approaches to writing web applications did their job, issues arose whenever state was a concern. Because the server is required to do the bookkeeping for all the logged-in users, managing state quickly becomes a difficult and memory-hungry task. How do you deal with users opening multiple tabs in your application and switching between them? How do you persist your session data when you want to scale the service across multiple (virtual) machines? How can you easily scale out horizontally in a consistent fashion if you have user-state stored on the server side?

As I started working on the open source project Montric (named EurekaJ at the time), I quickly decided that if I wanted to scale the application horizontally without requiring a separate session cache, I'd need to invest in learning one of the JavaScript frameworks that had started to gain momentum and popularity.

I assessed multiple frameworks and built prototypes in both Cappuccino (www.cappuccino-project.org) and SproutCore (http://sproutcore.com). Although Cappuccino had more complete tooling and promised a detailed and good-looking user interface for my application, I chose SproutCore because it enabled me to use my

existing skills, while promising an easier integration with third-party libraries. SproutCore also offered powerful views, which act as components that can be combined into a fully functional web application. Coming from a component-based, server-side framework world, these features made me feel at home with SproutCore. But despite this initial delight, I discovered that integrating SproutCore with third-party libraries wasn't as easy as I had anticipated.

When the SproutCore team was acquired and the momentum of the framework slowed down, the SproutCore community began to shift. Work on SproutCore v2.0 was well underway, but the gap between the original SproutCore and what was to become SproutCore v2.0 was widening. As a result, Ember.js was born as a fork of SproutCore.

Ember.js promised a framework that was thoroughly rooted in the technologies that drive our web experience, enabling you—as a developer—to use the skill set that you already have to develop JavaScript applications. Ember.js doesn't abstract or hide the details of your JavaScript, HTML, or CSS code, but instead it embraces these technologies to lift them up into the twenty-first century.

Needless to say, I followed along with Ember.js and decided to rewrite EurekaJ's front end with Ember.js. In the process I renamed the project Montric (http://montric.no). I've been on the Ember.js roller coaster ever since. Being part of the pre-v1.0.0 Ember.js community had its highs and its lows. With constantly changing APIs, and concepts being rethought and revisited on what felt like a weekly basis, the lows have mostly been ironed out, leaving only the highs. Deciding to write a comprehensive book about Ember.js in its pre v1.0.0 days undoubtedly exaggerated the lows in my mind.

As I'm writing this, Ember.js v1.2.0 is out, the API is stable, and the project is healthy and growing. Ember.js has become an awesome framework that lets you push the envelope on the applications you build for the web.

acknowledgments

This book is the most comprehensive and cohesive writing I've done to date, and it has been an experience with a steep learning curve. Crafting and writing this book has been an extremely humbling, difficult, and educational experience.

I wish to thank the Manning team for being willing to publish a book about Ember.js and for pushing to get the book realized as the story/journey that you're about to embark on. I would especially like to thank development editor Susanna Kline for putting up with my many missed deadlines and endless Skype questions, and for always giving great feedback on needed improvements. I also would like to thank the team of copyeditors—Lianna Wlasiuk along with Rosalie Donlon, Sharon Wilkey, and Teresa Wilson—who caught and fixed an embarrassing number of spelling and grammar mistakes throughout the text. Thanks also to proofreader Melody Dolab, compositor Marija Tudor, and project managers Mary Piergies and Kevin Sullivan.

The reviewers also made sure that the book remained on point and relevant during the various stages of its development, and I'd like to extend my gratitude for their work along the way. Thanks to Benoît Benedetti, Chetan Shenoy, Dineth Mendis, Jean-Christopher Remy, Leo Cassarani, Marius Butuc, Michael Angelo, Oren Zeev-Ben-Mordehai, Philippe Charrière, Richard Harriman, and Rob MacEachern. Finally, thanks to technical proofreader Deepak Vohra for his careful review of the manuscript shortly before it went into production.

