

## Auf einen Blick

1	Im Web, als App: Geschichte und Einstieg .....	25
2	Mächtiges modernes Web .....	67
3	Zehn Eigenschaften, die PWA einzigartig machen .....	99
4	Web App Manifest: Aussehen der App definieren .....	131
5	Service Worker: Einer muss ja arbeiten .....	171
6	Cache API: So lädt die App auch ohne Netzverbindung .....	225
7	Workbox .....	261
8	Push API: Rufen Sie nicht uns an – wir rufen Sie an! .....	285
9	PWA und Angular: Single-Page-Application-Framework einsetzen ....	321
10	App-like aussehen .....	373
11	Plattformverhalten .....	387
12	Alles richtig gemacht? – PWAs validieren mit Lighthouse & Co. ....	401
13	Migrationsstrategien mit Apache Cordova und GitHub Electron .....	417
14	Payment Request API: Wie Sie trotz fehlendem App Store an Ihr Geld kommen .....	463
15	Brandheiße Progressive Web Apps .....	495
16	Fazit: Eine Codebasis, alle Plattformen .....	503

# Inhalt

Materialien zum Buch .....	17
Geleitwort .....	19
Vorwort .....	21
<b>1 Im Web, als App: Geschichte und Einstieg</b> .....	<b>25</b>
<b>1.1 Wie Apps auf unser Handy kamen – eine kleine Zeitreise</b> .....	<b>27</b>
1.1.1 Der PC für die Hosentasche .....	28
1.1.2 Der Browser als Anwendungsplattform .....	28
1.1.3 Es gibt eine App dafür: der Siegeszug des App Store .....	30
1.1.4 Das Verhältnis von Web und Apps .....	34
<b>1.2 Progressive Web Apps: wohin die Reise der Anwendungsentwicklung geht</b> .....	<b>35</b>
1.2.1 Die Plattformproblematik .....	37
1.2.2 Das Web ist die Plattform .....	38
1.2.3 Das Rückgrat der Progressive Web Apps .....	39
1.2.4 Progressive Web Apps als Über-Pattern .....	41
<b>1.3 Voraussetzungen und Basics: welches Wissen Sie schon mitbringen sollten</b> .....	<b>42</b>
1.3.1 HTML, CSS, JavaScript und TypeScript .....	42
1.3.2 Single-Page Web Applications .....	43
1.3.3 Promises: asynchrone Operationen im Griff .....	45
1.3.4 Mobile Anwendungsentwicklung .....	47
<b>1.4 Tools installieren: das Handwerkszeug zur PWA-Entwicklung</b> .....	<b>48</b>
1.4.1 Visual Studio Code .....	48
1.4.2 Git .....	50
1.4.3 Google Chrome .....	51
1.4.4 Node.js .....	51
1.4.5 Angular CLI .....	53
1.4.6 Workbox CLI .....	53
1.4.7 ngrok .....	54
<b>1.5 Setup: Das erste PWA-Projekt</b> .....	<b>54</b>
1.5.1 Dateien anlegen .....	55
1.5.2 »index.html« – der Einsprungspunkt für Ihre PWA .....	56
1.5.3 »sw.js« – das Service-Worker-Skript .....	58

