

4.

Auflage



Ferdinand Malcher · Danny Koppenhagen · Johannes Hoppe

Angular

Das große Praxisbuch –
Grundlagen, fortgeschrittene Themen
und Best Practices

inkl. RxJS,
NgRx und
a11y



 EDITION

dpunkt.verlag

Liebe Leserin, lieber Leser,

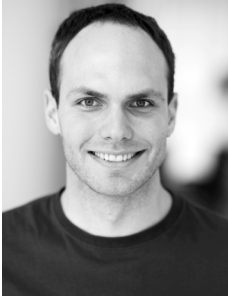
das Angular-Ökosystem wird kontinuierlich verbessert. Bitte haben Sie Verständnis dafür, dass sich seit dem Druck dieses Buchs unter Umständen Schnittstellen und Aspekte von Angular weiterentwickelt haben können. Die GitHub-Repositorys mit den Codebeispielen werden wir bei Bedarf entsprechend aktualisieren.

Unter <https://angular-buch.com/updates> informieren wir Sie ausführlich über Breaking Changes und neue Funktionen. Wir freuen uns auf Ihren Besuch.

Sollten Sie einen Fehler vermuten oder einen Breaking Change entdeckt haben, so bitten wir Sie um Ihre Mithilfe! Bitte kontaktieren Sie uns unter team@angular-buch.com mit einer Beschreibung des Problems.

Wir wünschen Ihnen viel Spaß mit Angular!

Alles Gute
Ferdinand, Danny und Johannes



Ferdinand Malcher ist Google Developer Expert (GDE) und arbeitet als selbständiger Entwickler, Berater und Mediengestalter mit Schwerpunkt auf Angular, RxJS und TypeScript. Gemeinsam mit Johannes Hoppe hat er die Angular.Schule gegründet und bietet Schulungen zu Angular an.



Danny Koppenhagen arbeitet als Softwarearchitekt und Entwickler. Sein Schwerpunkt liegt in der Frontend-Architektur und der Entwicklung von Enterprise Webanwendungen auf Basis von Node.js, TypeScript, Angular und Vue. Neben der beruflichen Tätigkeit ist Danny als Autor mehrerer Open-Source-Projekte aktiv.



Johannes Hoppe ist Google Developer Expert (GDE) und arbeitet als selbständiger Trainer und Berater für Angular, .NET und Node.js. Zusammen mit Ferdinand Malcher hat er die Angular.Schule gegründet und bietet Workshops und Beratung zu Angular an. Johannes ist Organisator des Angular Heidelberg Meetup.

Sie erreichen das Autorenteam auf Twitter unter [@angular_buch](https://twitter.com/angular_buch).

Mehr Infos und Kontaktmöglichkeiten finden Sie unter <https://angular-buch.com/autoren>.

Coypright und Urheberrechte:

Die durch die dpunkt.verlag GmbH vertriebenen digitalen Inhalte sind urheberrechtlich geschützt. Der Nutzer verpflichtet sich, die Urheberrechte anzuerkennen und einzuhalten. Es werden keine Urheber-, Nutzungs- und sonstigen Schutzrechte an den Inhalten auf den Nutzer übertragen. Der Nutzer ist nur berechtigt, den abgerufenen Inhalt zu eigenen Zwecken zu nutzen. Er ist nicht berechtigt, den Inhalt im Internet, in Intranets, in Extranets oder sonst wie Dritten zur Verwertung zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und eine gewerbliche Vervielfältigung der Inhalte wird ausdrücklich ausgeschlossen. Der Nutzer darf Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte im abgerufenen Inhalt nicht entfernen.

Ferdinand Malcher · Danny Koppenhagen · Johannes Hoppe

Angular

**Das große Praxisbuch –
Grundlagen, fortgeschrittene Themen
und Best Practices**

4., überarbeitete und aktualisierte Auflage



dpunkt.verlag



iX-Edition

In der iX-Edition erscheinen Titel, die vom dpunkt.verlag gemeinsam mit der Redaktion der Computerzeitschrift iX ausgewählt und konzipiert werden. Inhaltlicher Schwerpunkt dieser Reihe sind Software- und Webentwicklung sowie Administration.

Ferdinand Malcher · Danny Koppenhagen · Johannes Hoppe
team@angular-buch.com

Lektorat: René Schönfeldt
Projektkoordinierung: Anja Ehrlich
Copy-Editing: Annette Schwarz, Ditzingen
Satz: Da-TeX Gerd Blumenstein, Leipzig, www.da-tex.de
Herstellung: Stefanie Weidner
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck: Schleunungdruck GmbH, Marktheidenfeld
Bindung: Hubert & Co. GmbH & Co. KG. BuchPartner, Göttingen

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-86490-946-7

PDF 978-3-96910-862-8

ePub 978-3-96910-863-5

mobi 978-3-96910-864-2

4., überarbeitete und aktualisierte Auflage 2023
Copyright © 2023 dpunkt.verlag GmbH
Wiebinger Weg 17
69123 Heidelberg

Hinweis:

Der Umwelt zuliebe verzichten wir auf die Einschweißfolie.

Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: hallo@dpunkt.de.

