



# Building Browser Extensions

Create Modern Extensions for  
Chrome, Safari, Firefox, and Edge

---

Matt Frisbie

*Foreword by Stefan Aleksic and Louis Vilgo,  
cofounders of Plasmo*



Apress®

# **Building Browser Extensions**

**Create Modern Extensions  
for Chrome, Safari, Firefox,  
and Edge**

**Matt Frisbie**

*Foreword by Stefan Aleksic and Louis Vilgo,  
cofounders of Plasmo*

**Apress®**

## ***Building Browser Extensions: Create Modern Extensions for Chrome, Safari, Firefox, and Edge***

Matt Frisbie  
California, CA, USA

ISBN-13 (pbk): 978-1-4842-8724-8  
<https://doi.org/10.1007/978-1-4842-8725-5>

ISBN-13 (electronic): 978-1-4842-8725-5

Copyright © 2023 by Matt Frisbie

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Divya Modi

Development Editor: James Markham

Coordinating Editor: Divya Modi

Cover designed by eStudioCalamar

Cover image designed by Pixabay

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail [orders-ny@springer-sbm.com](mailto:orders-ny@springer-sbm.com), or visit [www.springeronline.com](http://www.springeronline.com). Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail [booktranslations@springernature.com](mailto:booktranslations@springernature.com); for reprint, paperback, or audio rights, please e-mail [bookpermissions@springernature.com](mailto:bookpermissions@springernature.com).

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at <https://github.com/Apress/Building-Browser-Extensions-by-Matt-Frisbie>. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my parents Mona and Patrick and my sister Lauren.  
You are all wonderful. I will continue to fill your  
bookshelves until you tell me to stop.*

*To Jordan.*

*Your unwavering support advances the world  
of web development and keeps the corners of our  
carpets nice and flat.*

# Table of Contents

<b>About the Author .....</b>	<b>xxiii</b>
<b>About the Technical Reviewer .....</b>	<b>xxv</b>
<b>Acknowledgments .....</b>	<b>xxvii</b>
<b>Foreword .....</b>	<b>xxix</b>
<b>Introduction .....</b>	<b>xxxI</b>
<b>Chapter 1: What Are Browser Extensions? .....</b>	<b>1</b>
History of Browser Extensions .....	2
Customizing Software with Plugins.....	2
History of Web Browsers .....	3
Native Browser Plugins .....	4
From Browser Add-ons to Extensions .....	4
The Browser Extension Landscape .....	5
Comparing Mobile Apps and Browser Extensions .....	5
Browser Extension Stores .....	6
Types of Browser Extensions.....	7
Summary.....	14
Credits.....	15
<b>Chapter 2: Fundamental Elements of Browser Extensions.....</b>	<b>17</b>
The Browser Model .....	17
Browser Tabs .....	19
Same-Origin Policy .....	20

## TABLE OF CONTENTS

<b>The Browser Extension Model .....</b>	<b>20</b>
Independent JavaScript Pages and Runtimes .....	21
Native APIs and User Interfaces .....	22
Tab and Domain Access.....	23
Observing and Intercepting Network Requests.....	23
<b>Elements of Browser Extensions .....</b>	<b>24</b>
Extension Manifest .....	24
Background Scripts .....	25
The Popup Page.....	26
The Options Page.....	28
Content Scripts .....	30
Devtools Panels and Sidebars .....	31
<b>Extension Elements in Action.....</b>	<b>33</b>
Honey .....	33
LastPass .....	35
Grammarly .....	38
React Developer Tools .....	40
<b>Summary.....</b>	<b>42</b>
<b>Chapter 3: Browser Extension Crash Course.....</b>	<b>43</b>
Creating the Manifest.....	44
Minimum Viable Extension.....	45
Installing Your Extension .....	45
Reloading Your Extension.....	50
Adding a Background Script .....	52
Adding a Popup Page .....	56
Adding an Options Page .....	60
Adding a Content Script .....	64