1.5.4	Website mit lite-server ausprobieren .....	61
1.5.5	Website mit ngrok auf einem Mobilgerät testen .....	63
<b>1.6</b>	<b>Zusammenfassung .....</b>	<b>65</b>
<b>2</b>	<b>Mächtiges modernes Web .....</b>	<b>67</b>
<b>2.1</b>	<b>Audio-/Videoelement: Multimediainhalte ohne Plug-in wiedergeben ....</b>	<b>70</b>
<b>2.2</b>	<b>Canvas-Element: ansprechende 2D- und 3D-Visualisierungen .....</b>	<b>72</b>
2.2.1	Paper.js .....	72
2.2.2	THREE.js .....	74
<b>2.3</b>	<b>Gamepad API: App mit dem Game-Controller steuern .....</b>	<b>78</b>
<b>2.4</b>	<b>WebAssembly: Binärcode für das Web mit nahezu nativer Performance .....</b>	<b>82</b>
<b>2.5</b>	<b>Web Share API: Teilen von Inhalten aus dem Browser heraus .....</b>	<b>84</b>
<b>2.6</b>	<b>Web Speech API: Text-to-Speech im Browser .....</b>	<b>87</b>
<b>2.7</b>	<b>Media Capture and Streams: auf Kamera und Mikrofon zugreifen .....</b>	<b>88</b>
<b>2.8</b>	<b>Generic Sensor API: Zugriff auf die Gerätesensoren .....</b>	<b>91</b>
<b>2.9</b>	<b>Pointer Events und Pressure.js: Force Touch im Web .....</b>	<b>93</b>
<b>2.10</b>	<b>Geolocation: Implementierung standortbezogener Dienste .....</b>	<b>94</b>
<b>2.11</b>	<b>Zusammenfassung .....</b>	<b>98</b>
<b>3</b>	<b>Zehn Eigenschaften, die PWA einzigartig machen .....</b>	<b>99</b>
<b>3.1</b>	<b>Voranschreiten mit Progressive Enhancement .....</b>	<b>100</b>
<b>3.2</b>	<b>App-ähnlich: Sieht aus wie eine App, fühlt sich an wie eine App .....</b>	<b>103</b>
<b>3.3</b>	<b>Verbindungsunabhängigkeit: Kein Funkloch hält Sie auf .....</b>	<b>106</b>
<b>3.4</b>	<b>Immer schön frisch bleiben: der Service-Worker-Updateprozess .....</b>	<b>109</b>
<b>3.5</b>	<b>Sicher: Mit großer Macht kommt große Verantwortung .....</b>	<b>110</b>
3.5.1	Sicherheitsmaßnahmen im Webbrowser .....	110
3.5.2	Sichere Verbindungen mit HTTPS .....	112
<b>3.6</b>	<b>Einer für alle: Responsive Webdesign .....</b>	<b>115</b>

<b>3.7</b>	<b>Auffindbarkeit: Websites und Apps unterscheiden .....</b>	<b>118</b>
<b>3.8</b>	<b>Installierbarkeit: So kommt Ihre PWA auf den Homebildschirm .....</b>	<b>120</b>
<b>3.9</b>	<b>Nutzer binden: Anwender mit Pushbenachrichtigungen zurückholen .....</b>	<b>123</b>
<b>3.10</b>	<b>Verlinkbar: auf Anwendungen und Zustände verweisen .....</b>	<b>126</b>
3.10.1	Hash-basiertes Routing .....	127
3.10.2	Pfad-basiertes Routing .....	128
3.10.3	Deep Linking und Link Capturing .....	129
<b>3.11</b>	<b>Zusammenfassung .....</b>	<b>129</b>
<b>4</b>	<b>Web App Manifest: Aussehen der App definieren .....</b>	<b>131</b>
<b>4.1</b>	<b>App-Aussehen auf dem Homebildschirm anpassen .....</b>	<b>135</b>
4.1.1	Name .....	135
4.1.2	Icons .....	136
4.1.3	Farben .....	142
4.1.4	Splashscreen .....	143
4.1.5	Weitere Metadaten .....	146
<b>4.2</b>	<b>App-Verhalten anpassen .....</b>	<b>146</b>
4.2.1	Start-URL .....	146
4.2.2	Scope .....	147
4.2.3	Anzeigemodi .....	148
4.2.4	Bildschirmausrichtung .....	153
4.2.5	Verwandte Apps .....	154
4.2.6	Service Worker .....	155
4.2.7	Alterskennzeichnung .....	155
<b>4.3</b>	<b>Web App Manifest referenzieren .....</b>	<b>156</b>
<b>4.4</b>	<b>Zur Installation auffordern .....</b>	<b>157</b>
4.4.1	Auslaufmodell App-Install-Banner .....	159
4.4.2	Eigene Methode .....	160
4.4.3	Manuelle Methode .....	163
4.4.4	Deinstallation .....	165
<b>4.5</b>	<b>Zukunftsmusik: Badging API .....</b>	<b>166</b>
<b>4.6</b>	<b>Microsoft Store Ingestion: Wie die App den Weg in den Microsoft Store findet .....</b>	<b>168</b>