Das Angular-Logo ist Eigentum von Google und ist frei verwendbar. Lizenz: Creative Commons BY 4.0

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autoren noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Inhaltsübersicht

I	Einführung	1
1	Schnellstart: unser erstes Angular-Projekt	3
2	Benötigte Werkzeuge: Editor, Node.js und Co.	11
3	Angular CLI: der Codegenerator für unser Projekt	19
II	TypeScript	25
4	Einführung in TypeScript	27
III	BookMonkey 5: Schritt für Schritt zur App	51
5	Projektvorstellung und Einrichtung	53
6	Komponenten: die Grundbausteine der Anwendung	73
7	Property Bindings: mit Komponenten kommunizieren	107
8	Event Bindings: Ereignisse in Komponenten verarbeiten ...	123
9	Power Tipp: Codeformatierung mit Prettier	137
10	NgModule: die Anwendung modularisieren	141
11	Dependency Injection: Code in Services auslagern	157
12	Routing: durch die Anwendung navigieren	175
13	Power Tipp: Chrome Developer Tools	213
14	HTTP-Kommunikation: ein Server-Backend anbinden	225
15	Reaktive Programmierung mit RxJS	243

16	Interceptors: HTTP-Requests abfangen und transformieren	301
17	Power Tipp: Analyse und Debugging mit den Angular DevTools	321
18	Formulare mit Template-Driven Forms	325
19	Formulare mit Reactive Forms	345
20	Formularvalidierung: die Eingaben prüfen	387
21	Pipes: Daten im Template formatieren	413
22	Direktiven: das Vokabular von HTML erweitern	433
23	Lazy Loading: Angular-Module asynchron laden	459
24	Guards: Routen absichern	471
25	Standalone Components: Komponenten ohne Module	485
IV Projektübergreifende Themen		507
26	Qualität fördern mit Softwaretests	509
27	Barrierefreiheit (a11y)	577
28	Lokalisierung (l10n)	599
29	Internationalisierung (i18n)	605
V Deployment: das Projekt ausliefern		629
30	Build und Deployment mit der Angular CLI	631
31	Angular-Anwendungen mit Docker bereitstellen	657

VI Fortgeschrittene Themen	677
32 State Management mit Redux und NgRx	679
33 Server-Side Rendering mit Angular Universal	741
34 Progressive Web Apps (PWA)	761
35 Fortgeschrittene Konzepte der Angular CLI	783
VII Wissenswertes	791
36 Fortgeschrittene Konzepte für Komponenten	793
37 Weitere Features des Routers	819
38 Nützliche Werkzeuge	827
39 Web Components mit Angular Elements	835
40 Angular Material und weitere UI-Komponenten- sammlungen	843
41 Angular updaten	847
42 Klassen-Propertys in JavaScript und TypeScript	851
VIII Anhang	857
A Befehle der Angular CLI	859
B Operatoren von RxJS	867
C Matchers von Jasmine	871
D Abkürzungsverzeichnis	875
E Linkliste	877
Index	887
Nachwort	897

Inhaltsverzeichnis

Vorwort **xxi**

Aktualisierungen in der vierten Auflage **xxix**

I Einführung **1**

1 Schnellstart: unser erstes Angular-Projekt **3**

2 Benötigte Werkzeuge: Editor, Node.js und Co. **11**

2.1 Konsole, Terminal und Shell 11

2.2 Visual Studio Code 11

2.3 Google Chrome 14

2.4 Paketverwaltung mit Node.js und NPM 14

2.5 Codebeispiele in diesem Buch 16

3 Angular CLI: der Codegenerator für unser Projekt **19**

3.1 Das offizielle Tool für Angular 19

3.2 Installation 20

3.3 Die wichtigsten Befehle 21

3.4 Autovervollständigung einrichten 22

II TypeScript **25**

4 Einführung in TypeScript **27**

4.1 TypeScript einsetzen 27

4.2 Variablen: const, let und var 29

4.3 Die wichtigsten Basistypen 31

4.4 Klassen 34

4.5 Interfaces 38

4.6 Template-Strings 39

4.7 Arrow-Funktionen/Lambda-Ausdrücke 40

4.8 Spread-Operator und Rest-Syntax 41

4.9 Weitere Features von JavaScript und TypeScript 44

4.10 Konfiguration 48

III	BookMonkey 5: Schritt für Schritt zur App	51
5	Projektvorstellung und Einrichtung	53
5.1	Unser Projekt: BookMonkey	53
5.2	Projekt mit der Angular CLI initialisieren	58
5.3	Aufbau des neuen Projekts	59
5.4	Das Projekt starten	65
5.5	Globale Styles einbinden: book-monkey5-styles	66
5.6	Statische Codeanalyse mit ESLint	68
6	Komponenten: die Grundbausteine der Anwendung	73
6.1	Komponenten	73
6.2	Komponenten in der Anwendung verwenden	78
6.3	Komponenten generieren mit der Angular CLI	79
6.4	Umgang mit Propertyts von Komponenten	80
6.5	Template-Syntax	82
6.6	Elemente gruppieren mit <code><ng-container></code>	89
6.7	Den BookMonkey erstellen: eine Buchliste anzeigen	94
7	Property Bindings: mit Komponenten kommunizieren	107
7.1	Komponenten verschachteln	107
7.2	Eingehender Datenfluss mit Property Bindings	108
7.3	Daten in Kindkomponenten verarbeiten	109
7.4	Property Bindings für native Elemente	110
7.5	Property Bindings notieren	111
7.6	Sonderformen von Property Bindings	113
7.7	Lifecycle Hooks von Komponenten	115
7.8	Den BookMonkey erweitern: Listeneinträge in eigener Komponente abbilden	117
8	Event Bindings: Ereignisse in Komponenten verarbeiten	123
8.1	Native DOM-Events	124
8.2	Eigene Events definieren	126
8.3	Den BookMonkey erweitern: Buchdetails anzeigen	128
9	Powertipp: Codeformatierung mit Prettier	137
10	NgModule: die Anwendung modularisieren	141
10.1	Module in Angular	141
10.2	Grundaufbau eines Moduls	142
10.3	Bestandteile eines Moduls deklarieren	143
10.4	Andere Module importieren	143
10.5	Bestandteile aus Modulen exportieren	145
10.6	Anwendung in Feature-Module aufteilen	146