## TABLE OF CONTENTS

Working with Multiple Tabs .....	67
Adding a Devtools Panel .....	69
Summary.....	72
<b>Chapter 4: Browser Extension Architecture .....</b>	<b>75</b>
Architecture Overview.....	76
Plurality, Lifecycles, and Updates.....	80
Background Service Worker .....	80
Popup and Options Pages.....	81
Devtools Pages .....	82
Content Scripts .....	82
Browser Extension File Server.....	83
Sandboxed Pages .....	94
Summary.....	96
<b>Chapter 5: The Extension Manifest .....</b>	<b>97</b>
The Manifest File .....	98
Supporting Different Locales .....	99
Match Patterns and Globs.....	101
File Path Match Patterns .....	101
URL Match Patterns.....	102
URL Globs .....	102
Manifest Properties.....	103
action.....	104
author .....	110
automation .....	110
background .....	111
browser_action .....	114
browser_specific_settings .....	115

## TABLE OF CONTENTS

chrome_settings_overrides.....	116
chrome_url_overrides .....	124
commands.....	125
content_capabilities .....	131
content_scripts.....	131
content_security_policy .....	134
converted_from_user_script.....	136
cross_origin_embedder_policy .....	136
cross_origin_opener_policy .....	137
declarative_net_request .....	138
default_locale.....	139
description.....	140
developer.....	140
devtools_page .....	141
differential_fingerprint .....	141
event_rules.....	142
externally_connectable .....	142
file_browser_handlers .....	143
file_system_provider_capabilities .....	144
homepage_url .....	145
host_permissions .....	146
icons .....	146
incognito.....	147
key.....	148
manifest_version.....	149
minimum_chrome_version .....	150
nacl_modules .....	150
name.....	150

## TABLE OF CONTENTS

oauth2 .....	151
offline_enabled.....	152
omnibox.....	152
optional_host_permissions .....	153
optional_permissions.....	153
options_page.....	154
options_ui.....	154
page_action.....	157
permissions.....	157
platforms .....	158
replacement_web_app.....	158
requirements .....	159
sandbox .....	159
short_name .....	160
storage .....	161
system_indicator.....	161
tts_engine .....	162
update_url .....	163
version.....	163
version_name.....	164
web_accessible_resources.....	164
Summary.....	166
<b>Chapter 6: Understanding the Implications of Manifest V3.....</b>	<b>167</b>
Motivation for Manifest V3 .....	168
Security .....	168
Performance.....	168
Privacy and Transparency.....	169
Revenue.....	169

## TABLE OF CONTENTS

<b>Implications of Background Service Workers .....</b>	<b>170</b>
DOM.....	171
XMLHttpRequest.....	171
Timer API .....	171
Event Handlers .....	172
Service Worker Persistence.....	174
Global State and Storage.....	176
Audio and Video.....	179
<b>Implications of Content Security Policy Restrictions.....</b>	<b>179</b>
<b>Implications of DeclarativeNetRequest.....</b>	<b>180</b>
Summary.....	185
<b>Chapter 7: Background Scripts .....</b>	<b>187</b>
Web Page Service Workers vs. Extension Service Workers .....	187
Similarities .....	190
Differences .....	192
Manifest v2 vs. Manifest v3 .....	194
Scripts vs. Service Workers.....	194
JavaScript Imports .....	195
No Access to DOM and Limited Global APIs.....	196
Nonpersistent .....	197
No Shutdown Event .....	198
No Programmatic Background Access .....	199
Working with Background Scripts .....	199
Inspecting Background Service Workers.....	200
Service Worker Errors.....	202
Service Worker Termination.....	205