I wish to extend a special and huge thank you to my beautiful wife, Lene, and my two amazing children, Nicolas and Aurora. Lene's support and understanding have been incredibly important during the writing of the book, especially as it consumed a

lot of my spare time during evenings and on weekends. Knowing that your family is safe, secure, and happy is of utmost importance when deciding to prioritize your spare time elsewhere.

about this book

Ember.js is the most ambitious web application framework for JavaScript. With the release of the final v1.0.0 version, after just under two years of development, the APIs have stabilized and the project has steadily moved on, quickly reaching versions 1.1.0 and 1.2.0.

Writing large, ambitious web applications is hard work. Ember.js exists because its creators wanted to build a framework that simplified and standardized the way we write applications for the web. This book aims to present the features and highlights of the framework while keeping the text driven by real-world examples.

Roadmap

The book is divided into three parts:

- *Part 1*—Ember.js fundamentals
- *Part 2*—Building ambitious web apps for the real world
- *Part 3*—Advanced Ember.js topics

Part 1 uses simple, self-contained examples to introduce you to Ember.js, its core features, and what Ember.js expects from your application:

- Chapter 1 introduces you to Ember.js and explains where the project comes from, as well as where it fits into the world of web applications. You'll learn the fundamental concepts and terminology used in Ember.js.
- Chapter 2 builds on chapter 1 by explaining the core features of Ember.js. In this chapter, you'll learn about bindings, computed properties, observers, and the Ember.js object model.

- Chapter 3 is dedicated to Ember Router, the glue that holds your entire application together.
- Chapter 4 presents Handlebars.js, the template library of choice for Ember.js applications. You'll learn the features Handlebars.js brings to the table for Ember.js applications as well as the Ember.js-specific add-ons that Ember.js adds to Handlebars.js.

Part 2 introduces the case study, Montric, which is used for the majority of the remaining chapters. This part delves into the harder parts of web application development: how to interact with the server side efficiently, both with and without Ember Data; writing custom components; and testing your Ember.js applications:

- Chapter 5 dives into how to use the third-party Ember Data library to communicate with the server side. Based on Ember Data beta 2, this chapter details what Ember Data expects from your server-side API and your Ember.js application before showing how to customize Ember Data to fit your existing server-side API.
- Chapter 6 shows how to interact with the server side without relying on a framework to help you. This chapter also shows how to build a complete CRUD data layer from scratch.
- Chapter 7 is all about custom components, a feature added late to Ember.js. Using Ember.js components, you can build atomic, self-contained components that can be reused either on their own or as building blocks of more complex components.
- Chapter 8 is dedicated to showing how to test your Ember.js application. You'll use QUnit and PhantomJS to build up a viable testing solution.

Part 3 moves into the more obscure parts of Ember.js and discusses other services and tools to facilitate application development and increase your understanding of Ember.js:

- Chapter 9 walks you through building in authentication and authorization by using a third-party authentication system. This chapter specifically implements Mozilla Persona, which is an open source solution.
- Chapter 10 eases you into the Ember run loop, called Backburner.js. This background motor drives your Ember.js application and ensures that your application views are always kept up-to-date while keeping performance high.
- Chapter 11 teaches you how to structure your Ember.js application as your source code grows and how to build, assemble, and package your application in preparation for deployment.

Who should read this book

This book is written to help you become familiar and productive with Ember.js. Depending on your background, developing JavaScript applications with a framework such as Ember.js may involve a steep learning curve. This book helps you learn the concepts of Ember.js quickly and familiarizes you with the Ember.js terminology and

application structure. This book is approachable to newcomers to Ember.js, as well as to developers with previous Ember.js experience.

As a prerequisite, this book expects you to be familiar with JavaScript as a language, as well as to have some familiarity with jQuery.

Code conventions and downloads

The following typographical conventions are used throughout this book:

- *Italic* type is used to introduce new terms.
- `Courier` typeface is used to denote code samples, as well as elements and attributes, method names, classes, interfaces, and other identifiers.
- Code annotations accompany many of the code listings and highlight important concepts.
- Some of the lines of code are long and break due to the limitations of the printed page. Because of this, line-continuation markers (➥) may be included in code listings when necessary.

