Functional Design Patterns *for* Express.js

```
POST /books HTTP/1.1
Content-Type: application/json
Content-Length: 292

{
  "author": "Jonathan Lee Martin",
  "category": "learn-by-building",
  "language": "JavaScript"
}
```

A step-by-step guide to building elegant, *maintainable* node backends.

Functional Design Patterns for Express.js

A step-by-step guide to building elegant, maintainable Node.js backends.

By Jonathan Lee Martin

Jonathan Lee Martin
hello@jonathanleemartin.com
https://jonathanleemartin.com

Introduction

Learn the design patterns that transcend Express.js and recur throughout high-quality production codebases.

You've built backends in another language for a decade. You're a seasoned frontend JavaScript developer. You're a recent web bootcamp graduate. You're searching for an Express.js primer that isn't another screencast or exhaustive reference guide.

If any of those personas describe you, and you want to:

- **Learn the intuitions of developing elegant, maintainable backends.**
- **Learn without the distractions of every tangential tool in the ecosystem.**
- **Solidly grasp the motivation behind each concept as you build step-by-step.**
- **Expand your design palate with patterns that will transfer to other platforms.**

This book is for you. The pedagogical approach of this book is aimed at transferring design *intuitions* — motivated by real-world consulting experiences — in the fastest way possible. That translates to a razor-focused topic scope and no contrived examples to motivate tools you probably won't use, or shouldn't be using because they indicate deeper "code smells."

If you're looking for an exhaustive Express reference guide, prefer to read passively, or value books and video courses by their length, this book isn't for you — unless you're looking for a handsome adornment for your bookshelf!

Why Express?

Express is arguably the ubiquitous library for building Node backends. It is partly responsible for Node's surge in popularity, and many other Node frameworks build on top of Express. As of mid-2019, it is a dependency of 3.75 *million* codebases on Github alone. So if you hop into a Node codebase, chances are Express is part of it.

Express 5 is in development, but because a sizable group of tech giants depend on the API — directly or through a dependency — Express has essentially been on feature freeze for some time and is unlikely to see substantial overhauls.

This book steers away from version peculiarities and clever utility methods in favor of good design patterns. Thanks to these patterns, the backend we will build together has been rewritten in two other Node.js backend libraries with minimal changes.

Good design in an Express.js backend is good design anywhere. Some design patterns may be more idiomatic in one language than another, but the patterns you learn to develop Node backends will outlive Express and influence your design approaches in unrelated platforms.

Approach

There are countless books out there on backend design, so what makes this one different? In a word, the *approach.*

Many well-meaning books and courses are built on a more-is-better ethos: a single step-by-step course about Express is crammed with tangential topics like ES2015 JavaScript, databases and React. When the teaching approach and learning outcomes become secondary to the topic list, the result is a grab bag of goodies that *entertains* the developer rather than *educates.*

As a globetrotting educator, author and international speaker with a passion for craft, I've guided hundreds of developers — from career switchers to senior developers at Fortune 100 companies — through their journey into web development.

Both in the workplace and in the classroom, I've watched the entertainment model of learning cripple developers. So over the last six years of teaching one to sixteen week bootcamps, I've developed a pedagogical approach for developers at all skill levels.

Pedagogy — the method and practice of teaching — asks the essential question, *what does it mean to teach well?* My approach to vocational teaching is based on a few axioms:

- Teach and apply one concept at a time to minimize cognitive load.
- Focus on contextual learning.
- Leverage the ability to generalize concepts and apply in new contexts.
- Emphasize transmutable concepts.
- Dispel magic by building magic from scratch.
- Encourage fearless curiosity that dispels magic.
- Facilitate self-discovery, then follow with reinforcement.
- Engender love for the abstract from the concrete — not the reverse.
- Transfer intuition — not concepts — as quickly as possible.
- Quality is inversely proportional to length. Conciseness is kindness in practice.

Like a well-designed app, good pedagogy becomes a transparent part of the learning process by removing obstacles to learning — including itself!

Topics

This course focuses on best practice, conventional backend design for *pure* backend APIs. It is not exhaustive, comprehensive or targeted at advanced Express developers who are trying to scale huge legacy backends.

As we build a full-featured backend together, expect to work through:

- HTTP from scratch
- Request-response (life)cycle
- Express.js features that appear in high-quality codebases
- Testing backend routes with Insomnia
- Conventional headers for pure APIs
- Router design pattern
- Decoupling backend code
- Functional-style design patterns
- Currying and partially applied functions
- Dynamic segments
- Working with bodies
- Function objects
- Middleware
- Global vs. route middleware
- Middleware factories
- Common middleware libraries
- Authentication vs. authorization
- Password authentication
- Authentication with JSON Web Tokens
- Authorization design patterns

Because of this book's razor-focused approach, it intentionally omits:

- ES2015–ES2017 JavaScript
- RESTful conventions
- Databases
- Node essentials
- Frontend
- Cookies and sessions
- Passport.js
- Templating
- Niche Express methods, especially if they are symptomatic of design flaws.

Instead, it is this book's intention to equip developers — who already have a thorough applied knowledge of JavaScript, some light Node experience, and who have preferably built a backend before in any language or framework — with design insights.

Prerequisites

It is recommended that you have a strong foundation in JavaScript, preferably through hands-on product development. If your JavaScript experience is academic or limited to occasional hacking, the learning outcomes of this book may not be valuable.

Specifically, it is strongly recommended that:

- You have solid hands-on experience in JavaScript and Node.js.

- You are immensely comfortable with async programming in JavaScript with callbacks, async functions and Promises.
- You have ES2015 (previously called ES6) under your belt, especially destructuring syntax and arrow functions.
- You have an experiential understanding of HTTP, though a rigorous understanding is unnecessary.

Some things are *not* required to get the most out of this book! You don't need prior backend experience. If you understand how servers and clients interact, experience from either side of the equation is sufficient.

Let's Get Started

Throughout this book, we'll be building a full-featured Express backend together called **Pony Express**. Starting from an empty directory, we will intentionally bump into codebase growing pains to motivate functional design patterns and Express features.

But first, in the next chapter we'll detour from Node altogether and demystify the core abstraction of the web: **HTTP**.