



Community Experience Distilled

Express.js Blueprints

Learn to use Express.js pragmatically by creating five fun and robust real-world APIs, with a bonus chapter on Koa.js

Ben Augarten, Marc Kuo, Eric Lin, Aidha Shaikh,
Fabiano Pereira Soriani, Geoffrey Tisserand,
Chiqing Zhang, Kan Zhang

www.allitebooks.com

[PACKT] open source[®]
PUBLISHING
community experience distilled

Express.js Blueprints

Learn to use Express.js pragmatically by creating five fun and robust real-world APIs, with a bonus chapter on Koa.js

**Ben Augarten, Marc Kuo, Eric Lin, Aidha Shaikh,
Fabiano Pereira Soriani, Geoffrey Tisserand,
Chiqing Zhang, Kan Zhang**



BIRMINGHAM - MUMBAI

Express.js Blueprints

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: May 2015

Production reference: 1080515

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78398-302-5

www.packtpub.com

Credits

Authors

Ben Augarten
Marc Kuo
Eric Lin
Aidha Shaikh
Fabiano Pereira Soriani
Geoffrey Tisserand
Chiqing Zhang
Kan Zhang

Reviewers

Aristides Villarreal Bravo
Aritrik Das
John Fawcett
Ajmal Sali
Dan Williams

Commissioning Editor

Ashwin Nair

Acquisition Editor

James Jones

Content Development Editors

Priyanka Shah
Ritika Singh

Technical Editor

Deepti Tuscano

Copy Editors

Vikrant Phadke
Adithi Shetty

Project Coordinator

Izzat Contractor

Proofreaders

Simran Bhogal
Martin Diver
Safis Editing

Indexer

Priya Sane

Production Coordinator

Nitesh Thakur

Cover Work

Nitesh Thakur

About the Authors

Ben Augarten is a senior software engineer at Twitter, in the beautiful city of Boulder, Colorado, USA. He is an electrical engineering and computer science graduate from the University of California, Berkeley. He is the author of node-RESTful, a library used to generate RESTful endpoints for domain resources. Ben writes programs that are widely used, functional, and scalable. He primarily works on distributed systems, data processing, and web technologies.

My thanks to Packt Publishing for their help and support while writing this book. Also, thanks to the Axiom Zen team for their help in crafting the content. Finally, thanks to my friends and family for their support, and my co-workers for refraining from making too many Node.js jokes.

Marc Kuo has a neat freak personality, which is reflected in his clean and proficient code. As a full stack developer at Axiom Zen, he architects shipshape infrastructure and efficient model databases. He loves to hack in Common Lisp, CoffeeScript, Angular, and Node.js.

The neat freak in Marc comes from two important principles: efficiency and optimization. He is the founder of Routific, a routing solution that reduces waste in the transportation sector. In the open source field, he is the author of Alike and Look-Alike(recommendation engines for Node.js), T3 (*Ultimate Tic-Tac-Toe*), and Zenbase-angular (gulp-angular-coffee-stylus-jade boilerplate).

Infinite gratitude goes out to my wife, Suzanne Ma, the cofounder of Routific as well as my life. Thanks for always joining me on my crazy adventures. I'd also like to express (no pun intended) exponential gratitude towards Axiom Zen for giving us this unique opportunity and for always fostering creativity.

Eric Lin is a software engineer at Axiom Zen in Vancouver, British Columbia, Canada. He completed his master's degree in statistics and has been developing software professionally for 2 years. He is always looking forward to picking up new languages and technologies while having a personal interest in data analytics and predictive modeling.

My thanks to Packt Publishing for their support throughout the writing process and their flexibility on deadlines; to Axiom Zen, for the numerous opportunities they gave me to expand my knowledge in new technologies; finally, to my wife, Min-Chee Lo, for always being patient and understanding, especially on the days where I ended up working late.

Aidha Shaikh has a doctoral degree in chemistry from the University of British Columbia, where she researched enzymes that cleave blood antigens to make universal O-type blood, and published several first-authored papers. After completing an NSERC Post Doctoral Industrial R&D Fellowship, she embraced her love for coding. She recently stepped out of her lab coat and donned a coder hoodie with pride.

