

The  
Pragmatic  
Programmers

# Programming WebRTC

Build Real-Time Streaming  
Applications for the Web



**Karl Stolley**  
*edited by Michael Swaine*

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit <https://www.pragprog.com>.

Copyright © The Pragmatic Programmers, LLC.

# Preface

---

WebRTC—or Web Real-Time Communication—is a standardized API exposed in all modern web browsers. The World Wide Web Consortium accepted the WebRTC specification as a full recommendation after a long decade of development.<sup>1</sup> Couple a complete, stable specification with browser support for WebRTC ranging from rock-solid to serviceable, and you find yourself in a perfect environment for developing and deploying real-time web applications in the browser.

Like any Web API, WebRTC doesn't enjoy a perfectly spec-aligned implementation in any browser. But this book will start you on your journey to developing real-time streaming applications, all according to the certainty of a stable specification. You'll also learn to write elegant, backward-compatible code to get your WebRTC apps working across the widest possible range of recent and modern browsers. Desktop and mobile devices, too. Support for WebRTC is everywhere.

## Your WebRTC Journey

You'll start your journey by getting straight to work on building a basic WebRTC application to support peer-to-peer video calling. Chapter by chapter, you'll refine that app and its core logic to then spin up additional WebRTC-powered apps that will have your users sharing all manner of data with one another, all in real time.

This book treats WebRTC as a part of the Web Platform. No third-party libraries or heavy downloads are required for you to make your way through this book, or for your users to use the WebRTC apps you build: you'll be writing and strengthening your knowledge of modern HTML, CSS, and JavaScript to get the most out of browser-native WebRTC APIs.

---

1. <https://www.w3.org/TR/webrtc/>

From the outside, WebRTC is pretty daunting. Okay, really daunting. I'm proud of you just for peeking inside this book. But don't put it down just because WebRTC will challenge you and twist your brain around in some profound ways. That's both expected and totally okay. It means you're on the path. We'll walk it together. Things look better and more manageable from the inside.

This book will get you on the path to where you want to go with WebRTC right away. You won't find any throwaway code or opaque, puzzling examples here. We're going to be developing real WebRTC applications together from the outset. And it will be from those real, functioning applications that we will tease out how WebRTC works. By the end of your journey, you will have all the foundational skills and knowledge you need to build your own wildly imaginative real-time applications. And because work on the WebRTC specification continues, you will also learn how to stay on top of the latest changes and discussions.

## Who Should Read This Book?

This book is aimed at intermediate and advanced web designers and developers looking to explore and implement real-time communication features in new or existing web applications.

Whether you consider yourself a designer or a developer, WebRTC is one of those rare Web APIs where design and development converge head on: not just conceptually, but in the actual code you'll be writing. WebRTC is a front-end technology that requires only the teeniest, tiniest server-side component, which I have provided for you in the codebase that accompanies this book. Almost all of your work will be run and rendered directly in the browser.

You should have at least some knowledge of JavaScript, along with HTML and CSS. I'll do my best to fill in any gaps that might arise for you and point out additional books and resources that you might find useful. You'll find yourself working with other Web APIs in this book—not just WebRTC. Many of those will equip you with added knowledge to enhance your work on all kinds of web applications, whether or not they include a real-time component.

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

This book covers WebRTC's APIs as natively implemented in recent and modern web browsers. You'll be working with those APIs directly in vanilla JavaScript to build your knowledge and command of WebRTC independent of any third-party libraries. The promise of WebRTC has always been to provide

real-time communication right in the browser without requiring users to download special plugins or add-ons. Your users' browsers already have everything needed to power your real-time app. All you have to do is build it!

But to build a stunningly accessible and usable real-time application requires more than just a strong JavaScript foundation. It is an all-too-common mistake to think of WebRTC as just another conduit for media streams and application data. Because of its real-time component, WebRTC knits together two or more remote interfaces where real, live users are interacting and cooperating with one another. Building a WebRTC app without thinking carefully about the user interface would be like building a bicycle without the seat and handlebars. Ouch, right? So don't be surprised to spend some time—perhaps more than you'd expect—working with HTML and CSS, too.

You will be doing WebRTC development within the friendly confines of your local network for most of the book, but a chapter at the end will walk you through the necessary requirements and steps for deploying your real-time applications to the web, and testing them out.

