



3.

Auflage



Ferdinand Malcher · Johannes Hoppe · Danny Koppenhagen

Angular

Grundlagen, fortgeschrittene Themen
und Best Practices

inkl. RxJS,
NgRx und
PWA

 EDITION

dpunkt.verlag

Liebe Leserin, lieber Leser,

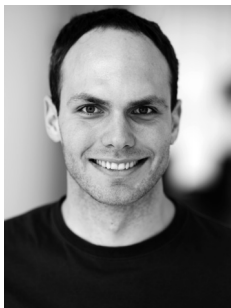
das Angular-Ökosystem wird kontinuierlich verbessert. Bitte haben Sie Verständnis dafür, dass sich seit dem Druck dieses Buchs unter Umständen Schnittstellen und Aspekte von Angular weiterentwickelt haben können. Die GitHub-Repositorys mit den Codebeispielen werden wir bei Bedarf entsprechend aktualisieren.

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 hierfür unter team@angular-buch.com mit einer Beschreibung des Problems.

Wir wünschen Ihnen viel Spaß mit Angular!

Alles Gute
Ferdinand, Johannes und Danny



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

🐦 [@fmalcher01](https://twitter.com/fmalcher01)



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

🐦 [@JohannesHoppe](https://twitter.com/JohannesHoppe)



Danny Koppenhagen arbeitet als Softwareentwickler und Berater für Enterprise-Webanwendungen. Sein Schwerpunkt liegt in der Entwicklung von nutzerzentrierten Anwendungen mit TypeScript und Angular sowie JavaScript und Vue.js. Neben der beruflichen Tätigkeit ist Danny als Autor mehrerer Open-Source-Projekte aktiv.

🐦 [@d_koppenhagen](https://twitter.com/d_koppenhagen)

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

Ferdinand Malcher · Johannes Hoppe · Danny Koppenhagen

Angular

**Grundlagen, fortgeschrittene Themen
und Best Practices –
inkl. RxJS, NgRx und PWA**

3., aktualisierte und erweiterte Auflage

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

Lektorat: René Schönfeldt
Lektoratsassistentin und Projektkoordinierung: Julia Griebel
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 und Bindung: mediaprint solutions GmbH, 33100 Paderborn

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 978-3-86490-779-1
PDF 978-3-96910-081-3
ePub 978-3-96910-082-0
mobi 978-3-96910-083-7

3., aktualisierte und erweiterte Auflage 2020
Copyright © 2020 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 Heidelberg

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich 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 Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

»*It's just ›Angular‹.*«
Igor Minar

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. Tatsächlich hat Angular seinen Ursprung beim wohl größten Player des Internets – Google. Obwohl kommerzielle Absichten hinter der Idee stehen, wurde Angular von Anfang an quelloffen unter der MIT-Lizenz veröffentlicht. Im September 2016 erschien Angular in der Version 2.0.0. Google setzte damit einen Meilenstein in der Welt der modernen Webentwicklung: Das Framework nutzt die Programmiersprache TypeScript, bietet ein ausgereiftes Tooling und komponentenbasierte Entwicklung. In kurzer Zeit haben sich rund um Angular ein umfangreiches Ökosystem und eine vielfältige Community gebildet.

Die Entwicklung wird maßgeblich von einem dedizierten Team bei Google vorangetrieben, wird aber auch stark aus der Community beeinflusst. Angular gilt neben React.js (Facebook) und Vue.js (Community-Projekt) als eines der weltweit beliebtesten Webframeworks. Sie haben also die richtige Entscheidung getroffen und haben Angular für die Entwicklung Ihrer Projekte ins Auge gefasst.

Das Framework ist modular aufgebaut und stellt eine Vielzahl an Funktionalitäten bereit, um wiederkehrende Standardaufgaben zu lösen. Der Einstieg ist umfangreich, aber die Konzepte sind durchdacht und konsequent. Hat man die Grundlagen erlernt, so kann man den Fokus auf die eigentliche Businesslogik legen. 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 gesam-

*Opinionated
Framework*

ten Community einheitliche Konventionen herrschen, Standardlösungen existieren und bestehende Bibliotheken vorausgewählt wurden.

Obwohl die hauptsächliche Zielplattform für Angular-Anwendungen der Browser ist, ist das Framework nicht darauf festgelegt: Durch seine Plattformunabhängigkeit kann Angular auf nahezu jeder Plattform ausgeführt werden, unter anderem auf dem Server und nativ auf Mobilgeräten.

*Grundlegende
Konzepte*

Sie werden in diesem Buch lernen, wie Sie mit Angular komponentenbasierte Single-Page-Applikationen entwickeln. Wir werden Ihnen vermitteln, wie Sie Abhängigkeiten und Asynchronität mithilfe des Frameworks behandeln. Weiterhin erfahren Sie, wie Sie mit Routing die Navigation zwischen verschiedenen Teilen der Anwendung implementieren. Sie werden lernen, wie Sie komplexe Formulare mit Validierungen in Ihre Anwendung integrieren und wie Sie Daten aus einer HTTP-Schnittstelle konsumieren können.

Beispielanwendung

Wir entwickeln mit Ihnen gemeinsam eine Anwendung, anhand derer wir Ihnen all diese Konzepte von Angular beibringen. Dabei führen wir Sie Schritt für Schritt durch das Projekt – vom Projektsetup über das Testen des Anwendungscodes bis zum Deployment der fertig entwickelten Anwendung. Auf dem Weg stellen wir Ihnen eine Reihe von Tools, Tipps und Best Practices vor, die wir in mehr als vier Jahren Praxisalltag mit Angular sammeln konnten.