## TABLE OF CONTENTS

<b>Common Patterns .....</b>	<b>207</b>
Event Handler .....	208
Secret Management and Authentication .....	210
Message Hub.....	211
Storage Manager.....	214
Injecting Scripts.....	215
Sniffing Web Traffic.....	216
Installed/Updated Events.....	217
Opening Tabs .....	218
Forcing Service Worker Persistence .....	220
Implementation .....	220
Drawbacks and Limitations .....	223
Summary.....	224
<b>Chapter 8: Popup and Options Pages.....</b>	<b>225</b>
<b>Popup Pages .....</b>	<b>225</b>
<b>Popup Page Properties .....</b>	<b>226</b>
<b>Opening and Closing Popup Pages.....</b>	<b>229</b>
<b>Changing the Popup Page .....</b>	<b>231</b>
<b>Detecting Popup State.....</b>	<b>231</b>
<b>Suggested Use.....</b>	<b>232</b>
<b>Options Pages .....</b>	<b>233</b>
<b>Options Page Properties .....</b>	<b>234</b>
<b>Opening and Closing Options Pages.....</b>	<b>236</b>
<b>Detecting Options Pages .....</b>	<b>237</b>
<b>Suggested Use.....</b>	<b>238</b>
<b>Content Script Restrictions.....</b>	<b>239</b>
Summary.....	243

## TABLE OF CONTENTS

<b>Chapter 9: Content Scripts .....</b>	<b>245</b>
Introduction to Content Scripts .....	245
Injecting CSS .....	246
Content Script Isolation .....	247
Page Automation .....	250
Logging and Errors .....	254
Extension API Access .....	255
Modules and Code Splitting .....	256
Bundling .....	256
Dynamic Imports .....	257
Dynamic Script Tags .....	260
Specialized Content Script Properties.....	261
Programmatic Injection.....	262
Summary.....	268
<b>Chapter 10: Devtools Pages.....</b>	<b>269</b>
Introduction to Devtools Pages .....	270
Creating a devtools Page.....	271
Adding panels and sidebars .....	272
The Devtools API .....	282
Sniffing network traffic .....	283
Inspecting the Page .....	289
Content Scripts and Devtools Messaging .....	296
Other Devtools API Features.....	298
Summary.....	299

## TABLE OF CONTENTS

<b>Chapter 11: Extension and Browser APIs .....</b>	<b>301</b>
Global API Namespace .....	301
Promises vs. Callbacks .....	302
Error Handling .....	303
Context-restricted APIs .....	303
Events API .....	304
Format .....	304
Event Filtering .....	306
WebExtensions API Quick Reference .....	306
Permissions .....	307
Messaging .....	308
Storage .....	313
Authentication .....	314
Network Requests .....	315
Internationalization .....	317
Browser and System Control .....	318
Page and Screen Capture .....	324
Proxy .....	326
Browser State Management .....	327
Tabs and Windows .....	333
Debugger .....	335
Search .....	335
Alarms .....	336
Scripting .....	336
DOM .....	338
Text to Speech .....	338

## TABLE OF CONTENTS

Privacy .....	338
Idle .....	339
Devtools .....	339
Extension Introspection .....	340
Extension Management .....	341
System State .....	341
Enterprise Only .....	342
Firefox Only .....	342
Chrome OS Only .....	343
Deprecated .....	344
Summary .....	344
<b>Chapter 12: Permissions .....</b>	<b>345</b>
Permissions Basics .....	345
Checking Permissions .....	348
Using Optional Permissions .....	350
Granting Permissions Declaratively vs. Imperatively .....	351
Permission Request Idempotence .....	352
Host Permissions .....	353
Permissions Lifetime .....	354
Permissions Warnings .....	354
Testing Permissions Warnings .....	356
Considerations for Published Extensions .....	359
Triggering the Slow Review Queue .....	359
Auto-Disable Updates .....	360
Permissions List .....	361
activeTab .....	361
alarms .....	363

