

Extracted from:

Node.js the Right Way

Practical, Server-Side JavaScript That Scales

This PDF file contains pages extracted from *Node.js the Right Way*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2013 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

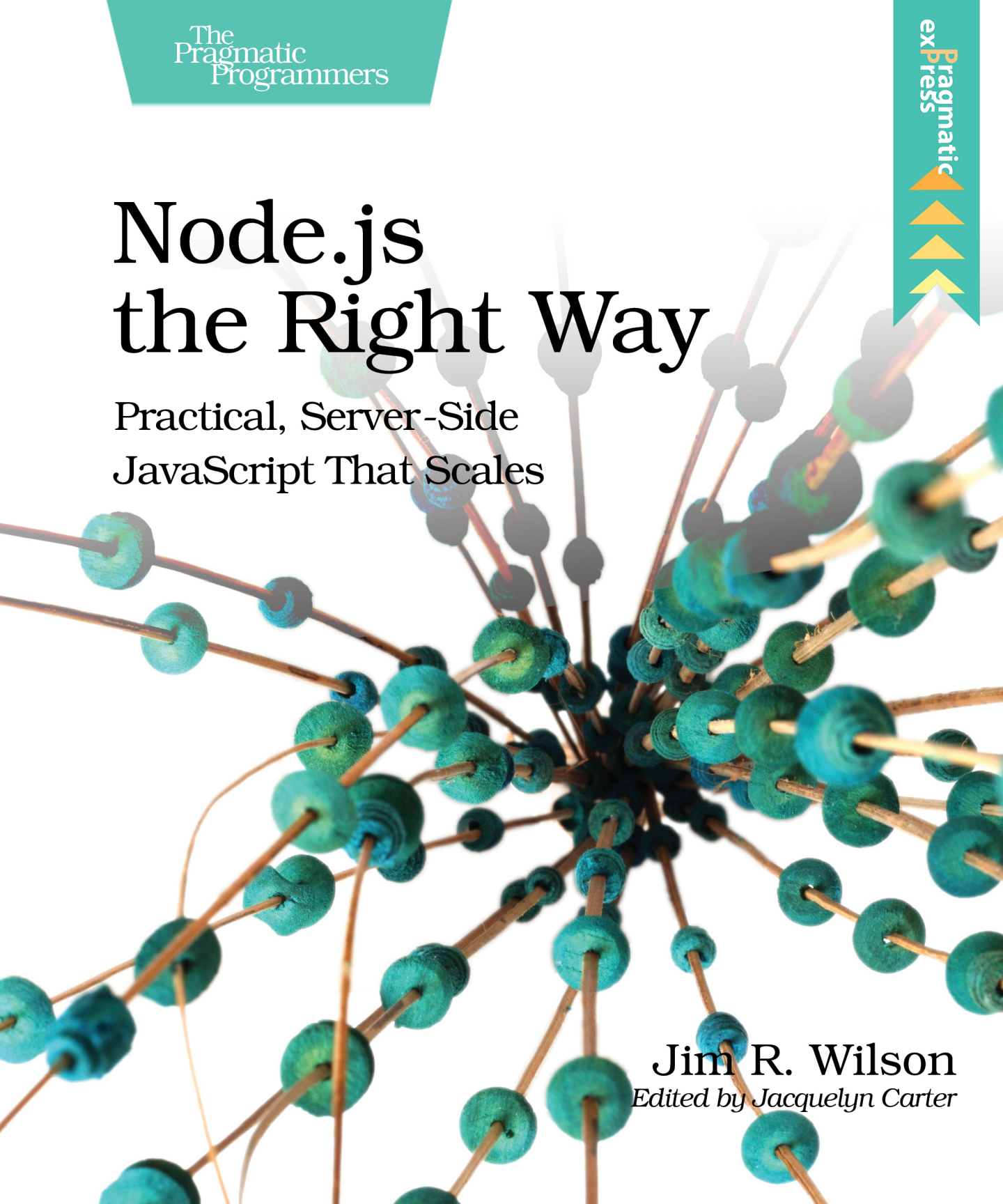
The
Pragmatic
Programmers

Pragmatic
express

Node.js the Right Way

Practical, Server-Side
JavaScript That Scales

Jim R. Wilson
Edited by Jacquelyn Carter



Node.js the Right Way

Practical, Server-Side JavaScript That Scales

Jim R. Wilson

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



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 <http://pragprog.com>.

The team that produced this book includes:

Jacquelyn Carter (editor)
Candace Cunningham (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2013 The Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-937785-73-4
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—December 2013

Preface

Two big shifts are happening right now in the practice of writing software, and Node.js is at the forefront of both.

First, software is becoming increasingly asynchronous. Whether you're waiting for a Big Data job, interacting with end users, or simply responding to an API call, chances are you'll need asynchronous programming techniques.

Second, JavaScript has quietly become the world's standard virtual machine—in web browsers, modern NoSQL databases, and now on the server as well.

Node.js is right at the intersection of these trends, and it's ready to take off in a big way.

Why Node.js the Right Way

In March of 2010, I gave a lightning talk titled “Full-Stack JavaScript” at the NoSQL Boston conference. Back then, and even more so now, I knew that using JavaScript for every layer of the application stack was not only possible, but a great way to reduce software complexity.

The *Right Way* in this book's title refers to both the process of learning Node and the practice of writing Node.

Learning Node.js

As for any growing technology, there are lots of resources available for learning Node.js. Many are intently focused on serving up web resources. The web is great, but it's not enough, and it's not the whole story of Node.