Nach dem Lesen des Buchs 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 von 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. Sucht man nach dem Begriff »Angular« im Internet, so stößt man auch oft noch auf die Bezeichnung »AngularJS«. Hinter dieser Bezeichnung verbirgt sich die Version 1 des Frameworks. Mit der Version 2 wurde Angular von Grund auf neu entwickelt. Die offizielle Bezeichnung für das neue 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. Weil es sich um eine vollständige Neuentwicklung handelt, ist Angular nicht ohne Weiteres mit dem alten AngularJS kompatibel. Um Verwechslungen auszuschließen, gilt also die folgende Konvention:

It's just »Angular«.

- **Angular** – das Angular-Framework ab **Version 2 und höher** (dieses Buch ist durchgängig auf dem Stand von Angular 10)
- **AngularJS** – das Angular-Framework in der **Version 1.x.x**

AngularJS, das 2010 erschien, ist zwar mittlerweile etwas in die Jahre gekommen, viele Webanwendungen setzen aber weiterhin auf das Framework. Die letzte Version 1.8.0 wurde im Juni 2020 veröffentlicht und wird ab Januar 2022 offiziell nicht mehr weiterentwickelt.¹

Long Term Support für AngularJS

Sie haben also die richtige Entscheidung getroffen, Angular ab Version 2.0.0 einzusetzen. Diese Versionsnummer *x.y.z* basiert auf *Semantic Versioning*.² Der Release-Zyklus von Angular ist kontinuierlich geplant: Im Rhythmus von durchschnittlich sechs Monaten erscheint eine neue Major-Version *x*. Die Minor-Versionen *y* werden monatlich herausgegeben, nachdem eine Major-Version erschienen ist.

Semantic Versioning

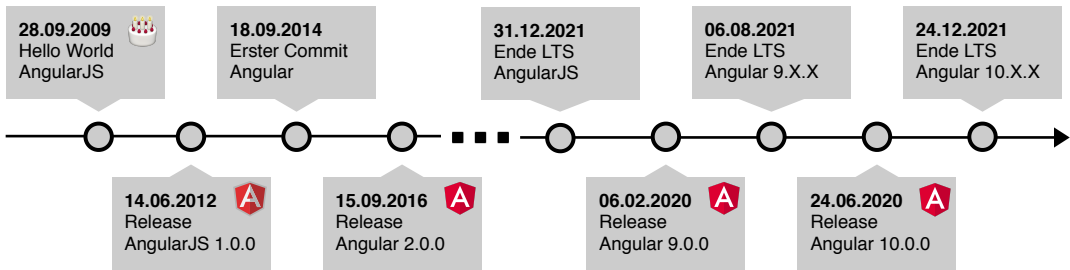


Abb. 1
Zeitleiste der Entwicklung von Angular

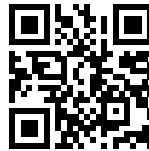
¹ <https://ng-buch.de/b/1> – AngularJS: Version Support Status
² <https://ng-buch.de/b/2> – Semantic Versioning 2.0.0

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 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.

Alle Beispiele aus diesem Buch sowie zusätzliche Links und Hinweise können Sie über eine zentrale Seite erreichen:

*Die Begleitwebsite
zum Buch*



<https://angular-buch.com>

Unter anderem veröffentlichen wir dort zu jeder Major-Version einen Artikel mit den wichtigsten Neuerungen und den nötigen Änderungen am Beispielpunkt. Wir empfehlen Ihnen aus diesem Grund, unbedingt einen Blick auf die Begleitwebsite des Buchs zu werfen, bevor Sie beginnen, sich mit den Inhalten des Buchs zu beschäftigen.

An wen richtet sich das Buch?

*Webentwickler mit
JavaScript-Erfahrung*

Dieses Buch richtet sich an Webentwickler, die einige Grundkenntnisse mitbringen. Wir setzen allgemeine Kenntnisse in JavaScript voraus. Wenn Sie bereits ein erstes JavaScript-Projekt umgesetzt haben und Ihnen Frameworks wie jQuery vertraut sind, werden Sie an diesem Buch sehr viel Freude haben. Mit Angular erwartet Sie das modulare Entwickeln von Single-Page-Applikationen in Kombination mit Unit- und UI-Testing.

*TypeScript-Einsteiger
und Erfahrene*

Für die Entwicklung mit Angular nutzen wir die populäre Programmiersprache TypeScript. Doch keine Angst: TypeScript ist lediglich eine Erweiterung von JavaScript, und die neuen Konzepte sind sehr eingängig und schnell gelernt.

In diesem Buch wird ein praxisorientierter Ansatz verfolgt. Sie werden anhand einer Beispielanwendung schrittweise die Konzepte und Funktionen von Angular kennenlernen. Dabei lernen Sie nicht nur die Grundlagen kennen, sondern wir vermitteln Ihnen auch eine Vielzahl von Best Practices und Erkenntnissen aus mehrjähriger Praxis mit Angular.

*Praxisorientierte
Einsteiger*

Was sollten Sie mitbringen?

Da wir Erfahrungen in der Webentwicklung mit JavaScript voraussetzen, ist es für jeden Entwickler, der auf diesem Gebiet unerfahren ist, empfehlenswert, sich die nötigen Grundlagen zu erarbeiten. Darüber hinaus sollten Sie Grundkenntnisse im Umgang mit HTML und CSS mitbringen. Der *dpunkt.verlag* bietet eine große Auswahl an Einstiegsliteratur für HTML, JavaScript und CSS an. Sollten Sie über keinerlei TypeScript-Kenntnisse verfügen: kein Problem! Alles, was Sie über TypeScript wissen müssen, um die Inhalte dieses Buchs zu verstehen, wird in einem separaten Kapitel vermittelt.

*Grundkenntnisse in
JavaScript, HTML und
CSS*

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.