10.7	Wiederverwendbarkeit: Shared Module	148
10.8	Den BookMonkey erweitern: die Anwendung modularisieren	149
11	Dependency Injection: Code in Services auslagern	157
11.1	Abhängigkeiten anfordern	159
11.2	Services in Angular	160
11.3	Abhängigkeiten registrieren	160
11.3.1	Abhängigkeiten explizit registrieren mit providers ...	160
11.3.2	Tree-Shakable Providers mit providedIn	162
11.4	Abhängigkeiten ersetzen	163
11.5	Eigene Tokens definieren mit InjectionToken	166
11.6	Abhängigkeiten anfordern mit @Inject()	167
11.7	Abhängigkeiten anfordern mit inject()	168
11.8	Multiprovider: mehrere Abhängigkeiten im selben Token	169
11.9	Providers in Komponenten registrieren	169
11.10	Den BookMonkey erweitern: einen Service nutzen	170
12	Routing: durch die Anwendung navigieren	175
12.1	Routen konfigurieren	176
12.2	Router einbinden: das AppRoutingModuleModule	177
12.3	Routing in Feature-Modulen	178
12.4	Komponenten anzeigen	181
12.5	Root-Route	182
12.6	Weiterleitung auf eine andere Route	182
12.7	Wildcard-Route	183
12.8	Links setzen	183
12.9	Routenparameter	185
12.10	Verschachtelung von Routen	188
12.11	Aktive Links stylen	191
12.12	Route programmatisch wechseln	192
12.13	Seitentitel setzen	193
12.14	Pfade in Single-Page-Applikationen	196
12.15	Den BookMonkey erweitern: Routing integrieren	197
13	Powertipp: Chrome Developer Tools	213
14	HTTP-Kommunikation: ein Server-Backend anbinden	225
14.1	Modul einbinden	226
14.2	Requests mit dem HttpClient durchführen	227
14.3	Optionen für den HttpClient	229
14.4	Ausblick: Codegenerierung mit OpenAPI	233
14.5	Den BookMonkey erweitern: Daten über HTTP abfragen	234

15	Reaktive Programmierung mit RxJS	243
15.1	Alles ist ein Datenstrom	243
15.2	Observables sind Funktionen	245
15.3	Das Observable aus RxJS	247
15.4	Observables abonnieren	249
15.5	Observables erzeugen	251
15.6	Observables und Promises	254
15.7	Operatoren: Datenströme modellieren	255
15.8	Heiße Observables, Multicasting und Subjects.....	259
15.9	Subscriptions verwalten & Memory Leaks vermeiden	266
15.10	Observables subscriben mit der AsyncPipe	270
15.11	Fehler behandeln	273
15.12	Flattening-Strategien für Higher-Order Observables	276
15.13	Den BookMonkey erweitern: Observables mit der AsyncPipe auflösen	282
15.14	Den BookMonkey erweitern: Typeahead-Suche	286
15.15	Den BookMonkey erweitern: Fehlerbehandlung	296
16	Interceptors: HTTP-Requests abfangen und transformieren	301
16.1	Funktionsweise der Interceptors	302
16.2	Interceptors anlegen	302
16.3	Den Request manipulieren	304
16.4	Die Response verarbeiten	304
16.5	Interceptors einbinden	305
16.6	Interceptors als Funktionen	307
16.7	OAuth 2 und OpenID Connect	307
16.8	Den BookMonkey erweitern: API-Aufrufe mit Credentials anreichern	310
17	Power Tipp: Analyse und Debugging mit den Angular DevTools	321
18	Formulare mit Template-Driven Forms	325
18.1	Angulars Ansätze für Formulare	325
18.2	Template-Driven Forms einrichten	326
18.3	Datenmodell in der Komponente	327
18.4	Template mit Two-Way Binding und ngModel	327
18.5	Eingaben validieren	328
18.6	Formularzustand verarbeiten	329
18.7	Formular abschicken	331
18.8	Formular zurücksetzen	331
18.9	Den BookMonkey erweitern: Template-Driven Forms nutzen .	333

19	Formulare mit Reactive Forms	345
19.1	Modul einbinden	346
19.2	Formularmodell in der Komponente	346
19.3	Template mit dem Modell verknüpfen	352
19.4	Eingebaute Validatoren nutzen	355
19.5	Formularzustand verarbeiten	357
19.6	Formular abschicken	358
19.7	Formular zurücksetzen	359
19.8	Formularwerte setzen	360
19.9	FormBuilder verwenden	361
19.10	Änderungen überwachen	363
19.11	Den BookMonkey erweitern: Reactive Forms nutzen	364
19.12	Den BookMonkey erweitern: Autor*innen erfassen	370
19.13	Den BookMonkey erweitern: Bücher bearbeiten	374
19.14	Welcher Ansatz ist der richtige?	384
20	Formularvalidierung: die Eingaben prüfen	387
20.1	Validatoren für einzelne Formularfelder	387
20.2	Validatoren für Formulargruppen und -Arrays	391
20.3	Validatoren kombinieren	393
20.4	Asynchrone Validatoren	394
20.5	Mit Fehlern arbeiten	397
20.6	Den BookMonkey erweitern: Felder für Autor*innen validieren	398
20.7	Den BookMonkey erweitern: ISBN-Format validieren	400
20.8	Den BookMonkey erweitern: existierende ISBN prüfen	402
20.9	Den BookMonkey erweitern: Fehlermeldungen anzeigen	405
21	Pipes: Daten im Template formatieren	413
21.1	Pipes verwenden	413
21.2	Eingebaute Pipes für den sofortigen Einsatz	414
21.3	Eigene Pipes entwickeln	424
21.4	Pipes in TypeScript nutzen	427
21.5	Den BookMonkey erweitern: Datum formatieren mit der DatePipe	428
21.6	Den BookMonkey erweitern: ISBN formatieren	429
22	Direktiven: das Vokabular von HTML erweitern	433
22.1	Was sind Direktiven?	433
22.2	Eigene Direktiven entwickeln	434