Aidha's research-rich past stays with her as she constantly looks for new ways to solve problems, and she loves learning new things everyday. She started off with frontend web development. She really loves to hack into Node.js and Express.js.

My deepest thanks go to Axiom Zen for the amazing opportunities, learning experiences, and creative avenues I've been given. I would also like to thank the Packt Publishing team for all their support and work on this book with us.

Fabiano Pereira Soriani does what he loves as a software developer at Axiom Zen in Santiago, Chile. He holds a bachelor's degree in computer science from the Universidade Estadual de Londrina, Brazil, and a certification in project management from Canada. He has developed software professionally for over 5 years, always focusing on new and productive web technologies, with an intense focus on the impact they have on users and other stakeholders alike. He aims for excellence in product and agile product lifecycles.

Fabiano has published open source Node.js packages and a number of how-to blog posts, ranging from backend concepts and Ruby on Rails all the way through to the cutting-edge frontend.

Thanks to the talented and patient staff at Packt Publishing for helping us instill the best book we could. It has been quite a long journey. Also, thanks to Axiom Zen, for the vision and encouragement, and allowing time for this project – this is a part of what makes this company so unique.

Finally, thanks to my companion, Asuka Kiriyama, who was kind and tolerant of the long work hours required to write the content of this book.

Geoffrey Tisserand is a full stack software engineer, who focuses on building reliable and scalable applications and services for Axiom Zen's products. He completed his master's degree in computer science at the Université de technologie in Belfort-Montbéliard, France. He is a nitpicky and detail-oriented JavaScript and Ruby ninja, who really enjoys discovering new technologies, APIs, and frameworks to play with.

A start-up enthusiast, Geoffrey is thrilled to be in an environment where he is constantly learning and improving his skills. He also loves to build side-projects and create experiments, and is always thinking about his next idea for a start-up.

Chiqing Zhang is an exceptional software architect, whose clean and simple code has leveraged scalable and maintainable systems for some of the world's top technology companies, such as Microsoft, Baidu, and AppAnnie. As a full stack developer at Axiom Zen, he is passionate about building highly reliable systems and delivering products with the best user experience. Chiqing was granted a patent for multilayer structured data operations and he has published a book on Microsoft Silverlight technologies.

Many thanks to the Packt Publishing team and the technical reviewers, whose insightful comments and kind suggestions were essential for improving the content. Thanks to Axiom Zen for this opportunity. Finally, thanks to my wife, Hanna Yang, for her constant support and patience.

Kan Zhang is an experienced software engineer with both a bachelor's degree in civil engineering and a bachelor's degree in computer science. He has gained substantial project management experience from leading personal team projects as well as previous civil engineering internships.

Kan has also worked on many Android apps, mobile games, and backend systems for various applications and services. He is currently working as a software engineer at Axiom Zen, discovering his love for new technologies, innovative products, and exciting start-ups.

My thanks for the amazing support from all the people at both Axiom Zen and Packt Publishing. And thanks to Irene Fung for her continued patience and support in everything I do.

About the Reviewers

Aristides Villarreal Bravo is a Java developer, a member of the NetBeans Dream Team, and a Java User Groups leader. He lives in Panama. He has organized and participated in various conferences and seminars related to Java, JavaEE, NetBeans, the NetBeans platform, free software, and mobile devices, both nationally and internationally. He is the author of tutorials and blogs about Java, NetBeans, and web development.

Aristides has participated in several interviews on sites about topics such as NetBeans, NetBeans DZone, and JavaHispano. He is a developer of plugins for NetBeans and the technical reviewer of a book about PrimeFaces. He is the CEO of Javscas Software Developers.

Aritrik Das is a web developer with expertise in various cutting-edge technologies used in the Web arena. He has strong analytical skills and a broad knowledge of open source technologies. He has gained skills in many web frameworks, both frontend and backend. Aritrik is an excellent problem solver, able to quickly grasp complex systems, and identify opportunities for improvement and the resolution of critical issues. He also has quite a good hold on various deployment techniques and infrastructure designs. As such, he can develop a scalable web application from the groundwork to deployment in live environments.