*Keine Angular-
Vorkenntnisse nötig!*

Für wen ist dieses Buch weniger geeignet?

Um Inhalte des Buchs zu verstehen, werden Erfahrungen im Webumfeld vorausgesetzt. Entwickler ohne Vorkenntnisse in der Webentwicklung werden womöglich an manchen Stellen Hilfe zurate ziehen müssen. Wir empfehlen, in diesem Fall zunächst die grundlegenden Kenntnisse in den Bereichen HTML, JavaScript und CSS zu festigen.

*Unerfahrene
Webentwickler*

Weiterhin ist dieses Buch 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. Dabei werden aber nicht alle Themen bis ins kleinste Detail betrachtet. 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 also mit den vielen praktischen Beispielen souverän meistern können.

*Kein klassisches
Nachschlagewerk*

*Offizielle Angular-
Dokumentation*

Wir hoffen, dass dieses Buch Ihr täglicher Begleiter bei der Arbeit mit Angular wird. Für Details zu den einzelnen Framework-Funktionen empfehlen wir die offizielle Dokumentation für Entwickler.³

Wie ist dieses Buch zu lesen?

*Einführung, Tools und
Schnellstart*

Wir beginnen im ersten Teil des Buchs mit einer Einführung, in der Sie alles über die verwendeten Tools und benötigtes Werkzeug erfahren. Im Schnellstart tauchen wir sofort in Angular ein und nehmen Sie mit zu einem schnellen 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 und erfahren, wie Sie die wichtigsten Features verwenden können. Entwickler, die bereits Erfahrung im Umgang mit TypeScript haben, können diesen Teil überspringen.

Beispielanwendung

Der dritte Teil ist der Hauptteil des Buchs. Hier möchten wir mit Ihnen zusammen eine Beispielanwendung entwickeln. Die Konzepte und Technologien von Angular wollen wir dabei direkt am Beispiel vermitteln. So stellen wir sicher, dass das Gelesene angewendet wird und jeder Abschnitt automatisch einen praktischen Bezug hat.

Iterationen

Nach einer Projekt- und Prozessvorstellung haben wir das Buch in mehrere Iterationen eingeteilt. In jeder Iteration gilt es Anforderungen zu erfüllen, die wir gemeinsam mit Ihnen implementieren.

- Iteration I: Komponenten & Template-Syntax (ab S. 73)
- Iteration II: Services & Routing (ab S. 131)
- Iteration III: HTTP & reaktive Programmierung (ab S. 189)
- Iteration IV: Formularverarbeitung & Validierung (ab S. 275)
- Iteration V: Pipes & Direktiven (ab S. 353)
- Iteration VI: Module & fortgeschrittenes Routing (ab S. 401)
- Iteration VII: Internationalisierung (ab S. 449)

Storys

Eine solche Iteration ist in mehrere Storys untergliedert, die jeweils ein Themengebiet abdecken. Eine Story besteht immer aus einer theoretischen Einführung und der praktischen Implementierung im Beispielprojekt. Neben Storys gibt es Refactoring-Abschnitte. Dabei handelt es sich um technische Anforderungen, die die Architektur oder den Codestil der Anwendung verbessern.

Refactoring

Powertipps

Haben wir eine Iteration abgeschlossen, prüfen wir, ob wir unseren Entwicklungsprozess vereinfachen und beschleunigen können. In den

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

Power Tipps demonstrieren wir hilfreiche Werkzeuge, die uns bei der Entwicklung zur Seite stehen.

Nachdem alle Iterationen erfolgreich absolviert wurden, wollen wir das Thema *Testing* genauer betrachten. Hier erfahren Sie, wie Sie Ihre Angular-Anwendung automatisiert testen und so die Softwarequalität sichern können. Dieses Kapitel kann sowohl nach der Entwicklung des Beispielsprojekts als auch parallel dazu bestritten werden.

Testing

Im vierten Teil dreht sich alles um das Deployment einer Angular-Anwendung. Sie werden erfahren, wie Sie eine fertig entwickelte Angular-Anwendung fit für den Produktiveinsatz machen. Dabei betrachten wir die Hintergründe und Konfiguration des Build-Prozesses und erläutern die Bereitstellung mithilfe von Docker.

Deployment

Im fünften Teil möchten wir Ihnen mit Server-Side Rendering und der Redux-Architektur zwei Ansätze näherbringen, die über eine Standardanwendung hinausgehen. Mit *Server-Side Rendering (SSR)* machen Sie Ihre Anwendung fit für Suchmaschinen und verbessern zusätzlich die Geschwindigkeit beim initialen Start der App. Anschließend stellen wir Ihnen das *Redux*-Pattern und das Framework *NgRx* vor. Sie erfahren, wie Sie mithilfe von Redux den Anwendungsstatus zentral und gut wartbar verwalten können.

Weiterführende

Themen

SSR

Redux

Der sechste Teil dieses Buchs dreht sich um mobile Anwendungen mit Angular: Nachdem wir die Begriffe rund um das Thema *App* eingeordnet haben, besprechen wir die Ideen und Implementierung einer *Progressive Web App (PWA)* mit Angular. Abschließend betrachten wir den Einsatz von *NativeScript*, um native mobile Anwendungen für verschiedene Zielplattformen (Android, iOS etc.) zu entwickeln.

Progressive Web Apps

NativeScript

Im letzten Kapitel des Buchs finden Sie weitere Informationen zu wissenswerten und begleitenden Themen. Hier haben wir weiterführende Inhalte zusammengetragen, auf die wir im Beispielsprojekt nicht ausführlich eingehen.