22.3	Attributdirektiven: Verhalten von Elementen ändern	436
22.3.1	Host Binding: Eigenschaften schreiben	437
22.3.2	Host Listener: Events abonnieren	439
22.3.3	Direktzugriff auf das Element mit <code>ElementRef</code>	440
22.3.4	Komponenten und Direktiven anfordern	441
22.4	Komposition mit Host-Direktiven	443
22.5	Strukturdirektiven: Elemente hinzufügen und entfernen	445
22.6	Den BookMonkey erweitern: Löschen mit Dialog bestätigen ..	449
22.7	Den BookMonkey erweitern: die Löschfunktion absichern ...	453
23	Lazy Loading: Angular-Module asynchron laden	459
23.1	Warum Module asynchron laden?	459
23.2	Das technische Konzept	460
23.3	Lazy Loading verwenden	461
23.4	Module asynchron vorladen: Preloading	464
23.5	Den BookMonkey erweitern: Module asynchron nachladen ..	465
24	Guards: Routen absichern	471
24.1	Grundlagen zu Guards	471
24.2	Guards verwenden	472
24.3	Guards implementieren	473
24.4	Guards als Klassen	478
24.5	Guards und Sicherheit	479
24.6	Den BookMonkey erweitern: die Admin-Route absichern	480
25	Standalone Components: Komponenten ohne Module ...	485
25.1	<code>NgModule</code> und Standalone Components	485
25.2	Standalone Components erzeugen	487
25.3	Abhängigkeiten definieren	487
25.4	Standalone Components in <code>NgModules</code> nutzen	489
25.5	Routing	490
25.6	Anwendungen ohne Module: <code>AppComponent</code> direkt bootstrappen	493
25.7	Projektstruktur	497
25.8	Fazit	498
25.9	Den BookMonkey erweitern: Pipes und Direktiven standalone verwenden	499
25.10	Den BookMonkey erweitern: Bücher-Feature mit Standalone Components	502

IV	Projektübergreifende Themen	507
26	Qualität fördern mit Softwaretests	509
26.1	Softwaretests	509
26.2	Vorgehen beim Testing	510
26.3	Test-Framework Jasmine	512
26.4	Test-Runner	517
26.4.1	Karma	517
26.4.2	Alternative: Jest	518
26.4.3	E2E-Test-Runner wählen	518
26.5	Unit- und Integrationstests mit Karma	519
26.5.1	TestBed: die Testbibliothek von Angular	519
26.5.2	Isolierte Unit-Tests: Services testen	521
26.5.3	Isolierte Unit-Tests: Pipes testen	523
26.5.4	Isolierte Unit-Tests: Komponenten testen	524
26.5.5	Shallow Component Test: einzelne Komponenten testen	528
26.5.6	Integrationstests: mehrere Komponenten testen	532
26.5.7	Abhängigkeiten durch Stubs ersetzen	534
26.5.8	Abhängigkeiten durch Mocks ersetzen	538
26.5.9	Leere Komponenten als Stubs oder Mocks einsetzen	541
26.5.10	HTTP-Requests testen	542
26.5.11	Komponenten mit Routen testen	546
26.5.12	Asynchronen Code testen	550
26.5.13	Code Coverage Report	556
26.5.14	Zusammenfassung: Tests mit Karma und Jasmine	557
26.6	Jest: ein alternativer Test-Runner mit zusätzlichen Features	558
26.7	Oberflächentests mit Cypress	562
26.8	Component Tests mit Cypress: Komponenten isoliert testen	572
27	Barrierefreiheit (a11y)	577
27.1	Gesetze und Standards	579
27.2	Features von Angular	584
27.3	ESLint-Regeln	588
27.4	Angular Component Development Kit (CDK)	589
27.5	Verifizierung & Tools zur Unterstützung	595
28	Lokalisierung (l10n)	599
28.1	Lokalisierung für ein spezifisches Locale	600
28.2	Mehrere Sprachdefinitionen laden	601
28.3	Pipes mit einem spezifischen Locale nutzen	602

29	Internationalisierung (i18n)	605
29.1	Was bedeutet Internationalisierung?	605
29.2	Der Übersetzungsprozess in Angular	606
29.3	Texte für die Übersetzung markieren und extrahieren	607
29.3.1	Projekt vorbereiten	607
29.3.2	Nachrichten im HTML mit dem i18n-Attribut markieren	608
29.3.3	Nachrichten im TypeScript-Code mit \$localize markieren	609
29.3.4	Feste IDs vergeben	610
29.3.5	Nachrichten extrahieren und übersetzen	611
29.4	Übersetzung während des Build-Prozesses	613
29.5	Übersetzung zur Laufzeit	619
29.6	Technische Einschränkungen	627

V Deployment: das Projekt ausliefern 629

30	Build und Deployment mit der Angular CLI	631
30.1	Build konfigurieren (angular.json)	631
30.2	Build ausführen	635
30.3	Bundles	636
30.3.1	Weitere Bundles und Dateien	637
30.3.2	Budgets konfigurieren	638
30.3.3	Bundles analysieren mit source-map-explorer	639
30.4	Umgebungen konfigurieren	640
30.5	Ahead-of-Time-Kompilierung: die Templates umsetzen	646
30.6	Webserver konfigurieren und die Anwendung ausliefern	649
30.7	ng deploy: Deployment mit der Angular CLI	652
30.8	Ausblick: Deployment mit einem Build-Service	654
31	Angular-Anwendungen mit Docker bereitstellen	657
31.1	Docker	658
31.2	Docker Registry	659
31.3	Lösungsskizze	659
31.4	Eine Angular-App über Docker bereitstellen	660
31.5	Build Once, Run Anywhere: Konfiguration über Docker verwalten	664
31.6	Multi-Stage Builds	670
31.7	Grenzen der vorgestellten Lösung	674
31.8	Fazit	675