## TABLE OF CONTENTS

<b>background</b> .....	363
<b>bookmarks</b> .....	363
<b>browserSettings</b> .....	364
<b>browsingData</b> .....	364
<b>captivePortal</b> .....	364
<b>certificateProvider</b> .....	364
<b>clipboardRead</b> .....	365
<b>clipboardWrite</b> .....	365
<b>contentSettings</b> .....	365
<b>contextMenus</b> .....	366
<b>contextualIdentities</b> .....	366
<b>cookies</b> .....	366
<b>debugger</b> .....	366
<b>declarativeContent</b> .....	367
<b>declarativeNetRequest</b> .....	367
<b>declarativeNetRequestFeedback</b> .....	367
<b>declarativeWebRequest</b> .....	367
<b>desktopCapture</b> .....	368
<b>Devtools page</b> .....	368
<b>dns</b> .....	368
<b>documentScan</b> .....	368
<b>downloads</b> .....	369
<b>downloads.open</b> .....	369
<b>enterprise.deviceAttributes</b> .....	369
<b>enterprise.hardwarePlatform</b> .....	369
<b>enterprise.networkingAttributes</b> .....	370
<b>enterprise.platformKeys</b> .....	370
<b>experimental</b> .....	370

## TABLE OF CONTENTS

fileBrowserHandler .....	370
fileSystemProvider .....	370
find .....	371
fontSettings .....	371
gcm.....	371
geolocation .....	371
history .....	372
Host Permissions.....	372
identity.....	374
idle.....	374
loginState .....	374
management .....	374
menus.....	375
menus.overrideContext .....	375
nativeMessaging .....	375
notifications.....	376
pageCapture .....	376
pkcs11 .....	376
platformKeys .....	377
power .....	377
printerProvider.....	377
printing .....	377
printingMetrics .....	378
privacy .....	378
processes .....	378
proxy.....	378
scripting .....	379
search.....	379

## TABLE OF CONTENTS

sessions.....	379
signedInDevices .....	379
storage .....	379
system.cpu .....	380
system.display.....	380
system.memory.....	380
system.storage.....	380
tabCapture.....	380
tabGroups .....	381
tabHide .....	381
tabs.....	381
theme .....	382
topSites .....	382
tts .....	382
ttsEngine .....	382
unlimitedStorage .....	383
vpnProvider .....	383
wallpaper.....	383
webNavigation.....	383
webRequest.....	384
webRequestBlocking .....	384
Summary.....	384
<b>Chapter 13: Networking.....</b>	<b>385</b>
Comparing Websites and Extensions .....	385
Networking Architecture .....	387
Options Pages.....	387
Popup and Devtools Pages .....	387

## TABLE OF CONTENTS

Content Scripts .....	388
Background Scripts .....	389
Pinning an Extension ID .....	389
Authentication Styles .....	392
No Authentication .....	392
Content Script Spoofing.....	393
Cookie Authentication.....	393
Json Web Token Authentication.....	394
OAuth and OpenID .....	394
OAuth, OpenID, and the Identity API .....	395
OAuth API Methods .....	396
OAuth Redirect URLs .....	396
Configuring the Authorization Platform .....	397
Additional Help .....	403
OAuth and OpenID Examples.....	403
Google OAuth with getToken() .....	404
Google OpenID with launchWebAuthFlow( ) .....	406
Manual Github OAuth with launchWebAuthFlow() .....	409
Networking APIs.....	412
The webNavigation API.....	413
The webRequest API.....	416
The declarativeNetRequest API .....	420
Summary.....	427
<b>Chapter 14: Extension Development and Deployment .....</b>	<b>429</b>
Local Development .....	429
Inspecting Your Extension .....	430
File Changes .....	438