Wissenswertes

Abtippen statt Copy & Paste

Wir alle kennen es: Beim Lesen steht vor uns ein großer Abschnitt Quelltext, und wir haben wenig Lust auf Tipparbeit. Schnell kommt der Gedanke auf, ein paar Codezeilen oder sogar ganze Dateien aus dem Repository zu kopieren. Vielleicht denken Sie sich: »Den Inhalt anzuschauen und die Beschreibung zu lesen reicht aus, um es zu verstehen.«

An dieser Stelle möchten wir einhaken: Kopieren und Einfügen ist nicht dasselbe wie *Lernen* und *Verstehen*. Wenn Sie die Codebeispiele selbst *eintippen*, werden Sie besser verstehen, wie Angular funktioniert,

Abtippen heißt Lernen und Verstehen.

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!

Beratung und Workshops

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 schnellsten 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-Workshops in den Räumen Ihres Unternehmens, in offenen Gruppen oder als Online-Kurs an. Das Angular-Buch verwenden wir dabei in unseren Einstiegskursen zur Nacharbeit. Haben Sie das Buch vollständig gelesen, so können Sie direkt in die individuellen Kurse für Fortgeschrittene einsteigen. Wir freuen uns auf Ihren Besuch.

*Die Angular.Schule:
Workshops und
Beratung*



<https://angular.schule>

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 seine starke Unterstützung beim Kapitel zu Docker. **Danilo Hoffmann**, **Jan Buchholz**, **Manfred Steyer** und **Jan-Niklas Wortmann** danken wir ebenso für die hilfreichen Anregungen und Korrekturvorschläge. Unser Dank geht außerdem an **Michael Hladky** für wertvollen Input zur Change Detection und zur Bibliothek RxAngular. Darüber hinaus hat uns **Nathan Walker** mit seiner Zeit und Expertise beim Kapitel zu NativeScript unterstützt.

Wir danken **Gregor Woiwode** für die Mitwirkung als Autor in der ersten Auflage. Dem Team vom dpunkt.verlag, insbesondere **René Schönfeldt**, danken wir für die persönliche Unterstützung und die guten Anregungen zum Buch. **Annette Schwarz** danken wir für das gewissenhafte Korrektorat unseres Manuskripts. Besonderer Dank gilt dem **Angular-Team** und der Community dafür, dass sie eine großartige Plattform geschaffen haben, die uns den Entwickleralltag angenehmer macht.

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

Aus Gründen der Lesbarkeit verzichten wir in diesem Buch auf eine geschlechtsneutrale Formulierung. Wir möchten betonen, dass wir selbstverständlich durchgängig alle Personen jeden Geschlechts ansprechen.

Aktualisierungen in der dritten 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 drei 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.

Die dritte Auflage, die Sie mit diesem Buch in Händen halten, wurde deshalb grundlegend aktualisiert und erweitert. Dabei haben wir das Feedback unserer Leser berücksichtigt, Fehler korrigiert und viele Erklärungen verständlicher formuliert.

Wir möchten Ihnen einen kurzen Überblick über die wichtigsten Neuerungen und Aktualisierungen der dritten Auflage geben. Alle Texte und Codebeispiele haben wir auf die Angular-Version 10 aktualisiert. Dabei betrachten wir auch neue Features. Schon in der zweiten Auflage haben wir umfassende Aktualisierungen vorgenommen, die wir am Ende dieses Abschnitts zusammengefasst haben.

Neue Kapitel

In der dritten Auflage sind die folgenden Kapitel neu hinzugekommen:

10.3.5 OAuth 2 und OpenID Connect (Seite 262) In der Iteration III erläutern wir die Kommunikation mit einem HTTP-Backend und betrachten Interceptoren, mit denen wir zum Beispiel Tokens in den Header einer HTTP-Nachricht einfügen können. In diesem Zusammenhang haben wir einen neuen Abschnitt hinzugefügt, in dem wir die Authentifizierung und Autorisierung mithilfe von OAuth 2 und OpenID Connect erläutern.

19 Angular-Anwendungen mit Docker bereitstellen (Seite 563)

Nachdem wir das Thema Docker in der zweiten Auflage nur in einem kurzen Abschnitt erwähnt hatten, haben wir nun ein ausführliches Kapitel neu ins Buch aufgenommen. Dort erläutern wir am praktischen

Beispiel, wie Sie eine Angular-Anwendung in einem Docker-Container bereitstellen und ausführen können. Bei diesem Kapitel hat uns unser treuer Leser *Michael Kaaden* mit Praxiserfahrung und viel Zuarbeit unterstützt – vielen Dank!

24 Progressive Web Apps (PWA) (Seite 673) Progressive Web Apps sind ein wichtiger Pfeiler für moderne Anwendungsentwicklung mit Webtechnologien. Hier spielen besonders Service Worker, Installierbarkeit, Offlinefähigkeit und Push-Benachrichtigungen eine Rolle. Da das Thema bisher in diesem Buch nicht behandelt wurde, haben wir ein neues Kapitel entwickelt. Sie lernen dabei die Grundlagen zu Progressive Web Apps und migrieren die Beispielanwendung zu einer PWA.

27 Fortgeschrittene Konzepte der Angular CLI (Seite 735) Die Angular CLI kann mehr als nur eine Anwendung in einem Projekt verwalten. In diesem neuen Kapitel werfen wir deshalb einen Blick auf die Architektur eines *Workspace*, der mehrere Anwendungen und Bibliotheken in einem gemeinsamen Repository pflegt. Zusätzlich betrachten wir hier kurz die *Schematics*, die für die Codegenerierung in der Angular CLI verantwortlich sind.

28.1 Web Components mit Angular Elements (Seite 743) Unter »Wissenswertes« sammeln wir interessante Themen, die im Verlauf des Buchs keinen Platz gefunden haben. Hier haben wir einen neuen Abschnitt zu Angular Elements hinzugefügt. Sie lernen, wie Sie Angular-Komponenten als Web Components verpacken, um sie auch in anderen Webanwendungen einzusetzen.