This book includes many code snippets and a large amount of source code. The source code for part 1 is available via the book's GitHub pages or as a downloadable zip file from the publishers's website at www.manning.com/Ember.jsinAction. Parts 2 and 3 are based on the Montric source code, and as such, the code is available on GitHub.

Because the examples are living projects, the source code will invariably have changed since this text was written. To account for this, use the following direct links to view the source code as it was during the writing of this book:

- Chapters 1 and 2—https://github.com/joachimhs/Ember.js-in-Action-Source/tree/master/chapter1/notes
- Chapter 3—https://github.com/joachimhs/Ember.js-in-Action-Source/tree/master/chapter3/blog
- Chapters 5, 7, 8, 9, and 11—https://github.com/joachimhs/Montric/tree/Ember.js-in-Action-Branch
- Chapter 6—https://github.com/joachimhs/EmberFestWebsite/tree/Ember.js-in-Action-branch

I've tried to document as much about Ember.js as possible, while keeping the examples real and down-to-earth. Using the Montric source code led to the occasional difficulty in isolating good examples, but I think this gives the book more depth. In addition, the number of updates and changes to the book's examples give you an idea of how much Ember.js has grown since its initial release.

Author Online

The purchase of *Ember.js in Action* includes free access to a private web forum run by Manning Publications, where you can make comments about the book, ask technical questions, and receive help from the author and from other users. To access the

forum and subscribe to it, point your web browser to www.manning.com/Ember.jsin-Action. This page provides information on how to access the forum after you're registered, what kind of help is available, and the rules of conduct on the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the author can take place. It isn't a commitment to any specific amount of participation on the part of the author, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking the author some challenging questions, lest his interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

About the author

Joachim Haagen Skeie is self-employed though his company Haagen Software AS. He spends his time working as a freelance consultant and an instructor of Ember.js and RaspberryPi courses, while working on launching the products Montric (an open source application-performance monitor) and Conticious (an open source CMS API used to host Ember.js-based RIAs). Both Montric and Conticious are based on Ember.js on the front end, with Java on the back end.

Joachim has worked on the development of web applications, big and small, since 2006, primarily focusing on Java and Ember.js. He lives in Oslo, Norway, with his wife and children.

about the cover illustration

The figure on the cover of *Ember.js in Action* is captioned "A theater director in Paris." The illustration is taken from a nineteenth-century edition of Sylvain Maréchal's four-volume compendium of regional dress customs published in France. Each illustration is finely drawn and colored by hand. The rich variety of Maréchal's collection reminds us vividly of how culturally apart the world's towns and regions were just 200 years ago. Isolated from each other, people spoke different dialects and languages. In the streets or in the countryside, it was easy to identify where they lived and what their trade or station in life was just by their dress.

Dress codes have changed since then and the diversity by region, so rich at the time, has faded away. It is now hard to tell apart the inhabitants of different continents, let alone different towns or regions. Perhaps we have traded cultural diversity for a more varied personal life—certainly for a more varied and fast-paced technological life.

At a time when it is hard to tell one computer book from another, Manning celebrates the inventiveness and initiative of the computer business with book covers based on the rich diversity of regional life two centuries ago, brought back to life by Maréchal's pictures.

Part 1

Ember.js fundamentals

Ember.js is a JavaScript MVC framework that helps you organize and structure the source code for large web applications. In comparison to other popular JavaScript application frameworks, it delivers a more complete MVC pattern, while also including features to help you build applications for the web of tomorrow. It's also one of the more opinionated frameworks available, relying heavily on convention over configuration to help you structure your application.

Because of its large number of features coupled with the conventions that it expects from your application, Ember.js has a steep learning curve. Part 1 of this book, comprised of the first four chapters, eases you into the mindset of Ember.js application development, while ensuring that you build something useful right from the get-go.

The first two chapters focus on the core features that Ember.js brings to the table. Chapter 3 focuses on Ember Router, and chapter 4 focuses on the template library of choice for Ember.js developers: Handlebars.js.