



PWA-Literatur

Ich nenne die Bücher in der Reihenfolge, wie sie meiner Ansicht nach am schnellsten in die Materie einführen. Anstatt Bücher zu kaufen, können auch Artikel im Internet gelesen werden. Der Vorteil: oft werden Beispielprogramme angeboten, die dann auch sofort ausprobieren werden können. Viele Beiträge sind auf Englisch.

Die *Codelabs* von Google sind gut aufbereitet. Allerdings versucht Google, sein Produkt *Firebase* in den Vordergrund zu stellen. *Firebase* ist nur für kleinere Projekte kostenlos. Weitere Bedenken: wo die Daten gespeichert sind, ist nicht klar – und schon sind wir mitten in der Problematik der Datenschutzgrundverordnung (siehe weiter unten).

<https://codelabs.developers.google.com/>

Ein neutrales Angebot kommt von der Mozilla Foundation:

<https://developer.mozilla.org/de/>

Zum Thema PWA:

https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps

Aber nun zurück zur Buchliste:

Progressive Web Apps

(Marcel Peters. ISBN 978 1521881644. Deutsch)

Das Buch ist gut geeignet, das Interesse an PWAs zu wecken. Der Preis ist nicht sehr hoch (13,95 € als Taschenbuch, 7,99 € als Kindle-E-Book). Wer sich nicht sicher ist, ob er in diese Materie einsteigen will, geht damit kein großes Risiko ein.

Das Buch verzichtet leider auf *Preact* und aktuelles *ECMAScript*. Die Beispiele können ohne großen Aufwand und mit Erfolg nachvollzogen werden.

Es gibt ein Literaturverzeichnis. Allerdings könnte durchaus prägnanter hervor gehoben werden, dass das Buch über weitere Strecken die Inhalte der entsprechenden Google-Kurse samt Abbildungen (jedoch auf Deutsch übersetzt) wiedergibt. Apropos Abbildungen: das Layout trägt durchaus eine Verbesserung. Wenn vier Screenshots in der Abbildung 3 dargestellt werden, trägt ein kleiner Abstand oder Rand dazwischen zur Übersichtlichkeit bei. Und das gilt auch für andere Abbildungen.

Building Progressive Web Apps

(Tal Ater. ISBN 978 1491961650. Englisch. 33,82 €)

Am Beispiel einer Webseite für das fiktive „Gotham Hotel“ wird Schritt für Schritt gezeigt, wie PWAs zu implementieren sind.

Positiv:

Nach jedem Kapitel liegt eine funktionsfähige App vor, das dann im folgenden Kapitel erweitert wird. Und wenn einmal etwas schiefgeht, steht der korrekte Programmcode für jedes Kapitel auf github.com bereit.

Alle Bestandteile einer PWA werden sehr detailliert erklärt. Während andere Bücher oft ein fertiges Programm vorsetzen, das dann Zeile für Zeile analysiert wird, werden hier neue Elemente zuerst in ihrer Grundstruktur erklärt und dann immer weiter ausgebaut und gleichzeitig verfeinert. Diese Vorgangsweise führt zu

einem wesentlich besseren Verständnis. Ein Beispiel: die Verwendung lokaler Datenbanken ist für PWAs sehr wichtig. Dem Umgang mit der Datenbank *IndexedDB* werden hier 42 Seiten gewidmet: Beispiele werden genau erklärt und immer weiter ausgebaut.

Wer das Buch genau durcharbeitet, bekommt eine sehr gute Vorstellung, wie PWAs aufgebaut sind. Besonderer Wert wird auch darauf gelegt, genau zu erklären, wie die wichtigen Komponenten zusammen arbeiten.

Negativ:

Neue *ECMAScript*-Versionen erlauben wesentlich kompaktere Schreibweisen, die aber nicht verwendet werden. Dadurch wird der Programmcode – vor allem bei *Callbacks* – rasch unübersichtlich.

Preact erfordert etwa 4kB an Speicher – eine wirklich kleine Bibliothek. Im Kapitel 6 taucht nun – quasi ohne Vorwarnung – die Verwendung von *jQuery* auf. Das ist ungewöhnlich, da eine *jQuery*-Bibliothek den Programmcode kräftig aufbläht. *Na ja, mit der jQuery-Alternative zepto.js sind es auch nur 9,6KB – aber da geht es ums Prinzip!*

Mobiltelefone bieten viele Zusatzfunktionen: eine eingebaute Kamera, einen GPS-Empfänger, einen Lage- und Beschleunigungssensor usw. Diese Komponenten waren vorerst nur für native Apps zugänglich. Jetzt erlauben *Preact*- und *React*-Programm-Bibliotheken den Zugriff auf diese Hardware und bringen PWAs auch sehr nahe an native Apps heran. Bedauerlich, dass dies im Buch nicht erwähnt wird.

Für größere Projekte sind moderne Programmansätze, wie etwa Klassen für Komponenten, sehr nützlich und wichtig. Da aber *Preact* und *React* nicht verwendet werden, gibt es auch keine Erklärungen dazu.