John Fawcett is a JavaScript application developer with over 10 years of professional development experience. In his home at Austin, Texas, USA, he organizes a small tech meetup and is an active member of the web development community. He contributes to, and is the author of, many open source projects, and is the CTO of the local start-up, www.goodybag.com. In his free time, he writes for his blog, performs visual experiments with processing, composes music, and drinks craft beer with his lovely girlfriend and partner, Courtney.

Ajmal M Sali is a technophile from Haripad, Kerala – a tourist destination. He started programming at the age of 12. He likes to help others in solving technical problems, and these problems range from setting up clustered messaging servers to fixing Tetra Pak machines. He has worked with few startups such as Sourcebits Inc. and Rightaway Inc. and has worked as a remote consultant. He has strong expertise with many frameworks and languages and focuses more on Angular.js, RubyOnRails, PHP and Android. He believes in continuous integration and deployment. He blogs at <https://ajm.al>.

Dan Williams has been programming since he was in high school. Having worked from the microcontroller level up to large-scale enterprise applications, he has now found a new home as a senior developer at Keaton Row. Developing with Node.js in the backend and React on the browser, he enjoys being fully immersed in JavaScript. Dan can often be found around Ontario giving talks and facilitating workshops on emerging technologies.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit
www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	v
Chapter 1: Building a Basic Express Site	1
Setting up Express for a static site	1
Saying Hello, World in Express	2
Jade templating	3
Local user authentication	4
User object modeling	5
Introducing Express middleware	7
Setting up passport	8
Registering users	10
Authenticating users	11
OAuth with passport	12
Adding OAuth to user model	12
Getting API tokens	13
Third-party registration and login	14
Profile pages	15
URL params	15
Profile templates	16
Testing	16
Introducing Mocha	17
Testing API endpoints	18
Automate builds and deploys	19
Introducing the Gruntfile	20
Continuous integration with Travis	21
Deploying Node.js applications	22
Summary	24

Table of Contents

Chapter 2: A Robust Movie API	25
Folder structure and organization	25
Responding to CRUD operations	27
Retrieving an actor with GET	28
Creating a new actor with POST	29
Updating an actor with PUT	30
Removing an actor with DELETE	30
Object modeling with Mongoose	32
Generating unique IDs	34
Validating your database	36
Extracting functions to reusable middleware	36
Testing	40
Summary	44
Chapter 3: Multiplayer Game API – Connect 4	45
Modeling game state with Mongoose	46
Creating a new game	49
Input validation	54
Getting the game state	56
Joining a game	58
Playing the game	61
Testing for a tie	71
Summary	73
Chapter 4: MMO Word Game	75
Gameplay	75
Real-time application overview	76
Keeping track of active users	77
Schema design	77
User schema	77
User join	78
Promises	79
The then and catch method	83
Chain multiple Promises	83
Prevent duplicates	84
User leaves the game	86
Show all active users	87
The words – Subdocuments	88
Validate input	89
Dealing with race conditions	91
Test case to test race conditions	93

Table of Contents

Socket.IO	94
Socket handshaking, user join	94
Adding and pushing updates to clients	95
Launch Socket.IO applications	97
Test Socket.IO applications with the Socket.IO client	98
Debug Socket.IO with Chrome Developer Tools	103
Summary	106
Chapter 5: Coffee with Strangers	107
Code structure	108
Defining routes	109
Persisting data	110
Exception handling	113
Naive pairing	113
Notes about tests	118
Considering user history	118
Optimizing for distance	124
E-mail follow up	126
Periodical tasks with node-cron	135
Summary	137
Chapter 6: Hacker News API on Koa.js	139
Generator syntax	140
Middleware philosophy	143
Context versus req,res	145
The link model	145
The link routes	146
Tying it together	147
Validation and error handling	148
Update route	150
Let's perform some tests	152
Parallel requests	154
Rendering HTML pages	155
Serving static assets	160
Summary	161
Appendix: Connect 4 – Game Logic	163
Index	171

Preface

APIs are at the core of every serious web application. Node.js is an especially exciting tool that is easy to use, allows you to build APIs, and develop your backend code in JavaScript. It powers the server side of web apps, including PayPal, Netflix, and Zenhub.