## TABLE OF CONTENTS

Error Monitoring .....	439
Extension Reloads .....	441
<b>Automated Extension Tests .....</b>	<b>442</b>
Unit Tests .....	443
Integration Tests .....	449
Additional Reading .....	450
<b>Publishing Extensions .....</b>	<b>451</b>
Store Listing .....	451
Privacy Practices .....	452
<b>Review Process.....</b>	<b>452</b>
<b>Updating Extensions .....</b>	<b>452</b>
Update Considerations .....	452
Cancelling Updates.....	453
Automated Update Publishing .....	454
<b>Tracking User Activity.....</b>	<b>454</b>
Dashboard Metrics .....	455
Analytics Libraries .....	455
Setting Up Google Analytics.....	456
Install and Uninstall Events .....	457
Summary.....	458
<b>Chapter 15: Cross-Browser Extensions .....</b>	<b>459</b>
Introduction to Cross-Browser Support .....	459
Browser Coverage Tradeoffs .....	460
Browser Share .....	460
Chromium Browser Extension Sharing.....	461

## TABLE OF CONTENTS

<b>Adapting Your Codebase .....</b>	<b>462</b>
<b>API Probing .....</b>	<b>462</b>
<b>Differential Manifests .....</b>	<b>463</b>
<b>Manifest v2/v3.....</b>	<b>463</b>
<b>Extension Marketplaces.....</b>	<b>464</b>
<b>Marketplace Similarities.....</b>	<b>464</b>
<b>Marketplace Differences .....</b>	<b>465</b>
<b>Chrome Web Store.....</b>	<b>465</b>
<b>Add-ons for Firefox.....</b>	<b>466</b>
<b>Microsoft Edge Add-ons .....</b>	<b>467</b>
<b>Safari Extensions App Store .....</b>	<b>468</b>
<b>Opera Addons .....</b>	<b>469</b>
<b>Mobile Extensions .....</b>	<b>470</b>
<b>Mobile Extension User Interfaces .....</b>	<b>470</b>
<b>Kiwi Browser .....</b>	<b>471</b>
<b>Firefox Mobile.....</b>	<b>472</b>
<b>iOS Safari.....</b>	<b>474</b>
<b>Automated Deployment.....</b>	<b>476</b>
<b>WebExtensions API Support .....</b>	<b>477</b>
<b>Safari Extension Development .....</b>	<b>478</b>
<b>Prerequisites .....</b>	<b>478</b>
<b>Architecture.....</b>	<b>479</b>
<b>Creating an Extension Project .....</b>	<b>479</b>
<b>Writing the App .....</b>	<b>482</b>
<b>Testing on macOS.....</b>	<b>488</b>
<b>Testing on iOS.....</b>	<b>494</b>
<b>Deploying to the App Store .....</b>	<b>502</b>
<b>Converting an Existing Extension .....</b>	<b>504</b>

## TABLE OF CONTENTS

Firefox Idiosyncrasies .....	505
Manifest Versions .....	505
Sidebars .....	506
API Additions.....	507
Summary.....	508
<b>Chapter 16: Tooling and Frameworks.....</b>	<b>509</b>
Building Extensions with React.....	509
Single Entrypoint React Extensions.....	510
Multiple Entrypoint React Extensions.....	513
Reactive State Management .....	514
Routing .....	515
Mozilla Tools.....	517
web-ext .....	517
webextension-polyfill .....	518
Bundlers and CLI Tools.....	518
Parcel .....	519
Webpack.....	520
Plasmo .....	521
High-level Overview .....	522
JavaScript Frameworks.....	523
Documentation and Examples.....	523
Differential Build Outputs .....	524
Automatic Manifest Generation.....	524
Icon Generation .....	525
Bundling Remote Code .....	526
Environment Variables.....	526
Content Script Mounting.....	527

## TABLE OF CONTENTS

Extension-friendly Hot Module Replacement .....	529
Automated Deployment with Browser Platform Publish .....	530
Useful Sites .....	531
Summary.....	532
<b>Index.....</b>	<b>533</b>