VI	Fortgeschrittene Themen	677
32	State Management mit Redux und NgRx	679
32.1	Ein Modell für zentrales State Management	680
32.2	Das Architekturmodell Redux	691
32.3	NgRx: Reactive Extensions for Angular	693
32.3.1	Projekt vorbereiten	694
32.3.2	Store einrichten	694
32.3.3	Schematics nutzen	694
32.3.4	Grundstruktur	695
32.3.5	Feature anlegen	696
32.3.6	Struktur des Feature-States definieren	698
32.3.7	Actions: Kommunikation mit dem Store	699
32.3.8	Dispatch: Actions in den Store senden	701
32.3.9	Reducers: den State aktualisieren	702
32.3.10	Selektoren: Daten aus dem State lesen	706
32.3.11	Effects: Seiteneffekte ausführen	711
32.4	Debugging mit den Redux DevTools	717
32.5	Redux und NgRx: Wie geht's weiter?	720
32.5.1	Actions gruppieren mit createActionGroup()	720
32.5.2	Routing	721
32.5.3	Entity Management	721
32.5.4	Testing	724
32.5.5	Hilfsmittel für Komponenten: @ngrx/component	733
32.5.6	Facades: Zustandsverwaltung abstrahieren	735
32.6	Ausblick: lokaler State mit @ngrx/component-store	738
33	Server-Side Rendering mit Angular Universal	741
33.1	Single-Page-Anwendungen, Suchmaschinen und Start-Performance	742
33.2	Dynamisches Server-Side Rendering	745
33.3	Statisches Pre-Rendering	750
33.4	Hinter den Kulissen von Angular Universal	753
33.5	Browser oder Server? Die Plattform bestimmen	754
33.6	Routen ausschließen	755
33.7	Wann setze ich serverseitiges Rendering ein?	757
33.8	Ausblick: Pre-Rendering mit Scully	758
34	Progressive Web Apps (PWA)	761
34.1	Die Charakteristiken einer PWA	761
34.2	Service Worker	762
34.3	Eine bestehende Angular-Anwendung in eine PWA verwandeln	763

34.4	Add to Homescreen	765
34.5	Offline-Funktionalität	768
34.6	Push-Benachrichtigungen	773

35 Fortgeschrittene Konzepte der Angular CLI 783

35.1	Workspace und Monorepo: Heimat für Apps und Bibliotheken	783
35.1.1	Applikationen: Angular-Apps im Workspace	784
35.1.2	Bibliotheken: Code zwischen Anwendungen teilen ..	786
35.2	Schematics: Codegenerierung mit der Angular CLI	788

VII Wissenswertes

791

36 Fortgeschrittene Konzepte für Komponenten 793

36.1	Else-Block für die Direktive ngIf	793
36.2	TrackBy-Funktion für die Direktive ngFor	794
36.3	Container und Presentational Components	796
36.4	Content Projection: Inhalt des Host-Elements verwenden	800
36.5	Lifecycle Hooks	802
36.6	Change Detection	805

37 Weitere Features des Routers 819

37.1	Auxiliary Routes: mehrere RouterOutlets verwenden	819
37.2	Erweiterte Konfigurationen für den Router	820
37.3	Resolvers: Daten beim Routing vorladen	822

38 Nützliche Werkzeuge 827

38.1	Monorepos mit Nrwl Nx	827
38.2	Angular-Anwendungen dokumentieren und visualisieren	830
38.2.1	Compodoc	831
38.2.2	Storybook	832

39 Web Components mit Angular Elements 835

40 Angular Material und weitere UI-Komponenten-sammlungen 843

41 Angular updaten 847

42 Klassen-Propertyts in JavaScript und TypeScript 851

VIII	Anhang	857
A	Befehle der Angular CLI	859
B	Operatoren von RxJS	867
C	Matchers von Jasmine	871
D	Abkürzungsverzeichnis	875
E	Linkliste	877
	Index	887
	Nachwort.....	897

Vorwort

»Angular is one of the most adopted frameworks on the planet.«

Brad Green
(chem. Angular Engineering Director)

Angular ist eines der populärsten Frameworks für die Entwicklung von Single-Page-Applikationen. Das Framework wird weltweit von großen Unternehmen eingesetzt, um modulare, skalierbare und gut wartbare Applikationen zu entwickeln. Mit Angular in Version 2.0.0 setzte Google im Jahr 2016 einen Meilenstein in der Welt der modernen Webentwicklung: Das Framework nutzt die Programmiersprache TypeScript, bietet ein ausgereiftes Tooling und ermöglicht die komponentenbasierte Entwicklung von Single-Page-Anwendungen für den Browser und für Mobilgeräte.

In kurzer Zeit haben sich rund um Angular ein umfangreiches Ökosystem und eine vielfältige Community gebildet. Angular gilt neben React.js und Vue.js als eines der weltweit beliebtesten Webframeworks. Sie haben also die richtige Entscheidung getroffen, als Sie Angular für die Entwicklung Ihrer Projekte ins Auge gefasst haben.

React und Vue.js

Der Einstieg in Angular ist umfangreich, aber die Konzepte sind durchdacht und konsequent. Häufig verwendet man im Zusammenhang mit Angular das Attribut *opinionated*, das wir im Deutschen mit dem Begriff *meinungsstark* ausdrücken können: Angular ist ein meinungsstarkes Framework, das viele klare Richtlinien zu Architektur, Codestruktur und Best Practices definiert. Das kann zu Anfang umfangreich erscheinen, sorgt aber dafür, dass in der gesamten Community einheitliche Konventionen herrschen, Standardlösungen existieren und bestehende Bibliotheken vorausgewählt wurden.

*Opinionated
Framework*

Sie werden in diesem Buch lernen, wie Sie mit Angular komponentenbasierte Single-Page-Applikationen erstellen. Dazu entwickeln wir mit Ihnen gemeinsam eine Anwendung, anhand derer wir Ihnen die Konzepte und Features von Angular beibringen. Wir führen Sie Schritt für Schritt durch das Framework – vom Projektsetup über Komponenten, Routing, Formulare und HTTP bis hin zum Testing und Deployment der Anwendung. Auf dem Weg stellen wir Ihnen eine

Beispielanwendung

Reihe von Tools, Tipps und Best Practices vor, die wir in mehr als sechs Jahren Praxisalltag mit Angular sammeln konnten. Die umfangreichen Theorieteile eignen sich auch später als Nachschlagewerk im Entwicklungsalltag.

Nach dem Lesen dieses Praxisbuchs sind Sie in der Lage,

- das Zusammenspiel der Funktionen von Angular sowie das Konzept hinter dem Framework zu verstehen,
- modulare, strukturierte und wartbare Webanwendungen mithilfe von Angular zu entwickeln sowie
- durch die Entwicklung von Tests qualitativ hochwertige Anwendungen zu erstellen.