So what's not covered? In a phrase, this book does not cover issues of scale or millisecond-obsessed WebRTC optimization. While there are a growing number of server-, platform-, and system-based implementations of WebRTC, as well as numerous WebRTC-based communication platforms as a service (CPaaS), those are all beyond the scope of this book. That means you won't find coverage here of scaling apps up to handle dozens or thousands of concurrent users supported by server-side technologies like selective forwarding units (SFUs) or multi-conference units (MCUs). Although you will work with a small server that provides a signaling channel, that is the extent of the server-side content in this book.

And while you'll learn about some fundamentals of streaming-media CODECs and optimization, this book does not go deep on those topics. Nor does it encourage session description protocol (SDP) “munging” to coerce browsers to use a particular CODEC, or fiddling with `RTCRtpTransceiver` objects. Native browser code for those matters is better tuned and tested than app-based adjustments likely ever will be.

However, the core principles of WebRTC that we'll look at in depth—working with a signaling channel, establishing peer connections, adding and managing media streams and data channels—will have you well prepared to tackle WebRTC implementations and third-party services wherever you might encounter them.

## How This Book Is Organized

This book is organized in a set of sequential chapters. It's meant to be read more or less front to back. If that feels overly prescriptive and stifling, or if you've just got a rebellious streak that you like to take out on tech books, try anyway to make it through at least the first four chapters before you jump around to the later ones whose topics most interest you.

In Chapter 1, you'll learn how to set up a development environment that will play nicely with WebRTC. You'll also learn how to get the most out of the starter and example code that accompanies the book.

With your development environment set up and tested out, Chapter 2 dives right into the only necessary server component of WebRTC apps: a signaling channel. As part of working with the signaling channel, you'll start to build the basic interface of your first WebRTC app, which provides peer-to-peer video calls.

Chapter 3 is where you'll really hit your stride working directly with WebRTC's APIs, all of which orbit around the `RTCPeerConnection` interface. By the end of this chapter, you'll be streaming video between two connected peers which—to start—will be two browser windows on your desktop.

Streaming real-time user video and audio is WebRTC's most famous feature. But that's not its only feature. Over the course of Chapter 4 and Chapter 5, you'll go from streaming basic data to streaming more complex data—including JSON as well as images and other binary files. WebRTC provides a powerful and flexible low-level interface for streaming arbitrary application data between two peers, all in real time. You'll learn to command that interface, and abstract away subtle differences found in browsers with incomplete WebRTC implementations. You'll even bring what you learn full circle by safely implementing user audio to complete the silent streaming video you'll work with at first.

Buckle your seatbelt when you get to Chapter 6. Connecting two peers is one thing. But how about connecting three or more peers? Chapter 6 will have you establishing WebRTC calls using a mesh-network topography to enable multiple peers to join the same call simultaneously. You'll also experience the theoretical and practical upper limits on the number of peers who can join a call on a mesh network, depending on what your app does and the amount of bandwidth and processing power it consumes.

In Chapter 7, you'll work more in depth with the `MediaDevices` interface on the Media Capture and Streams API to do things like help users determine what

mics and cameras they have available, and handle edge cases in your logic when either there are no devices available, or users deny permission to access them. You'll learn to do some minor media-stream optimization, too, with the aid of the built-in, real-time statistics that WebRTC implementations provide in the browser.

Closing out the body of the book, in Chapter 8, you'll learn how to deploy WebRTC applications to production. You'll find a concrete deployment example, but you'll also learn how to adjust that example to suit your own needs and preferences.

And finally, you will find an appendix at the end of the book that will show you the necessary fixes for making your WebRTC applications work with legacy browsers that don't support the WebRTC APIs necessary for perfect negotiation.

## Online Resources

You can download the source code for studying and working alongside the examples in the book from [pragprog.com](https://pragprog.com).<sup>2</sup> If you spot an error or even just come across something that is blocking your path forward in the book, consider this my personal invitation to you to join and post to the book's forum on DevTalk.<sup>3</sup>

If you would like to contact me directly, I am available on Mastodon at [@stolley@hachyderm.io](https://hachyderm.io/@stolley).<sup>4</sup> I also blog about WebRTC and other web topics at <https://stolley.dev/>

All right. Enough with the formalities that we classy preface readers enjoy while thoughtfully adjusting our monocles from the comfort of a high-back leather chair. Let's pop out the monocle, pull open a laptop, and get down to it: it's time to set up a development environment that will be your trusty companion on your exciting, monocle-free journey with WebRTC.

---

2. <https://pragprog.com/titles/ksrtc/>

3. <https://devtalk.com/books/programming-webrtc/errata>

4. <https://hachyderm.io/@stolley>