Stark überarbeitete und erweiterte Kapitel

10.2 Reaktive Programmierung mit RxJS (Seite 206) Das Kapitel zu RxJS haben wir um einige wichtige Details ergänzt: So wurde die Erläuterung zu Higher-Order Observables überarbeitet und mit Marble-Diagrammen illustriert, wir haben den Unterschied zwischen Observer und Subscriber stärker herausgestellt und viele Erläuterungen vereinfacht.

15.1 i18n: mehrere Sprachen und Kulturen anbieten (Seite 449) Das Thema Internationalisierung wurde mit Angular 9.0 neu aufgerollt und verfügt nun über erweiterte Funktionen, z. B. Übersetzungen im TypeScript-Code. Wir nutzen in diesem Kapitel jetzt das neue Paket `@angular/localize`, um Übersetzungen zu rendern.

18 Build und Deployment mit der Angular CLI (Seite 539) Das Kapitel zum Deployment haben wir neu strukturiert. Hier wird nun die Build-Konfiguration in der `angular.json` detailliert erläutert. Mit dem Release des neuen Ivy-Compilers ist auch das Thema JIT mehr in den Hintergrund gerückt. Außerdem haben wir einen neuen Abschnitt zum Befehl `ng deploy` hinzugefügt.

20 Server-Side Rendering mit Angular Universal (Seite 587) Der Workflow für Server-Side Rendering wurde mit Angular 9.0 stark vereinfacht. Wir haben das Kapitel zu Angular Universal aktualisiert und erweitert: Dabei gehen wir auf den neuen Builder für statisches Pre-Rendering ein, geben Tipps für den Praxiseinsatz und betrachten das Community-Projekt *Scully*.

21 State Management mit Redux und NgRx (Seite 607) Das Framework NgRx wird stetig weiterentwickelt, und so haben wir das Kapitel zum State Management grundlegend aktualisiert. Wir setzen nun durchgehend auf die neuen Creator Functions und haben viele Erläuterungen ausführlicher und verständlicher gestaltet. Außerdem gehen wir auf das neue Paket `@ngrx/component` ein und werfen einen kurzen Blick auf das Community-Projekt *RxAngular*.

Sonstiges

Neben den genannten Kapiteln haben wir alle Texte im Buch erneut kritisch überarbeitet. An vielen Stellen haben wir Formulierungen angepasst, Details ergänzt und Fehler korrigiert. Wenn Sie weitere Fehler finden oder Anregungen zum Buch haben, so schreiben Sie uns bitte! Wir werden uns Ihr Feedback in der nächsten Auflage zu Herzen nehmen.

Fehler gefunden?

Für die einzelnen Iterationsschritte aus dem Beispielpjekt bieten wir eine Differenzansicht an. So können Sie die Änderungen am Code zwischen den einzelnen Kapiteln besser nachvollziehen. Wir gehen darauf auf Seite 53 genauer ein.

Zu guter Letzt haben wir an ausgewählten Stellen in diesem Buch Zitate von Persönlichkeiten aus der Angular-Community aufgeführt. Die meisten dieser Zitate haben wir direkt für dieses Buch erbeten. Wir freuen uns sehr, dass so viele interessante und humorvolle Worte diesem Buch eine einmalige Note geben.

Aktualisierungen in der zweiten Auflage

Neue Kapitel

Folgende Kapitel und Abschnitte sind in der **zweiten Auflage** neu hinzugekommen:

- 10.3 Interceptoren: HTTP-Requests abfangen und transformieren (Seite 257)
- 20 Server-Side Rendering mit Angular Universal (Seite 587)
- 28 Wissenswertes (Seite 743)
 - 27.2 Schematics: Codegenerierung mit der Angular CLI (Seite 740)
 - 28.2 Container und Presentational Components (Seite 751)
 - 28.4 TrackBy-Funktion für die Direktive ngFor (Seite 756)
 - 28.6 Angular Material und weitere UI-Komponentensammlungen (Seite 762)
 - 28.11 Angular updaten (Seite 785)

Vollständig neu geschriebene Kapitel

Einige bereits in der ersten Auflage existierende Kapitel wurden für die **zweite Auflage** vollständig neu aufgerollt:

1 Schnellstart (Seite 3) Der Schnellstart basierte in der ersten Auflage auf einer lokalen Lösung mit SystemJS und Paketen aus einem CDN. Der neue Schnellstart setzt auf die Online-Plattform StackBlitz zum schnellen Prototyping von Webanwendungen.

10.2 Reaktive Programmierung mit RxJS (Seite 206) Das Prinzip der reaktiven Programmierung und das Framework RxJS haben in den letzten Jahren weiter an Bedeutung gewonnen. Das alte Kapitel zu RxJS lieferte nur einen kurzen Überblick, ohne auf Details einzugehen. Mit dieser Neufassung finden Sie jetzt eine ausführliche Einführung in die Prinzipien von reaktiver Programmierung und Observables, und es werden alle wichtigen Konzepte anhand von Beispielen erklärt. Im Gegensatz zur ersten Auflage verwenden wir die neuen *Pipeable Operators*.

12 Formularverarbeitung & Validierung: Iteration IV (Seite 275) In der ersten Auflage haben wir sowohl *Template-Driven Forms* als auch *Reactive Forms* gleichbedeutend vorgestellt. Wir empfehlen mittlerweile nicht mehr den Einsatz von Template-Driven Forms. Daher stellen

wir zwar beide Ansätze weiterhin vor, legen aber im Kapitel zur Formularverarbeitung einen stärkeren Fokus auf Reactive Forms. Das Praxisbeispiel wurde neu entworfen, um eine saubere Trennung der Zuständigkeiten der Komponenten zu ermöglichen. Die Erläuterungen im Grundlagenteil wurden neu formuliert, um besser für die Anforderungen aus der Praxis geeignet zu sein.