Die Entwicklung mit Angular macht vor allem eines: *Spaß!* Diesen Enthusiasmus für das Framework und für Webtechnologien möchten wir Ihnen in diesem Buch vermitteln – wir nehmen Sie mit auf die Reise in die Welt der modernen Webentwicklung!

Versionen und Namenskonvention: Angular vs. AngularJS

In diesem Buch dreht sich alles um das Framework Angular. Die Geschichte dieses Projekts reicht zurück bis ins Jahr 2009 zur Vorgängerversion *AngularJS*. Bis auf den ähnlichen Namen und einige Konzepte haben die beiden Frameworks aber nichts miteinander zu tun: Angular ab Version 2 ist eine vollständige Neuentwicklung und ist nicht mit dem alten AngularJS kompatibel.

It's just »Angular«.

Die offizielle Bezeichnung für das Framework ist *Angular*, ohne Angabe der Programmiersprache und ohne eine spezifische Versionsnummer. Angular erschien im September 2016 in der Version 2.0.0 und hat viele neue Konzepte und Ideen in die Community gebracht. Um Verwechslungen auszuschließen, gilt also die folgende Konvention:

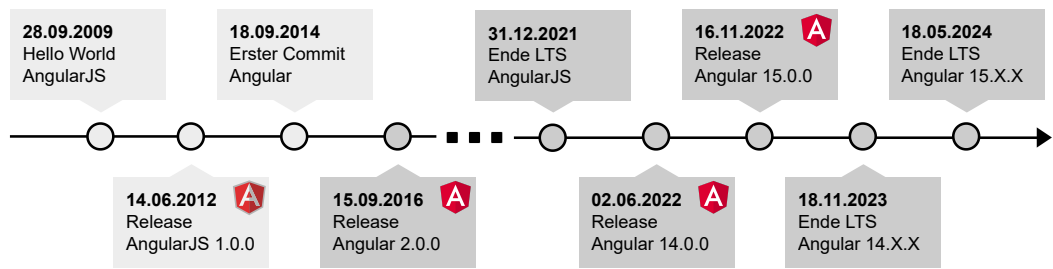
- **Angular** – das Angular-Framework ab **Version 2 und höher** (dieses Buch ist durchgängig auf dem Stand von Angular 15)
- **AngularJS** – das Angular-Framework in der **Version 1.x.x**

AngularJS wird seit Januar 2022 offiziell nicht mehr weiterentwickelt und sollte nicht mehr verwendet werden.¹ Sie haben also die richtige Entscheidung getroffen, Angular ab Version 2.0.0 einzusetzen.

¹<https://ng-buch.de/c/1> – AngularJS: Version Support Status

Die Versionsnummer *x.y.z* basiert auf *Semantic Versioning*.² Der Release-Zyklus von Angular ist kontinuierlich geplant: Im Rhythmus von ungefähr sechs Monaten erscheint eine neue Major-Version *x*. Die Minor-Versionen *y* werden monatlich herausgegeben, nachdem eine Major-Version erschienen ist. Jede Major-Version wird planmäßig für 1,5 Jahre unterstützt und weiterentwickelt (Long-Term Support).

Semantic Versioning



Umgang mit Aktualisierungen

Das Release einer neuen Major-Version von Angular bedeutet keineswegs, dass alle Ideen verworfen werden und Ihre Software nach einem Update nicht mehr funktioniert. Auch wenn Sie eine neuere Angular-Version verwenden, behalten die in diesem Buch beschriebenen Konzepte ihre Gültigkeit. Die Grundideen von Angular sind seit Version 2.0.0 konsistent und auf Beständigkeit über einen langen Zeitraum ausgelegt. Alle Updates zwischen den Major-Versionen waren in der Vergangenheit problemlos möglich, ohne dass Breaking Changes die gesamte Anwendung unbenutzbar machen. Gibt es doch gravierende Änderungen, so werden stets ausführliche Informationen und Tools zur Migration angeboten.

Auf der Website zu diesem Buch finden Sie die Codebeispiele für das Beispielpjekt und viele weiterführende Informationen. Unter anderem veröffentlichen wir dort zu jeder Major-Version einen Artikel mit den wichtigsten Neuerungen in Angular. Wir empfehlen Ihnen aus diesem Grund, unbedingt einen Blick auf die Begleitwebsite zu werfen, bevor Sie beginnen, sich mit den Inhalten des Buchs zu beschäftigen:

Abb. 1
*Zeitleiste der
Entwicklung von
Angular*



*Die Begleitwebsite
zum Buch*

<https://angular-buch.com>

²<https://ng-buch.de/c/2> – Semantic Versioning 2.0.0

An wen richtet sich das Buch?

*Erfahrung in
Softwareentwicklung*

Dieses Buch richtet sich an Menschen, die bereits grundlegende Kenntnisse in der Softwareentwicklung mitbringen. Vorwissen zu JavaScript und HTML ist von Vorteil – es ist aber keine Voraussetzung, um mit diesem Buch Angular zu lernen. Wenn Sie jedoch bereits mit der Webentwicklung vertraut sind, werden Sie mit diesem Buch schnell starten können. Falls Sie gar keine Erfahrung in HTML und JavaScript mitbringen, empfehlen wir Ihnen, zunächst die grundlegenden Kenntnisse in diesen Bereichen zu festigen.

TypeScript

Für die Entwicklung mit Angular nutzen wir die populäre Programmiersprache TypeScript. Doch keine Angst: TypeScript ist eine Erweiterung von JavaScript, und die Konzepte sind sehr eingängig und schnell gelernt. Wenn Sie bereits eine stark typisierte Sprache wie Java oder C# kennen, wird Ihnen der Einstieg in TypeScript nicht schwerfallen.

*Keine Angular-
Vorkenntnisse nötig!*

Sie benötigen *keinerlei* Vorkenntnisse im Umgang mit Angular bzw. AngularJS. Ebenso müssen Sie sich nicht vorab mit benötigten Tools und Hilfsmitteln für die Entwicklung von Angular-Applikationen vertraut machen. Das nötige Wissen darüber wird Ihnen in diesem Buch vermittelt.