4.7	PWA Builder .....	170
4.8	Zusammenfassung .....	170
<b>5</b>	<b>Service Worker: Einer muss ja arbeiten</b> .....	<b>171</b>
5.1	Vom Web Worker zum Service Worker .....	171
5.1.1	Web Worker .....	172
5.1.2	Shared Worker .....	173
5.1.3	Service Worker .....	174
5.2	Kontrollzwang mit Vorteilen: Service Worker als zentraler Proxy .....	176
5.2.1	Wo lebt der Service Worker in meinem Browser? .....	178
5.2.2	Erst mal offline: das Offline-First-Paradigma .....	178
5.2.3	IndexedDB: Wer noch alles im Offlineteam mitspielt .....	180
5.3	Lebenszyklus .....	188
5.3.1	Registrieren .....	189
5.3.2	Installieren .....	194
5.3.3	Aktivieren .....	197
5.3.4	Überschreiben .....	199
5.4	Schnittstellen .....	199
5.4.1	Netzanfragen abfangen und manipulieren .....	200
5.4.2	Briefchen schreiben mit postMessage .....	202
5.4.3	Mit Clients interagieren .....	205
5.5	Wir alle machen Fehler: Service Worker debuggen .....	207
5.5.1	Google Chrome .....	207
5.5.2	Mozilla Firefox .....	213
5.5.3	Apple Safari .....	214
5.5.4	Microsoft Edge .....	216
5.6	Background Sync API .....	217
5.6.1	Progressive Enhancement berücksichtigen .....	218
5.6.2	Registrieren eines Synchronisationsereignisses .....	219
5.6.3	Hintergrundsynchronisation ausführen .....	220
5.7	Navigation Preload .....	221
5.8	Zusammenfassung .....	223

<b>6</b>	<b>Cache API: So lädt die App auch ohne Netzverbindung</b> .....	<b>225</b>
6.1	HTTP-Crashkurs: So war das noch mal mit Anfragen und Antworten .....	226
6.1.1	Auslaufmodell XMLHttpRequest .....	229
6.1.2	Fetch API .....	231
6.2	Auslaufmodell Application Cache .....	235
6.3	Caches verwalten .....	236
6.3.1	Caches benennen .....	236
6.3.2	Caches öffnen .....	237
6.3.3	Caches auflisten und löschen .....	237
6.4	Antworten zur Seite legen: Man weiß nie, wann man sie wieder gebrauchen kann .....	238
6.4.1	Abfrage zwischenspeichern .....	238
6.4.2	Kombination mit dem Service Worker: install und addAll .....	241
6.4.3	Speicherkontingent abfragen .....	243
6.4.4	Websitespeicher persistent machen .....	245
6.5	It's a match! – Passende Antworten aus dem Cache holen .....	246
6.5.1	Einzelantwort beziehen .....	246
6.5.2	Kombination mit dem Service Worker: fetch und match .....	247
6.5.3	Mehrere Antworten beziehen .....	249
6.5.4	Alle Zwischenspeicher durchsuchen .....	250
6.6	Cacheeinträge löschen .....	250
6.7	Caches debuggen .....	251
6.8	Caching-Strategien .....	252
6.8.1	Cache Only .....	253
6.8.2	Network Only .....	254
6.8.3	Cache falling back to network .....	254
6.8.4	Generic Fallback .....	255
6.8.5	Network falling back to cache .....	256
6.8.6	Cache then network .....	257
6.8.7	Cache & network race .....	258
6.8.8	Service-Worker-Side Templating .....	259
6.9	Weitere Service-Worker-Use-Cases .....	259
6.10	Zusammenfassung .....	260

<b>7</b>	<b>Workbox</b>	261
<b>7.1</b>	<b>Befehle der Workbox CLI</b>	262
<b>7.2</b>	<b>Workbox-Projekt aufsetzen</b>	263
<b>7.3</b>	<b>Precaching</b>	265
<b>7.4</b>	<b>Runtime Caching</b>	270
7.4.1	Routen und Caching-Strategien	271
7.4.2	Konfiguration über runtimeCaching	271
7.4.3	Plug-ins	274
7.4.4	Navigations-Fallbacks	275
<b>7.5</b>	<b>Service Worker erweitern</b>	276
7.5.1	Precaching	276
7.5.2	Runtime-Caching	279
<b>7.6</b>	<b>Workbox in den Buildprozess integrieren</b>	279
7.6.1	Node.js und Gulp	280
7.6.2	Webpack	281
<b>7.7</b>	<b>Navigation Preload aktivieren</b>	282
<b>7.8</b>	<b>Offlineanalysedaten erfassen</b>	282
<b>7.9</b>	<b>Zusammenfassung</b>	283
<b>8</b>	<b>Push API: Rufen Sie nicht uns an – wir rufen Sie an!</b>	285
<b>8.1</b>	<b>Das Push-Prinzip</b>	286
<b>8.2</b>	<b>Nur eine Chance: Pushregistrierung beantragen</b>	287
8.2.1	Progressive Enhancement berücksichtigen	288
8.2.2	Pushregistrierung auslösen	288
8.2.3	Einwilligung des Anwenders einholen	291
<b>8.3</b>	<b>Informationsaustausch</b>	293
<b>8.4</b>	<b>Den Server Pushnachrichten verschicken lassen</b>	297
<b>8.5</b>	<b>Pushereignisse behandeln</b>	302
8.5.1	Pushereignisse entgegennehmen	303
8.5.2	Benachrichtigungsbanner anzeigen	303
8.5.3	Auf die Auswahl des Anwenders reagieren	307
<b>8.6</b>	<b>Sonderfall Apple</b>	310