21 State Management mit Redux und NgRx (Seite 607) In den letzten zwei Jahren hat sich unserer Ansicht nach das Framework *NgRx* gegen weitere Frameworks wie *angular-redux* klar durchgesetzt. Während die erste Auflage in diesem Kapitel noch auf *angular-redux* setzte, arbeitet das Kapitel der zweiten Auflage durchgehend mit den *Reactive Extensions for Angular (NgRx)*. Wir erarbeiten in der Einführung schrittweise ein Modell für zentrales State Management, um die Architektur von Redux zu erläutern, ohne eine konkrete Bibliothek zu nutzen.

Stark überarbeitete und erweiterte Kapitel

4 Einführung in TypeScript (Seite 27) Das Grundlagenkapitel zu TypeScript wurde neu strukturiert und behandelt zusätzlich auch neuere Features von ECMAScript/TypeScript, z. B. Destrukturierung, Spread-Operator und Rest-Syntax.

10.1 HTTP-Kommunikation: ein Server-Backend anbinden (Seite 189) Das HTTP-Kapitel setzt durchgehend auf den `HttpClient`, der mit Angular 4.3 eingeführt wurde. Dabei wird der Blick auch auf die erweiterten Features des Clients geworfen. Themen, die spezifisch für RxJS sind, wurden aus diesem Kapitel herausgelöst und werden nun im RxJS-Kapitel behandelt.

14.4.1 Resolver: asynchrone Daten beim Routing vorladen (Seite 441) Resolver sind aus unserer Sicht nicht die beste Wahl, um reguläre Daten über HTTP nachzuladen. Der Iterationsschritt zu Resolvieren wurde deshalb aus dem Beispielpjekt entfernt, und das Thema wird in dieser Auflage nur noch in der Theorie behandelt.

15.1 i18n: mehrere Sprachen und Kulturen anbieten (Seite 449) Die Möglichkeiten zur Konfiguration des Builds wurden mit Angular 6.0 stark vorangebracht. Viele zuvor notwendige Kommandozeilenparameter sind nun nicht mehr notwendig, die Konfigurationsdatei

angular.json löst diese ab. Dadurch konnten wir das Kapitel zur Internationalisierung (i18n) kürzen und verständlicher gestalten. Im Gegensatz zur ersten Auflage zeigen wir nicht mehr, wie man eine Anwendung im JIT-Modus internationalisiert, der hauptsächlich für die Entwicklung vorgesehen ist, aber nicht für produktive Anwendungen.

17.1 Softwaretests (Seite 483) Das Kapitel zum Testen von Angular-Anwendungen wurde stark erweitert. Neben den reinen Werkzeugen wird der Fokus besonders auf Philosophien, Patterns und Herangehensweisen gelegt. Zusätzlich werden die mitgelieferten Tools zum Testen von HTTP und Routing betrachtet.

25 NativeScript: mobile Anwendungen entwickeln (Seite 695) Zur Entwicklung einer nativen mobilen Anwendung nutzen wir in diesem Kapitel die neue Version 6 von NativeScript, die insbesondere Verbesserungen in Sachen Codegenerierung und Wiederverwendbarkeit von Code mitbringt.

28.9 Change Detection (Seite 770) Das Kapitel zur Change Detection wurde für besseres Verständnis neu strukturiert. Insbesondere wird auf Debugging und Strategien zur Optimierung eingegangen.

Inhaltsverzeichnis

Vorwort	vii
----------------------	------------

Aktualisierungen in der dritten Auflage	xvii
--	-------------

I Einführung 1

1 Schnellstart	3
1.1 Das HTML-Grundgerüst	6
1.2 Die Startdatei für das Bootstrapping	6
1.3 Das zentrale Angular-Modul	7
1.4 Die erste Komponente	8
2 Haben Sie alles, was Sie benötigen?	11
2.1 Visual Studio Code	11
2.2 Google Chrome	14
2.3 Paketverwaltung mit Node.js und NPM	14
2.4 Codebeispiele in diesem Buch	17
3 Angular CLI: der Codegenerator für unser Projekt	21
3.1 Das offizielle Tool für Angular	21
3.2 Installation	22
3.3 Die wichtigsten Befehle	23

II TypeScript 25

4 Einführung in TypeScript	27
4.1 Was ist TypeScript und wie setzen wir es ein?	27
4.2 Variablen: const, let und var	30
4.3 Die wichtigsten Basistypen	32
4.4 Klassen	35
4.5 Interfaces	39
4.6 Template-Strings	40

4.7	Arrow-Funktionen/Lambda-Ausdrücke	40
4.8	Spread-Operator und Rest-Syntax	42
4.9	Union Types	45
4.10	Destrukturierende Zuweisungen	45
4.11	Decorators	47
4.12	Optional Chaining	47
4.13	Nullish Coalescing	48