# About the Author



**Matt Frisbie** has worked in web development for over a decade. During that time, he's been a startup co-founder, an engineer at a Big Four tech company, and the first engineer at a Y Combinator startup that would eventually become a billion-dollar company. As a Google software engineer, Matt worked on both the AdSense and Accelerated Mobile Pages (AMP) platforms; his code contributions run on

most of the planet's web browsing devices. Prior to this, Matt was the first engineer at DoorDash, where he helped lay the foundation for a company that has become the leader in online food delivery. Matt has written three books, *Professional JavaScript for Web Developers*, *Angular 2 Cookbook*, and *AngularJS Web Application Development Cookbook*, and recorded two video series, "Introduction to Modern Client-Side Programming" and "Learning AngularJS." He speaks at frontend meetups and webcasts, and is a level 1 sommelier. He majored in Computer Engineering at the University of Illinois Urbana-Champaign. Matt's Twitter handle is @mattfriz.

# About the Technical Reviewer



**Jeff Friesen** is a freelance software developer and educator conversant in multiple operating systems, programming languages, and numerous technologies. He is currently exploring bare metal programming for the Raspberry Pi and developing his own assembler and associating tooling to facilitate the development of a simple Pi-based operating system.

# Acknowledgments

I'd like to acknowledge the tremendous work done by editors Divya Modi, Shonmirin P.A., James Markham, and everyone else involved with the book's production. A capable and agile staff does so much to improve the quality of the product. You were all invaluable resources, and it was a pleasure working with you.

I'd also like to thank the book's technical reviewer Jeff Friesen. Technical authors are nothing without this meticulous work behind the scenes, and his contributions were nothing short of outstanding.

Finally, I would like to thank Apress for publishing this book with me. This was my first title where the book was entirely my own concept, and I am thankful that it found a home with such a terrific publisher.

# Foreword

We are Stefan and Louis, co-founders of Plasmo. We first started our company building browser extensions for cyber security, experiencing firsthand the many difficulties extension developers face. We quickly realized that the process of building an extension was way too difficult. We decided to open source our framework, and dedicated our company's vision to solving the toughest problems in browser extension development. As of today, our framework is the most popular browser extension SDK worldwide, and our fast-growing extension product suite is used by some of the world's leading extension developers.

A large community soon gathered and shared even more issues blocking their development with us. We were excited to hear that Matt was writing a book on the subject! Extension development is tricky and filled with poor documentation, difficulty getting started, a massive transition from one extension version to another looming, and much more. There has never been a comprehensive guide to extension development, but Matt's book is changing that.

People use browsers for most of their lives. You might use your browser to read the news or find a good breakfast burrito recipe when you wake up. Software engineers might use their browsers to review pull requests and test their front end. Sales representatives use their browsers to send outbound messages to prospective clients on LinkedIn, and security engineers use their browsers to review new phishing alerts. These use cases are different, but each person uses the same tool to do their job.

Browser extensions augment the browser and transform it from a generic tool into a highly specialized one. A sales representative has different needs compared to a security engineer. In a few years, sales

## FOREWORD

representatives and security engineers will have a stack of browser extensions that automate, categorize, and track their work. Some professions have picked up on this faster than others. Ask a high-performing sales representative about browser extensions they have installed, and you might hear a list of more than ten.

Browser extensions are potent but underutilized tools. The lack of know-how and a complete manual such as Matt's book has curbed people's imagination of what their browser could be. As knowledge of building extensions increases, there will be a tremendous appetite for new browser extensions that serve all kinds of niches we have never seen before.

This book encapsulates years of experience and research. It digs deep into critical details you could only find in random forums, obscure documentation, and fragmented source code online. It is everything you need to know about building browser extensions, from a well-rounded beginner-friendly introduction to a comprehensive guide for veterans. We especially love the following chapters:

- Chapter 4, “Browser Extension Architecture,” is for adept developers who have yet to work with browser extensions.
- Chapter 6, “Understanding the Implications of Manifest V3,” is for developers who need immediate help transitioning their extension to the newer manifest version.
- Chapter 16, “Tooling and Frameworks,” is a must-read for developers who want to supercharge their workflow with robust frameworks and DevOps/CI automation to produce enterprise-grade browser extensions.