Andererseits verbergen *Preact* und *React* die Details der Abläufe – aus dieser Sicht ist der Einstieg ohne Bibliotheken durchaus lehrreich.

Progressive Web Apps with React

(Scott Domes. ISBN 978 1788297554. Englisch. Taschenbuch 25,91 €)

Das Projekt *Chatastrophe* wird Schritt für Schritt auf- und ausgebaut. Auch hier steht der Sourcecode im Internet zur Verfügung.

Positiv:

Aktuelle *ECMAScript*-Versionen werden verwendet.

Sehr nützlich sind auch die Hinweise, wie Programmflüsse analysiert und die Programme optimiert werden können.

Negativ:

Ab dem Kapitel 4 wird *Firebase* verwendet. Für Versuche und kleine Applikationen stellt Google *Firebase* kostenlos zur Verfügung. Bei intensiverer Nutzung fallen Kosten an, deren Höhe nicht genau vorher gesagt werden können. Außerdem ist nicht nachvollziehbar, wo Google die *Firebase*-Daten speichert und ob dabei die europäischen Datenschutzbestimmungen eingehalten werden. Es wäre gut, wenn zusätzlich gezeigt wird, wie das Projekt ohne *Firebase* verwirklicht werden kann.

Preact bietet (praktisch) alle Möglichkeiten von *React*, braucht aber viel weniger Speicherplatz.

Zu *React* gibt es eine Unmenge an Literatur, zu *Preact* viel weniger. Ist es daher notwendig, zuerst *React* zu lernen?

Progressive Web-Apps

(Manfred Steyer. ISBN 978 3868027532. Deutsch, E-Book. 2,99 €)

Das E-Book kostet nur 2,99 € und ist auf Deutsch. Es liefert einen schnellen Überblick über die Prinzipien der PWAs. Zur Realisierung eigener Projekte reicht die Beschreibung aber nicht aus – ich nenne es daher nur als Ergänzung.

Außerdem sollte schon im Titel darauf hingewiesen werden, dass *Angular-2* verwendet wird. *Angular* ist wie *React* ein *Javascript Framework* für eigene Programmentwicklungen. Dieser Artikel beschäftigt sich aber mit dem – aus meiner Sicht moderneren und einfacher zu lernenden – *Preact*. Der Aufwand, *Angular-2* zu erlernen, ist nicht zu unterschätzen.

Progressive Web Apps With Preact

(Vu Tran u.a. ISBN 978 1939902535. Englisch, Kindle E-Book 16,99 €)

Ein anderer Zugang zu dem Thema und ein guter Überblick über die wichtigsten Teile von *Preact*. An einigen Stellen wird auf den Unterschied zu *React* hingewiesen: für *React*-Kenner nützlich, für alle, die erst in die Materie einsteigen, nicht sehr hilfreich. Jedenfalls wird hervorgehoben, warum *Preact* dem speicherintensiveren *React* vorzuziehen ist.

Auch in diesem Buch wird ein Beispiel – einer Webseite für *Hacker-News* – gezeigt, wie ein PWA entsteht. Alle Beschreibungen sind sehr detailliert – für den Einstieg eventuell zu detailliert. Schön wäre es, wenn mit jedem Kapitel auch schon ein lauffähiges Programm entwickelt würde. Da ist aber nicht der Fall und es bedarf einiger Geduld, bis man als Leser Dinge auch ausprobieren kann.

An Effective Guide to Modern JavaScript

ECMAScript 2017 / ES8

(Chong Lip Phang. ISBN 978 1974207923. Englisch. 12,31 € bei Amazon)

Ja, *JavaScript* wird ständig weiter entwickelt – nicht nur in der Namensänderung zu *ECMAScript*. Klassen, Objekte, Lambda-Funktionen, Template-Strings. Die neuen Konzepte erlauben das Schreiben von übersichtlichen Programmen. *Trotzdem hat es ECMAScript 8 auch nicht geschafft, eine vernünftige Programm-Bibliothek für das Formatieren von Texten, Zahlen und Datumsangaben bereit zu stellen. Vielleicht sollten die ECMAScript-Entwickler einmal die entsprechenden Routinen von Python ein wenig näher ansehen?*

Das ist aber eine Kritik an *ECMAScript* und nicht an diesem Buch. Wer nicht die x-te Version von Büchern, die alle mit dem "Hello World"-Programm beginnen, lesen will, sondern die Konzepte (samt Beispielen!) übersichtlich geordnet lesen oder nachschlagen möchte, sollte dieses kleine Handbuch kaufen. Das ist kein Buch zum Einstieg in *ECMAScript* – grundlegende *ECMA(Java)Script*-Kenntnisse sind vorteilhaft. Sehr hilfreich, wenn man die neuen Konzepte kennen lernen oder verwenden will. Auch für das Lesen PWA-Büchern ist das Verständnis der Konstrukte wichtig.