III BookMonkey 4: Schritt für Schritt zur App 51

5	Projekt- und Prozessvorstellung	53
5.1	Unser Projekt: BookMonkey	53
5.2	Projekt mit Angular CLI initialisieren	57
5.3	Style-Framework Semantic UI einbinden	70
6	Komponenten & Template-Syntax: Iteration I	73
6.1	Komponenten: die Grundbausteine der Anwendung	73
6.1.1	Komponenten	74
6.1.2	Komponenten in der Anwendung verwenden	79
6.1.3	Template-Syntax	80
6.1.4	Den BookMonkey erstellen	90
6.2	Property Bindings: mit Komponenten kommunizieren	102
6.2.1	Komponenten verschachteln	102
6.2.2	Eingehender Datenfluss mit Property Bindings	103
6.2.3	Andere Arten von Property Bindings	106
6.2.4	DOM-Propertys in Komponenten auslesen	109
6.2.5	Den BookMonkey erweitern	110
6.3	Event Bindings: auf Ereignisse in Komponenten reagieren	114
6.3.1	Native DOM-Events	114
6.3.2	Eigene Events definieren	117
6.3.3	Den BookMonkey erweitern	119
7	Powertipp: Styleguide	129
8	Services & Routing: Iteration II	131
8.1	Dependency Injection: Code in Services auslagern	131
8.1.1	Abhängigkeiten anfordern	133
8.1.2	Services in Angular	134
8.1.3	Abhängigkeiten registrieren	134
8.1.4	Abhängigkeiten ersetzen	137
8.1.5	Eigene Tokens definieren mit InjectionToken	140
8.1.6	Abhängigkeiten anfordern mit @Inject()	141

8.1.7	Multiprovider: mehrere Abhängigkeiten im selben Token	141
8.1.8	Zirkuläre Abhängigkeiten auflösen mit forwardRef ...	142
8.1.9	Provider in Komponenten registrieren	142
8.1.10	Den BookMonkey erweitern	143
8.2	Routing: durch die Anwendung navigieren	147
8.2.1	Routen konfigurieren	149
8.2.2	Routing-Modul einbauen	149
8.2.3	Komponenten anzeigen	152
8.2.4	Root-Route	153
8.2.5	Routen verlinken	154
8.2.6	Routenparameter	156
8.2.7	Verschachtelung von Routen	158
8.2.8	Routenweiterleitung	161
8.2.9	Wildcard-Route	162
8.2.10	Aktive Links stylen	162
8.2.11	Route programmatisch wechseln	163
8.2.12	Den BookMonkey erweitern	165
9	Power Tipp: Chrome Developer Tools	177
10	HTTP & reaktive Programmierung: Iteration III	189
10.1	HTTP-Kommunikation: ein Server-Backend anbinden	189
10.1.1	Modul einbinden	191
10.1.2	Requests mit dem HttpClient durchführen	191
10.1.3	Optionen für den HttpClient	193
10.1.4	Den BookMonkey erweitern	196
10.2	Reaktive Programmierung mit RxJS	206
10.2.1	Alles ist ein Datenstrom	207
10.2.2	Observables sind Funktionen	208
10.2.3	Das Observable aus RxJS	211
10.2.4	Observables abonnieren	212
10.2.5	Observables erzeugen	214
10.2.6	Operatoren: Datenströme modellieren	217
10.2.7	Heiße Observables, Multicasting und Subjects	222
10.2.8	Subscriptions verwalten & Memory Leaks vermeiden	227
10.2.9	Flattening-Strategien für Higher-Order Observables	231
10.2.10	Den BookMonkey erweitern: Daten vom Server typisieren und umwandeln	237
10.2.11	Den BookMonkey erweitern: Fehlerbehandlung	242
10.2.12	Den BookMonkey erweitern: Typeahead-Suche	246
10.3	Interceptoren: HTTP-Requests abfangen und transformieren	257
10.3.1	Warum HTTP-Interceptoren nutzen?	257

10.3.2	Funktionsweise der Interceptoren	257
10.3.3	Interceptoren anlegen	258
10.3.4	Interceptoren einbinden	260
10.3.5	OAuth 2 und OpenID Connect	262
10.3.6	Den BookMonkey erweitern	265
11	Power Tipp: Komponenten untersuchen mit Augury	271
12	Formularverarbeitung & Validierung: Iteration IV	275
12.1	Angulars Ansätze für Formulare	276
12.2	Template-Driven Forms	276
12.2.1	FormsModule einbinden	277
12.2.2	Datenmodell in der Komponente	277
12.2.3	Template mit Two-Way Binding und ngModel	278
12.2.4	Formularzustand verarbeiten	279
12.2.5	Eingaben validieren	280
12.2.6	Formular abschicken	281
12.2.7	Formular zurücksetzen	282
12.2.8	Den BookMonkey erweitern	284
12.3	Reactive Forms	303
12.3.1	Warum ein zweiter Ansatz für Formulare?	303
12.3.2	Modul einbinden	304
12.3.3	Formularmodell in der Komponente	304
12.3.4	Template mit dem Modell verknüpfen	307
12.3.5	Formularzustand verarbeiten	309
12.3.6	Eingebaute Validatoren nutzen	310
12.3.7	Formular abschicken	311
12.3.8	Formular zurücksetzen	312
12.3.9	Formularwerte setzen	312
12.3.10	FormBuilder verwenden	313
12.3.11	Änderungen überwachen	314
12.3.12	Den BookMonkey erweitern	315
12.4	Eigene Validatoren entwickeln	335
12.4.1	Validator für einzelne Formularfelder	335
12.4.2	Validator für Formulargruppen und -Arrays	338
12.4.3	Asynchrone Validatoren	340
12.4.4	Den BookMonkey erweitern	343
12.5	Welcher Ansatz ist der richtige?	351
13	Pipes & Direktiven: Iteration V	353
13.1	Pipes: Daten im Template formatieren	353
13.1.1	Pipes verwenden	353
13.1.2	Die Sprache fest einstellen	354