Ruby is more than Rails, and Python is more than Django. Node.js is more than serving web content, and this book treats it that way.

Node.js the Right Way teaches you the core concepts you'll need to be an effective Node.js programmer, no matter what kinds of programs you need to write.

Writing Node.js

One thing I love about JavaScript is that there are seven ways to do anything. There's breathing room, where developers can explore and experiment and find better approaches to everything.

The community of Node developers, conventions in Node.js development, and even the semantics of the JavaScript language itself are all rapidly evolving. With eyes to the near future, the code examples and recommendations in this book reflect current best practices and standards.

What's in This Book

This book is for developers who want to learn how to write asynchronous JavaScript for the server using Node.js. Some prior JavaScript experience will help, but you don't have to be an expert.

[Chapter 1, *Getting Started*, on page ?](#), introduces the Node.js event loop, explaining how it empowers Node to be highly parallel and single-threaded at the same time. This chapter also outlines the five aspects of Node.js development that frame each subsequent chapter and has some brief instructions on getting Node installed on your machine.

The remaining chapters each deal with a specific practical programming area.

Wrangling the File System

In [Chapter 2, *Wrangling the File System*, on page ?](#), we'll get our first look at writing Node.js programs. If you've done any server-side programming in the past, chances are you've had to access a file system along the way. We'll start in this familiar domain, using Node's file-system tools to create asynchronous, nonblocking file utilities. You'll use Node's ubiquitous EventEmitter and Stream classes to pipe data, and you'll spawn and interact with child processes.

Networking with Sockets

We'll expand on those concepts while exploring Node's network I/O capabilities in [Chapter 3, *Networking with Sockets*, on page ?](#). We'll create TCP servers and client programs to access them. We'll also develop a simple JSON-based protocol and a custom module for working with these messages. This will offer insight into Node application design and provide experience creating testable and fault-tolerant systems.

Robust Message-Passing

Then, in [Chapter 4, *Robust Messaging Services*, on page ?](#), we'll branch away from the Node core and into the realm of third-party libraries. You'll use npm to import and build ØMQ (pronounced "Zero-M-Q")—a high-efficiency, low-latency library for developing networked applications. With ØMQ, we'll develop programs that communicate using several important patterns, such as publish/subscribe and request/response. We'll create suites of programs that work together in concert, and you'll learn the clustering tools to manage them.

Accessing Databases

[Chapter 5, *Accessing Databases*, on page ?](#), introduces databases and how to interact with them asynchronously in Node. In particular, we'll work with CouchDB—a RESTful, JSON document database. You'll learn how to parse XML documents, throttle Node using a worker queue, and develop and run unit tests. The database we create in this chapter is the foundation for RESTful APIs you'll develop in later chapters.

Scalable Web Services

Node has fantastic support for writing HTTP servers, and in [Chapter 6, *Scalable Web Services*, on page ?](#), we'll do exactly that. You'll use Express, a popular Node.js web framework for routing requests. We'll dive deeper into REST semantics, and you'll use objects called promises for managing asynchronous code flows. You'll also learn about a bleeding-edge feature of ECMAScript called generator functions, and how they couple with promises in interesting ways.

Web Apps

Finally, in [Chapter 7, *Web Apps*, on page ?](#), we'll build a front end for our web services. We'll use a Node module called Passport for implementing authenticated APIs that use Google account credentials. And we'll serialize our session data in Redis—a very fast key/value datastore. You'll learn the basics of writing a static single-page web application that uses RESTful APIs, including how to pull in dependencies using a front-end package manager called Bower.

What This Book Is Not

Before you commit to reading this book, you should know what it doesn't cover.

Everything About Everything

At the time of this writing, npm houses more than 43,000 modules, with an average growth rate of 100-plus new modules per day.¹ Since the ecosystem and community around Node.js is growing and changing so rapidly, this book does not attempt to cover everything. Instead, this short book teaches you the essentials you need to get out there and start coding.

The book also stays close to the topic of Node.js. You'll learn a lot about clustering Node processes and how to write scalable web services, but little about front-end concerns like HTML, CSS, and browser JavaScript.

JavaScript Beginner's Guide

The JavaScript language is one of the most misunderstood languages in wide use today. Although this book does discuss language syntax from time to time (especially where it's brand-new), this is not a beginner's guide to JavaScript.

A Note to Windows Users

The examples in this book assume you're using a Unix-like operating system. We'll make use of standard input and output streams, and pipe data between processes. The shell session examples have been tested with Bash, but other shells may work as well.

If you run Windows, I recommend setting up Cygwin.² This will give you the best shot at running the example code successfully, or you could run a Linux virtual machine.

Code Examples and Conventions

The code examples in this book contain JavaScript, shell sessions, and a few HTML/XML excerpts. For the most part, code listings are provided in full—ready to be run at your leisure.

Samples and snippets are syntax-highlighted according to the rules of the language. Shell commands are prefixed by \$.

When you write Node.js code, you should always handle errors and exceptions, even if you just rethrow them. You'll learn how to do this throughout the book. However, some of the code examples lack error handling. This is to aid readability and save space only—you should always handle your errors.

1. <http://www.modulecounts.com/>

2. <http://cygwin.com/>

Online Resources

The Pragmatic Bookshelf's page for this book is a great resource.³ You'll find downloads for all the source code presented in this book, and feedback tools, including a community forum and an errata-submission form.

Thanks for choosing this book to show you Node.js *the right way*.

Jim R. Wilson (jimbojw, hexlib)

November 2013

3. <http://pragprog.com/book/jwnode/node-js-the-right-way>