*Kein klassisches
Nachschlagewerk*

Wir erschließen uns die Welt von Angular praxisorientiert anhand eines Beispielprojekts. Jedes Thema wird zunächst ausführlich in der Theorie behandelt, sodass Sie die Grundlagen auch losgelöst vom Beispielprojekt nachlesen können. Wir wollen einen soliden Einstieg in Angular bieten, Best Practices zeigen und Schwerpunkte bei speziellen fortgeschrittenen Themen setzen. Die meisten Aufgaben aus dem Entwicklungsalltag werden Sie mit den vielen praktischen Beispielen souverän meistern können.

*Offizielle Angular-
Dokumentation*

Wir hoffen, dass dieses Buch Ihre tägliche Begleitung bei der Arbeit mit Angular wird. Für Details zu den einzelnen Framework-Funktionen empfehlen wir immer auch einen Blick in die offizielle Dokumentation.³

Wie ist dieses Buch zu lesen?

*Einführung, Tools und
Schnellstart*

Im ersten Teil des Buchs lernen Sie die verwendeten Tools und die benötigten Werkzeuge kennen. Im Schnellstart tauchen wir sofort in Angular ein und nehmen Sie mit zu einem einfachen Einstieg in das Framework und den Grundaufbau einer Anwendung.

*Einführung in
TypeScript*

Der zweite Teil vermittelt Ihnen einen Einstieg in TypeScript. Sie werden hier mit den Grundlagen dieser typisierten Skriptsprache vertraut gemacht. Wenn Sie bereits Erfahrung im Umgang mit TypeScript

³<https://ng-buch.de/c/3> – Angular Docs

haben, können Sie diesen Teil auch überspringen und bei Bedarf später einzelne Themen nachlesen.

Der dritte Teil ist der Hauptteil des Buchs: Hier werden wir mit Ihnen zusammen eine Beispielanwendung entwickeln. Die Konzepte und Technologien von Angular wollen wir dabei direkt am Beispiel vermitteln. Wir haben das Projekt in 17 einzelne Kapitel eingeteilt. In jedem Teil setzen wir gemeinsam mit Ihnen neue Anforderungen und technische Aspekte im Beispielpunkt um.

Beispielanwendung

17 Praxiskapitel

- Komponenten: die Grundbausteine der Anwendung (ab S. 73)
- Property Bindings: mit Komponenten kommunizieren (ab S. 107)
- Event Bindings: Ereignisse in Komponenten verarbeiten (ab S. 123)
- NgModule: die Anwendung modularisieren (ab S. 141)
- Dependency Injection: Code in Services auslagern (ab S. 157)
- Routing: durch die Anwendung navigieren (ab S. 175)
- HTTP-Kommunikation: ein Server-Backend anbinden (ab S. 225)
- Reaktive Programmierung mit RxJS (ab S. 243)
- Interceptors: HTTP-Requests abfangen und transformieren (ab S. 301)
- Formulare mit Template-Driven Forms (ab S. 325)
- Formulare mit Reactive Forms (ab S. 345)
- Formularvalidierung: die Eingaben prüfen (ab S. 387)
- Pipes: Daten im Template formatieren (ab S. 413)
- Direktiven: das Vokabular von HTML erweitern (ab S. 433)
- Lazy Loading: Angular-Module asynchron laden (ab S. 459)
- Guards: Routen absichern (ab S. 471)
- Standalone Components: Komponenten ohne Module (ab S. 485)

Jedes dieser Kapitel besteht immer aus einem umfangreichen Theorieteil und der praktischen Implementierung im Beispielpunkt. Neben fachlichen Themen führen wir Refactorings durch, die die Architektur oder den Codestil der Anwendung verbessern. In mehreren *Powertipps* zwischen den Kapiteln zeigen wir außerdem hilfreiche Werkzeuge, die uns bei der Entwicklung zur Seite stehen.

Powertipps

Nachdem alle Praxiskapitel erfolgreich absolviert wurden, widmen wir uns einer Auswahl von projektübergreifenden Themen: Im Kapitel zu *Softwaretests* erfahren Sie, wie Sie Ihre Angular-Anwendung automatisiert testen und so die Softwarequalität sichern können. Dieses Kapitel kann sowohl nach der Entwicklung des Beispielpunkts als auch parallel dazu bestritten werden. Außerdem widmen wir uns ausführlich der *Barrierefreiheit*: In diesem Kapitel stellen wir Grundlagen und konkrete Maßnahmen vor, um die Anwendung für möglichst viele Menschen zugänglich zu machen. Zum Schluss werfen wir einen differen-

Projektübergreifende Themen

zierten Blick auf die *Lokalisierung* und *Internationalisierung*, um die Anwendung für den mehrsprachigen Betrieb vorzubereiten.

Deployment

Im fünften Teil des Buchs dreht sich alles um das Deployment einer Angular-Anwendung. Dabei betrachten wir die Hintergründe und Konfiguration des Build-Prozesses und erläutern die Bereitstellung mithilfe von Docker.

Fortgeschrittene Themen: NgRx, SSR und PWA

Im sechsten Teil möchten wir Ihnen einige Ansätze näherbringen, die über eine Standardanwendung hinausgehen. Hier stellen wir zunächst das *Redux*-Pattern und das populäre Framework *NgRx* vor, mit dem wir den Anwendungszustand zentral und gut wartbar verwalten können. Mit *Server-Side Rendering (SSR)* machen Sie Ihre Anwendung fit für Suchmaschinen und verbessern zusätzlich die wahrgenommene Geschwindigkeit beim initialen Start. Außerdem werfen wir einen ausführlichen Blick auf *Progressive Web Apps (PWA)*, um eine Webanwendung elegant auf Mobilgeräten zu nutzen.