Express.js is the most popular framework that can be used to build on top of Node.js – it provides an essential level of abstraction to develop robust web applications. With the emergence of this minimal and flexible Node.js web application framework, creating Node.js applications has become much simpler, faster, and also requires minimal effort.

This book takes a pragmatic approach to leveraging what Express.js has to offer, introduces key libraries, and fully equips you with the skills and tools necessary to build scalable APIs from start to finish while offering subtle details and nuggets of wisdom that come from years of experience.

What this book covers

Chapter 1, Building a Basic Express Site, will provide a basic application (scaffolding), which we will use for the upcoming examples. You will get an insight into what Express applications look like.

Chapter 2, A Robust Movie API, will walk you through building a movie API that allows you to add actor and movie information to a database and connect actors to movies and vice versa.

Chapter 3, Multiplayer Game API – Connect 4, will revolve around building a multiplayer game API. We will also build the app using test-driven development with maximum code coverage.

Chapter 4, MMO Word Game, will teach you how to build a real-time application with Express and SocketIO, perform authentication for socket handshaking, and deal with race conditions using MongoDB's atomic update.

Chapter 5, Coffee with Strangers, will enable you to write an API that allows users to go for a coffee! It will comprise a simple, yet extendable user-matching system.

Chapter 6, Hacker News API on Koa.js, will take you through building a CRUD backend to post links and upvote on Koa.js. We will also look at centralized error handling and avoid callback hell with thunks.

Appendix, Connect 4 – Game Logic, shows the accompanying game logic that we omitted in Chapter 3, Multiplayer Game API – Connect 4.

What you need for this book

You'll need the following to get started with the examples in this book:

- Nvm: <https://github.com/creationix/nvm>
- MongoDB: <https://www.mongodb.org/downloads>
- RoboMongo: <http://robomongo.org/>
- Mocha: Use the `npm i -g mocha` command to download it

Mac OS is preferred but not a necessity.

Who this book is for

This book is for beginners to Node.js and also for those who are technically advanced. By the end of this book, every developer will have the expertise to build web applications with Express.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows:
"If it is, then we render the `users/profile.jade` template with `req.user` as the data."

A block of code is set as follows:

```
var express = require('express');
var app = express();

app.get('/', function(req, res, next) {
  res.send('Hello, World!');
});

app.listen(3000);
console.log('Express started on port 3000');
```

Any command-line input or output is written as follows:

```
$ npm install --save express
```

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "You can also right click on the page, and select **Inspect Element**."



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail feedback@packtpub.com, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files from your account at <http://www.packtpub.com> for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at questions@packtpub.com, and we will do our best to address the problem.

1

Building a Basic Express Site

Express is a web development framework for Node.js. Node.js is an open source, cross-platform runtime environment for server-side and networking applications. It uses Google Chrome's JavaScript engine, V8, to execute code. Node.js is single-threaded and event-driven. It uses non-blocking I/O to squeeze every ounce of processing power out of the CPU. Express builds on top of Node.js, providing all of the tools necessary to develop robust web applications with node.

In addition, by utilizing Express, one gains access to a host of open source software to help solve common pain points in development. The framework is unopinionated, meaning it does not guide you one way or the other in terms of implementation or interface. Because it is unopinionated, the developer has more control and can use the framework to accomplish nearly any task; however, the power Express offers is easily abused. In this book, you will learn how to use the framework in the right way by exploring the following different styles of an application:

- Setting up Express for a static site
- Local user authentication
- OAuth with passport
- Profile pages
- Testing

Setting up Express for a static site

To get our feet wet, we'll first go over how to respond to basic HTTP requests. In this example, we will handle several GET requests, responding first with plaintext and then with static HTML. However, before we get started, you must install two essential tools: node and npm, which is the node package manager.