<b>8.7</b>	<b>Drittanbieterdienste: OneSignal &amp; Co.</b>	314
8.7.1	OneSignal	314
8.7.2	PushCrew	315
8.7.3	Pushpad	315
<b>8.8</b>	<b>Pushnachrichten zur Laufzeit</b>	316
8.8.1	socket.io	316
8.8.2	ASP.NET Core SignalR	317
8.8.3	Anzeige von Benachrichtigungen	318
<b>8.9</b>	<b>Zusammenfassung</b>	319
<b>9</b>	<b>PWA und Angular: Single-Page-Application-Framework einsetzen</b>	321
<b>9.1</b>	<b>Projekt-Setup</b>	323
9.1.1	Angular CLI	323
9.1.2	Neues Projekt anlegen	324
<b>9.2</b>	<b>Responsive und App-like: Navigationsgrundgerüst mit Angular Material</b>	325
9.2.1	Angular Material installieren	326
9.2.2	App Shell implementieren	327
<b>9.3</b>	<b>Linkable: Routing implementieren</b>	329
9.3.1	Komponenten generieren	329
9.3.2	Routen anpassen	330
9.3.3	Navigationsleiste anpassen	332
<b>9.4</b>	<b>App-Like, die zweite: Moderne Web-APIs einsetzen</b>	333
<b>9.5</b>	<b>PWA-Unterstützung installieren</b>	335
<b>9.6</b>	<b>Discoverable: Web App Manifest anpassen</b>	336
<b>9.7</b>	<b>Connectivity Independent, die erste: Quelldateien der Anwendung offlinefähig machen</b>	338
9.7.1	Service Worker registrieren	339
9.7.2	Caching des Angular Service Workers konfigurieren	339
9.7.3	Produktionsbuild ausführen	344
9.7.4	Debugging	346
9.7.5	Grenzen des Angular Service Workers	348
9.7.6	Service Worker entfernen	348

<b>9.8 Connectivity Independent, die zweite: strukturierte Daten zwischenspeichern</b>	348
9.8.1 Modellklasse generieren	349
9.8.2 Service anlegen	349
9.8.3 To-do-Einträge hinzufügen	350
9.8.4 To-do-Einträge abrufen	353
9.8.5 To-do-Einträge als erledigt markieren	354
9.8.6 To-do-Einträge synchronisieren	356
<b>9.9 Reengageable: Pushereignisse mit SwPush</b>	359
<b>9.10 Fresh: Updateprozess mit SwUpdate</b>	364
<b>9.11 Installable: Installation anbieten</b>	366
<b>9.12 Angular Universal: mit Server-Side-Rendering zur App Shell</b>	368
<b>9.13 PRPL-Entwurfsmuster</b>	369
<b>9.14 Zusammenfassung</b>	370
<b>10 App-like aussehen</b>	373
<hr/>	
<b>10.1 Native Schriftarten einsetzen</b>	373
<b>10.2 Textauswahl und Link-Highlighting verhindern</b>	375
10.2.1 Standardcursor setzen	375
10.2.2 Textauswahl verhindern	376
10.2.3 Link-Highlighting verhindern	377
10.2.4 Keine Callouts	378
<b>10.3 App-like Anwendungsframeworks</b>	379
10.3.1 ngx-admin	379
10.3.2 Angular Material	380
10.3.3 Framework7	380
<b>10.4 Notches unterstützen</b>	381
<b>10.5 Zusammenfassung</b>	386
<b>11 Plattformverhalten</b>	387
<hr/>	
<b>11.1 macOS</b>	387
<b>11.2 iOS</b>	389