Wissenswertes

Im letzten Teil »Wissenswertes« finden Sie weitere Informationen zu wissenswerten und begleitenden Themen. Hier haben wir weiterführende Inhalte zusammengetragen, auf die wir im Beispielpjekt nicht ausführlich eingehen.

Selbst tippen statt Copy & Paste

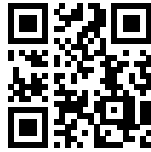
Abtippen heißt Lernen und Verstehen.

Der gesamte Code für das Beispielpjekt steht auf der Plattform GitHub zur Verfügung. Wir wissen genau, wie groß die Versuchung ist, größere Teile des Codes von dort zu kopieren und so die Tipparbeit zu sparen. Aber: Kopieren und Einfügen ist nicht dasselbe wie *Lernen* und *Verstehen*. Wenn Sie die Codebeispiele selbst *tippen*, werden Sie besser verstehen, wie Angular funktioniert, und werden die Software später erfolgreich in der Praxis einsetzen können. Jeder einzelne Quelltext, den Sie abtippen, trainiert Ihre Hände, Ihr Gehirn und Ihre Sinne. Wir möchten Sie deshalb ermutigen: Betrügen Sie sich nicht selbst. Der bereitgestellte Quelltext im Repository sollte lediglich der Überprüfung dienen. Wir wissen, wie schwer das ist, aber vertrauen Sie uns: Es zahlt sich aus, denn Übung macht den Meister!

Angular.Schule: Workshops und Beratung

Wir, die Autoren dieses Buchs, arbeiten seit Langem als Berater und Trainer für Angular. Wir haben die Erfahrung gemacht, dass man Angular in kleinen Gruppen am effektivsten lernen kann. In einem Workshop kann auf individuelle Fragen und Probleme direkt eingegangen werden – und es macht auch am meisten Spaß!

Schauen Sie auf <https://angular.schule> vorbei. Dort bieten wir Ihnen Angular-Schulungen in den Räumen Ihres Unternehmens, in offenen Gruppen oder als Online-Kurs an. Das Angular-Buch verwenden wir dabei in unseren Kursen zur Nacharbeit. Wir freuen uns auf Ihren Besuch!



<https://angular.schule>

*Die Angular.Schule:
Workshops und
Beratung*

Danksagung

Dieses Buch hätte nicht seine Reife erreicht ohne die Hilfe und Unterstützung verschiedener Menschen. Besonderer Dank geht an **Michael Kaaden** für seine unermüdlichen Anregungen, kritischen Nachfragen und die Geduld, unser Beispielpjekt zum vierten Mal durchzuarbeiten.

Wir bedanken uns bei **Mohammed Malekzadeh** und **Maximilian Franzke** für wertvolles Feedback und viele Anregungen zum Thema digitale Barrierefreiheit. **Jan Buchholz** danken wir für die hilfreichen Korrekturvorschläge. Wir danken **Lisa Möller** für die Zeichnungen zu unseren Personas (Seite 56). Ein großer Dank gilt außerdem unseren Familien, die uns auch abends an diesem Buch haben arbeiten lassen!

Dem Team vom dpunkt.verlag, insbesondere **Anja Ehrlich** und **René Schönfeldt**, danken wir für die Unterstützung und die Anregungen zum Buch. **Annette Schwarz** danken wir für das gewissenhafte Korrekturat unseres Manuskripts. Besonderer Dank gilt dem **Angular-Team** und der Community dafür, dass sie eine großartige Plattform geschaffen haben, die uns den Entwicklungsalltag angenehmer macht. Zuletzt danken wir **Gregor Woiwode** für die Mitwirkung als Autor in der ersten Auflage. Ohne Gregor würde es das Angular-Buch in dieser Form nicht geben.

Viele Menschen haben uns E-Mails mit persönlichem Feedback zum Buch zukommen lassen – vielen Dank für diese wertvollen Rückmeldungen.

Aktualisierungen in der vierten Auflage

Die Webplattform bewegt sich schnell, und so muss auch ein Framework wie Angular stets an neue Gegebenheiten angepasst werden und mit den Anforderungen wachsen. In den sechs Jahren seit Veröffentlichung der ersten Auflage dieses Buchs haben sich viele Dinge geändert: Es wurden Best Practices etabliert, neue Features eingeführt, und einige wenige Features wurden wieder entfernt.

Mit dieser Auflage haben wir die bislang größte und aufwendigste Überarbeitung gewagt: Wir haben uns mehr als ein Jahr Zeit genommen, um das Konzept dieses Buchs zu überdenken und auch Ideen und Baustellen zu bearbeiten, auf die wir bei den früheren Auflagen nicht den Fokus gesetzt hatten.

Alle Texte und Beispiele haben wir grundlegend überarbeitet und zum großen Teil neu verfasst. Dabei haben wir uns auch nicht davor gescheut, ganze Abschnitte zu löschen oder unsere Ideen aus der Vergangenheit kritisch zu hinterfragen.

Es ist unser Ziel, einen umfassenden Einstieg in das Angular-Framework zu ermöglichen – und gleichzeitig ein modernes und zeitloses Nachschlagewerk zu schaffen. Die Arbeit mit dem Buch dieser vierten Auflage lohnt sich daher auch für Leserinnen und Leser, die bereits eine der früheren Ausgaben besitzen. Diesen »frischen Wind der Veränderung« haben wir auch auf dem Buchcover mit einem Motiv aus der Raumfahrt aufgegriffen.

*Modernes und zeitloses
Nachschlagewerk*

Wir möchten Ihnen einen kurzen Überblick über die wichtigsten Neuerungen und Aktualisierungen der vierten Auflage geben. Alle Inhalte haben wir auf die Angular-Version 15 aktualisiert, sodass dieses Buch auch für die Arbeit mit den folgenden Versionen geeignet ist.

*Angular 15 und
folgende Versionen*

Neu in dieser Auflage

Wir haben das Buch neu strukturiert und das Beispielprojekt von Grund auf neu entwickelt. Dabei haben wir die Schwerpunkte anders gesetzt und die Erkenntnisse aus unserer täglichen Arbeit mit Angular