

Extracted from:

# Design and Build Great Web APIs

Robust, Reliable, and Resilient

This PDF file contains pages extracted from *Design and Build Great Web APIs*, 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 © 2020 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

Raleigh, North Carolina

The  
Pragmatic  
Programmers

# Design and Build Great Web APIs

Robust, Reliable, and Resilient



Mike Amundsen  
*edited by Katharine Dvorak*



# Design and Build Great Web APIs

Robust, Reliable, and Resilient

Mike Amundsen

The Pragmatic Bookshelf

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 books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt

VP of Operations: Janet Furlow

Executive Editor: Dave Rankin

Copy Editor: Molly McBeath

Indexing: Potomac Indexing, LLC

Layout: Gilson Graphics

For sales, volume licensing, and support, please contact [support@pragprog.com](mailto:support@pragprog.com).

For international rights, please contact [rights@pragprog.com](mailto:rights@pragprog.com).

Copyright © 2020 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.

ISBN-13: 978-1-68050-680-8

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—October 2020

# Preface

---

Welcome to the world of API development! This book is designed to help you go from start to finish on the path to designing, building, and deploying great application programming interfaces (APIs) for the web. Along the way you'll learn a handful of important practices and principles that will help you design and build APIs that are robust, reliable, and resilient. You'll also acquire skills in a wide range of developer tools, including tools for design, documentation, building, testing, security, and deployment. By the time you work through this book, you should have enough experience to tackle just about any challenge that comes your way.

## Your API Journey

The content in this book follows a learning path I've developed over the years to help developers from varying backgrounds get a foothold and gain experience in the API space.

## Who Should Read This Book

This book is aimed at software developers and designers looking to quickly get up to speed on how to design and build web APIs. Since the content is based on training material I've used over the years, this book could be handy should you put together your own API training course for your team or company.

It can also be used as a general reference guide on basic topics like API design methods, the API build process, testing, and other steps in API development. For this reason, the book is also valuable to senior software architects and API product managers looking to provide common guidance to all their teams.

## How This Book Is Organized

The book is organized into five parts: Getting Started, The Design Phase, The Build Phase, The Release Phase, and Appendixes. Each part marks a key set of skills in the journey from thinking about APIs to actually getting them up and running in production. The following is a quick summary of each of the five parts of the book.

### Part I: Getting Started

The first part helps set the foundation for the world of APIs on the web. We first explore what it means to adopt an “API-First” approach to designing and building APIs. We then get some background on the existing services we’ll be working with and our plan for implementing an API requested by a fictitious customer, BigCo, Inc.

In the second chapter, we spend time exploring the basics of the protocols and practices commonly used to create APIs on the web, including the HTTP protocol and the REST style for creating services that run on HTTP.

### Part II: The Design Phase

In Part II, we focus on the process of modeling, designing, and describing our API. Along the way we look at tools to help us model and visualize the API and ones that help us create a machine-readable description that can be used to produce our final API.

### Part III: The Build Phase

The build phase of API development is all about converting our well-designed and well-described API into working code. In this part we take a three-step approach by sketching, prototyping, and finally building the API itself. We use the Apiary Blueprint format for sketches, the Open API Specification (aka Swagger) to create the exact definition, and then NodeJS and a custom framework called DARRT to turn the Swagger spec into working code.

### Part IV: The Release Phase

In the fourth part of the book, we look at how to test, secure, and deploy our completed API into the cloud using the local Postman platform for testing and the Auth0 cloud service for authentication and access control, and then we release the finished API onto the Heroku public cloud.

Once we know that's all working as expected, we look at how to introduce minor changes to our production API without breaking any existing API consumers.

## Part V: Appendixes

At the end of the book is a small collection of appendixes that you can use as a reference while you're reading through the book. The first appendix covers what programs you may want to install as you follow along in the chapters. The next appendix offers solutions to the chapter exercises. The final appendix lists all the API project assets you should be creating and managing as you go through all the stages of API design, build, and release.

## In Reality

While the book takes a very linear approach to all this (driving from modeling to building to releasing the API), in reality API development involves lots of iteration and bouncing back and forth between skills along the way. In most production work, you end up working in a series of “loops” that allow you to model, then design, then go back and fix the model a bit, and then design some more, and so on until you finally get to the point where your API is ready to be released to the cloud.

While this looping is the right way to do the work, it isn't a good way to write a book about it. For that reason, we'll stick to our straight-line approach from start to finish, with lots of reminders along the way about iteration and looping as a key element in designing and building great APIs.