[ Navigate to <https://nodejs.org/download/> to install node and npm.]

Saying Hello, World in Express

For those unfamiliar with Express, we will start with a basic example—Hello World! We'll start with an empty directory. As with any Node.js project, we will run the following code to generate our package.json file, which keeps track of metadata about the project, such as dependencies, scripts, licenses, and even where the code is hosted:

```
$ npm init
```

The package.json file keeps track of all of our dependencies so that we don't have versioning issues, don't have to include dependencies with our code, and can deploy fearlessly. You will be prompted with a few questions. Choose the defaults for all except the entry point, which you should set to `server.js`.

There are many generators out there that can help you generate new Express applications, but we'll create the skeleton this time around. Let's install Express. To install a module, we use `npm` to install the package. We use the `--save` flag to tell `npm` to add the dependency to our package.json file; that way, we don't need to commit our dependencies to the source control. We can just install them based on the contents of the package.json file (`npm` makes this easy):

```
$ npm install --save express
```

We'll be using Express v4.4.0 throughout this book.

[ Warning: Express v4.x is not backwards compatible with the versions before it.]

You can create a new file `server.js` as follows:

```
var express = require('express');
var app = express();

app.get('/', function(req, res, next) {
  res.send('Hello, World!');
}
```

```
});  
  
app.listen(3000);  
console.log('Express started on port 3000');
```

This file is the entry point for our application. It is here that we generate an application, register routes, and finally listen for incoming requests on port 3000. The `require('express')` method returns a generator of applications.

We can continually create as many applications as we want; in this case, we only created one, which we assigned to the variable `app`. Next, we register a GET route that listens for GET requests on the server root, and when requested, sends the string '`Hello, World`' to the client. Express has methods for all of the HTTP verbs, so we could have also done `app.post`, `app.put`, `app.delete`, or even `app.all`, which responds to all HTTP verbs. Finally, we start the app listening on port 3000, then log to standard out.

It's finally time to start our server and make sure everything works as expected.

```
$ node server.js
```

We can validate that everything is working by navigating to `http://localhost:3000` in our browser or `curl -v localhost:3000` in your terminal.

Jade templating

We are now going to extract the HTML we send to the client into a separate template. After all, it would be quite difficult to render full HTML pages simply by using `res.send`. To accomplish this, we will use a templating language frequently in conjunction with Express -- `jade`. There are many templating languages that you can use with Express. We chose Jade because it greatly simplifies writing HTML and was created by the same developer of the Express framework.

```
$ npm install --save jade
```

After installing Jade, we're going to have to add the following code to `server.js`:

```
app.set('view engine', 'jade');  
app.set('views', __dirname + '/views');  
  
app.get('/', function(req, res, next) {  
  res.render('index');  
});
```

sample content of Express.js Blueprints

- [download Hegel: A Very Short Introduction \(Very Short Introductions\)](#)
 - [download Nursing Diagnoses 2015-17: Definitions and Classification \(10th Edition\) online](#)
 - [GR20: Corsica: The High-level route \(A Cicerone Guide\) for free](#)
 - [100 Birds and How They Got Their Names book](#)
 - [click Advanced Bread and Pastry: A Professional Approach here](#)
-
- <http://hasanetmekci.com/ebooks/5-Habits-of-a-Woman-Who-Doesn-t-Quit.pdf>
 - <http://www.khoi.dk/?books/Nursing-Diagnoses-2015-17--Definitions-and-Classification--10th-Edition-.pdf>
 - <http://toko-gumilar.com/books/Fantasy-Gone-Wrong.pdf>
 - <http://aneventshop.com/ebooks/Love-s-Abiding-Joy--Love-Comes-Softly--Book-4-.pdf>
 - <http://aneventshop.com/ebooks/Display-of-Power--How-FUBU-Changed-a-World-of-Fashion--Branding-and-Lifestyle.pdf>