13.1.3	Eingebaute Pipes für den sofortigen Einsatz	356
13.1.4	Eigene Pipes entwickeln	367
13.1.5	Pipes in Komponenten nutzen	370
13.1.6	Den BookMonkey erweitern: Datum formatieren mit der DatePipe	371
13.1.7	Den BookMonkey erweitern: Observable mit der AsyncPipe auflösen	373
13.1.8	Den BookMonkey erweitern: eigene Pipe für die ISBN implementieren	376
13.2	Direktiven: das Vokabular von HTML erweitern	380
13.2.1	Was sind Direktiven?	380
13.2.2	Eigene Direktiven entwickeln	381
13.2.3	Attributdirektiven	383
13.2.4	Strukturdirektiven	388
13.2.5	Den BookMonkey erweitern: Attributdirektive für vergrößerte Darstellung	393
13.2.6	Den BookMonkey erweitern: Strukturdirektive für zeitverzögerte Sterne	396
14	Module & fortgeschrittenes Routing: Iteration VI	401
14.1	Die Anwendung modularisieren: das Modulkonzept von Angular	401
14.1.1	Module in Angular	401
14.1.2	Grundaufbau eines Moduls	402
14.1.3	Bestandteile eines Moduls deklarieren	403
14.1.4	Anwendung in Feature-Module aufteilen	405
14.1.5	Aus Modulen exportieren: Shared Module	408
14.1.6	Den BookMonkey erweitern	409
14.2	Lazy Loading: Angular-Module asynchron laden	419
14.2.1	Warum Module asynchron laden?	420
14.2.2	Lazy Loading verwenden	420
14.2.3	Module asynchron vorladen: Preloading	424
14.2.4	Den BookMonkey erweitern	425
14.3	Guards: Routen absichern	430
14.3.1	Grundlagen zu Guards	431
14.3.2	Guards implementieren	432
14.3.3	Guards verwenden	435
14.3.4	Den BookMonkey erweitern	435
14.4	Routing: Wie geht's weiter?	441
14.4.1	Resolver: asynchrone Daten beim Routing vorladen .	441
14.4.2	Mehrere Router-Outlets verwenden	445
14.4.3	Erweiterte Konfigurationen für den Router	446

15	Internationalisierung: Iteration VII	449
15.1	i18n: mehrere Sprachen und Kulturen anbieten	449
15.1.1	Was bedeutet Internationalisierung?	449
15.1.2	Eingebaute Pipes mehrsprachig verwenden	450
15.1.3	Texte übersetzen: Vorgehen in fünf Schritten	451
15.1.4	@angular/localize hinzufügen	452
15.1.5	Nachrichten im HTML mit dem i18n-Attribut markieren	452
15.1.6	Nachrichten im TypeScript-Code mit \$localize markieren	453
15.1.7	Nachrichten extrahieren und übersetzen	453
15.1.8	Feste IDs vergeben	454
15.1.9	Die App mit Übersetzungen bauen	455
15.1.10	Übersetzte Apps mit unterschiedlichen Konfigurationen bauen	459
15.1.11	Ausblick: Übersetzungen dynamisch zur Startzeit bereitstellen	466
15.1.12	Den BookMonkey erweitern	469
16	Power Tipp: POEditor	477
17	Qualität fördern mit Softwaretests	483
17.1	Softwaretests	483
17.1.1	Testabdeckung: Was sollte man testen?	484
17.1.2	Testart: Wie sollte man testen?	486
17.1.3	Test-Framework Jasmine	487
17.1.4	»Arrange, Act, Assert« mit Jasmine	490
17.1.5	Test-Runner Karma	492
17.1.6	E2E-Test-Runner Protractor	492
17.1.7	Weitere Frameworks	494
17.2	Unit- und Integrationstests mit Karma	495
17.2.1	TestBed: die Testbibliothek von Angular	495
17.2.2	Isolierte Unit-Tests: Services testen	496
17.2.3	Isolierte Unit-Tests: Pipes testen	498
17.2.4	Isolierte Unit-Tests: Komponenten testen	500
17.2.5	Shallow Unit-Tests: einzelne Komponenten testen ...	503
17.2.6	Integrationstests: mehrere Komponenten testen ...	506
17.2.7	Abhängigkeiten durch Stubs ersetzen	508
17.2.8	Abhängigkeiten durch Mocks ersetzen	513
17.2.9	Leere Komponenten als Stubs oder Mocks einsetzen	515
17.2.10	HTTP-Requests testen	516
17.2.11	Komponenten mit Routen testen	520
17.2.12	Asynchronen Code testen	524

17.3	Oberflächentests mit Protractor	526
17.3.1	Protractor verwenden	527
17.3.2	Elemente selektieren: Locators	528
17.3.3	Aktionen durchführen	529
17.3.4	Asynchron mit Warteschlange	530
17.3.5	Redundanz durch Page Objects vermeiden	531
17.3.6	Eine Angular-Anwendung testen	532

IV Das Projekt ausliefern: Deployment 537

18	Build und Deployment mit der Angular CLI	539
18.1	Build-Konfiguration	540
18.2	Build erzeugen	542
18.3	Umgebungen konfigurieren	544
18.3.1	Abhängigkeit zur Umgebung vermeiden	547
18.3.2	Konfigurationen und Umgebungen am Beispiel: BookMonkey	548
18.4	Produktivmodus aktivieren	550
18.5	Die Templates kompilieren	551
18.5.1	Ahead-of-Time-Kompilierung (AOT)	552
18.5.2	Just-in-Time-Kompilierung (JIT)	552
18.6	Bundles analysieren mit source-map-explorer	554
18.7	Webserver konfigurieren und die Anwendung ausliefern	555
18.7.1	ng deploy: Deployment mit der Angular CLI	558
18.7.2	Ausblick: Deployment mit einem Build-Service	560
19	Angular-Anwendungen mit Docker bereitstellen	563
19.1	Docker	564
19.2	Docker Registry	565
19.3	Lösungsskizze	565
19.4	Eine Angular-App über Docker bereitstellen	566
19.5	Build Once, Run Anywhere: Konfiguration über Docker verwalten	571
19.6	Multi-Stage Builds	577
19.7	Grenzen der vorgestellten Lösung	582
19.8	Fazit	583