## What's Covered (And What's Not)

The goal of this book is to give you comprehensive “front-to-back” coverage of what's needed to get a new API up and running in production. To keep things focused, we'll be covering the skills needed to create “RESTful” APIs. These are HTTP-based APIs that rely on the common request-response style used by the vast majority of APIs on the web today. There are other important API styles (such as gRPC and GraphQL) that we won't have time to cover in a single book. The good news is that most other API styles rely on the HTTP-based web API approach we'll cover here. Getting a solid understanding of this RESTful style will come in handy if you want to explore other styles in the future.



The topic of APIs for the web is quite large, and we won't be able to cover it all. This book focuses on a series of steps centered around the external interfaces (the APIs) used to access back-end services. However, I won't be getting into how to build and deploy back-end services (sometimes called microservices) themselves. If you're interested in that topic, I recommend you check out Chris Richardson's *Microservices Patterns*<sup>1</sup> or Sam Newman's *Building Microservices*.<sup>2</sup>

I also won't be covering the details of writing the client applications that *consume* web APIs. If you're looking for a source on that, I suggest you check out another of my books, *RESTful Web Clients*.<sup>3</sup>

Finally, I won't be covering topics related to the broader topic of API management. Things like API gateways, portals, monitoring, and overall governance are covered well in books like *Continuous API Management*,<sup>4</sup> which I co-wrote with Mehdi Medjaoui, Erik Wilde, and Ronnie Mitra.

## About the BigCo, Inc., Sample Project

In this book, we'll be helping a fictional company (BigCo, Inc.) modernize its process of bringing on new customers by designing, building, and deploying a brand-new API called the "Onboarding API." The company's onboarding process is still handled mostly by humans filling in forms and then adding data to several stand-alone applications running in the company's network. Our job will be to learn how BigCo currently handles this customer onboarding process, convert that into an API design, and then—using existing services already in place—implement a single onboarding API that handles all the steps currently handled by BigCo staff.

Along the way, we'll look at the basics of API design and how to produce and document our design in a consistent way that will be understood by both BigCo onboarding staff and developers. We'll also look at how to use a simple three-stage process (sketch, prototype, and build) for converting an API design into running code. And once the code is available, we'll work through the release process of testing, securing, and finally deploying the working onboarding API. Finally, we'll look at what it takes to make updates to an already-released API without breaking any of the apps currently using it.

1. <https://microservices.io/book>
2. <https://learning.oreilly.com/library/view/building-microservices/9781491950340>
3. <https://learning.oreilly.com/library/view/restful-web-clients/9781491921890>
4. <https://learning.oreilly.com/library/view/continuous-api-management/9781492043546>

## About BigCo, Inc.

As a bit of background, here's the "origin story" for the company behind all the examples in this book.

The Bayesian International Group of Companies (BigCo) is an organization with a long history of both prosperity and generosity. Founded in 1827 in Scotland, BigCo's motto is "The principle end of the company is the happiness of its customers and employees." Originally focused on demography and finance and based on the work of Richard Price, by the 1900s BigCo was a manufacturer of important turn-of-the-century technologies, including mobile x-ray machines and hydrophones, by working with inventors Frederick Jones and Reginald Fessenden, respectively. BigCo was also one of the few non-U.S. contractors attached to the Manhattan project, providing parts for the first nuclear reactors designed by Dr. Leona Woods.

During the postwar boom, BigCo opened offices in the United States and played an instrumental role in helping to establish fields of study in the modern computer era, including predictive algorithms and robotics controls. Currently, BigCo products focus on "inverse probability" problems and how they can be used to help improve decisions and policy-making in both the private and public sectors.

While BigCo is an imaginary company, all the people mentioned in its description (Price, Jones, Fessenden, and Woods) are real historical figures. If BigCo *did* exist, it's possible that these innovators would have been part of its history, too.

## Online Resources

To follow along with the project throughout the book, download the source code from the book's web page at [pragprog.com](https://pragprog.com).<sup>5</sup> And if you find any errata, there's also a place to let us know about it.<sup>6</sup>

But before we dive into designing and programming the API, let's spend a little "getting started" time on the concepts of API First and refreshing our knowledge of the HTTP protocol, web concepts, and the API style known as REST (REpresentational State Transfer).

Let's get started!

5. <https://pragprog.com/book/maapis>

6. <https://devtalk.com/books/design-and-build-great-web-apis>