<b>11.3 Android</b>	391
<b>11.4 Windows</b>	395
<b>11.5 Linux</b>	397
<b>11.6 Zusammenfassung</b>	399
<b>12 Alles richtig gemacht? – PWAs validieren mit Lighthouse &amp; Co.</b>	401
<hr/>	
<b>12.1 Barrierefreiheit testen mit aXe</b>	402
<b>12.2 Lighthouse: der Leuchtturm der Websitevalidierung</b>	405
12.2.1 Performance	407
12.2.2 Progressive Web App	409
12.2.3 Best Practices	410
12.2.4 Accessibility	410
12.2.5 Search-Engine Optimization	410
<b>12.3 webhint: ein Linter für das Web</b>	411
12.3.1 Accessibility	414
12.3.2 Interoperability	414
12.3.3 Performance	415
12.3.4 Progressive Web App	415
12.3.5 Security	416
<b>12.4 Zusammenfassung</b>	416
<b>13 Migrationsstrategien mit Apache Cordova und GitHub Electron</b>	417
<hr/>	
<b>13.1 Apache Cordova</b>	419
13.1.1 Projekt anlegen und konfigurieren	420
13.1.2 Plattformen	423
13.1.3 Ereignisse	427
13.1.4 Plug-ins	429
13.1.5 Anwendung bauen	437
13.1.6 Ionic	438
<b>13.2 GitHub Electron</b>	440
13.2.1 Projektkonfiguration	441

13.2.2	Ereignisse .....	443
13.2.3	Mit nativen Schnittstellen interagieren .....	448
13.2.4	Electron-Packager .....	453
<b>13.3</b>	<b>Plattformunterschiede elegant verbergen .....</b>	<b>455</b>
<b>13.4</b>	<b>Chromium Embedded Framework: Desktopanwendungen schrittweise entkernen .....</b>	<b>459</b>
<b>13.5</b>	<b>Zusammenfassung und Fazit .....</b>	<b>460</b>

## **14 Payment Request API: Wie Sie trotz fehlendem App Store an Ihr Geld kommen** 463

---

<b>14.1</b>	<b>Warum Check-out-Formulare nicht die Lösung sind .....</b>	<b>464</b>
<b>14.2</b>	<b>Einfaches Check-out mit der Payment Request API .....</b>	<b>466</b>
14.2.1	Zahlungsmethode .....	468
14.2.2	Zahlungsdetails .....	468
14.2.3	Optionen .....	470
<b>14.3</b>	<b>Ablauf einer Zahlungsanforderung .....</b>	<b>470</b>
14.3.1	Prüfung und Anzeige der Zahlungsanforderung .....	471
14.3.2	Auf Änderungen reagieren .....	472
14.3.3	Abschluss des Verkaufs .....	473
<b>14.4</b>	<b>Payment Method: Basic Card .....</b>	<b>474</b>
<b>14.5</b>	<b>Google Pay .....</b>	<b>478</b>
<b>14.6</b>	<b>Apple Pay .....</b>	<b>482</b>
14.6.1	Konfiguration des Zahlungsmethodenobjekts .....	483
14.6.2	Extraschritt: Validierung des Händlers .....	485
14.6.3	Antwortobjekt auswerten .....	486
<b>14.7</b>	<b>Fazit: Viele Wege führen zum Fallback .....</b>	<b>489</b>
<b>14.8</b>	<b>Ausblick: Payment Handler API .....</b>	<b>491</b>
<b>14.9</b>	<b>Zusammenfassung .....</b>	<b>492</b>

## **15 Brandheiße Progressive Web Apps** 495

---

<b>15.1</b>	<b>Twitter Lite .....</b>	<b>496</b>
<b>15.2</b>	<b>Financial Times .....</b>	<b>497</b>

<b>15.3</b>	<b>Telegram .....</b>	<b>498</b>
<b>15.4</b>	<b>Pokédex .....</b>	<b>499</b>
<b>15.5</b>	<b>QR Scanner .....</b>	<b>500</b>
<b>15.6</b>	<b>Zusammenfassung .....</b>	<b>502</b>

## **16 Fazit: Eine Codebasis, alle Plattformen** 503

---

<b>16.1</b>	<b>Ideales technologisches Umfeld .....</b>	<b>503</b>
<b>16.2</b>	<b>Interessen der Plattformhersteller .....</b>	<b>504</b>
<b>16.3</b>	<b>Wer heute schon PWAs baut .....</b>	<b>506</b>
<b>16.4</b>	<b>Limitationen .....</b>	<b>506</b>
<b>16.5</b>	<b>Chancen .....</b>	<b>507</b>
<b>16.6</b>	<b>Ausblick .....</b>	<b>508</b>

Über den Autor .....	511
Index .....	513