We can't wait to use your extension one day!

Stefan and Louis  
Co-founders of Plasmo

# Introduction

The world of browser extensions has far more than meets the eye. Consider the following:

- On average, 40% of Internet users in the United States use an adblocker on any device; overwhelmingly, these adblockers take the form of browser extensions.
- The tech company Honey, whose primary product is a browser extension, was acquired by PayPal in 2020 for \$4 billion.
- As of 2021, there were 1.8 million apps in Apple's App Store; the Chrome Web Store has 180,000 extensions.

When I saw there were 0 relevant Amazon search results for “build chrome extension,” I nearly fell out of my chair. I knew at once that this book *must* be written.

*Building Browser Extensions: Create Modern Extensions for Chrome, Safari, Firefox, and Edge* covers all the knowledge you will need to write cross-browser extensions with the latest web development tools. Browser extensions are given access to *extremely* powerful APIs. I believe most developers are blind to that power – and unaware of just how much it is within their reach.

This book is designed to enlighten web developers and illuminate the true potential of the browser extension software platform. It is geared for developers who have experience building websites and can apply their knowledge to a new software domain. This book is not ideal for people new to programming – it would be like an inexperienced cook starting off by learning to make a sauce.

## INTRODUCTION

A major barrier to developing browser extensions is the appalling status quo of documentation. The fragmentation between different browsers and different manifest versions turns slogging through the documentation into a mind-numbingly onerous affair. I wrote this book specifically to address this problem. The reader will learn what is possible with the APIs, how they can best be applied, and all the traps to avoid. The book is not intended to *replace* the API documentation, as it is changing all the time. Instead, it is intended to supplement the API documentation; the book has plenty of direct links to the Chrome Developers and MDN sites throughout.

The transition to manifest v3 is upon us, and already it is causing problems. If you are confused about what manifest v3 is, what are its implications, and how best to navigate the ongoing transition, this book is for you. I dedicated an entire chapter to the manifest v2/v3 transition.

The lingua franca of web development is React, and this book gives special attention to the best ways in which you can write a browser extension in React. It also covers all the supplemental tools you'll need along the way, such as Webpack, Parcel, and Plasmo.

Like many developers, I learn by example. I was annoyed that so many APIs listed in the documentation were totally inscrutable. For example, the omnibox API is amazing and incredibly useful, but the documentation on how to use it is *garbage*. I just wanted a simple example to pick apart and play with, and there was nothing to be found. To fill this need, I created a companion extension for the book: *Browser Extension Explorer*. It's an open source browser extension with dozens of interactive demos. Each demo shows how various browser extension pieces and APIs work, and each includes links to the specific source files so you can see how it was built.

You can download *Browser Extension Explorer* from the Chrome Web Store, or find a link to it on the companion website to this book: <https://buildingbrowserextensions.com>.

## CHAPTER 1

# What Are Browser Extensions?

Browser extensions are strange and powerful parasites. Browser extensions are parasites because they reside in an unusual corner of the world of software: they run on top of web pages yet are totally independent of any website or server. Browser extensions are powerful because they are afforded an unusual degree of agency: some browser extensions can view and modify everything you see in your browser, some can view every HTTP request your browser sends, and others can manage your digital currency wallets. Browser extensions are strange because they can directly antagonize the very companies that support them: uBlock Origin, one of the most popular Google Chrome extensions, exists only to prevent showing the very ads that make up Google's primary revenue channel.

Strange and powerful as they may be, browser extensions are an undoubtedly useful tool. They afford the end user a degree of control over their web browsing experience that is not possible in any other way. The most popular browser extensions have tens of millions of installs, and as of 2022, the Chrome Web Store has over 180,000 extensions published. In recognition of the broad adoption and utility, in 2021 the W3C formed the WebExtensions Community Group to “explore how browser vendors and other interested parties can work together to advance a common browser extension platform.”