

Vor etwas mehr als zehn Jahren galt für gute Webanwendungen noch die Regel, so viele Aufgaben wie möglich auf dem Server zu erledigen. Inzwischen stützen sich moderne Webanwendungen jedoch auf clientseitige Techniken, allen voran JavaScript. Dies steigert die Benutzerfreundlichkeit und schafft die Möglichkeit, die jeweilige Anwendung an die Auflösungen und Formfaktoren der vielen unterschiedlichen klassischen und mobilen Plattformen anzupassen.

Single Page Applications (SPAs) bilden einen derzeit äußerst beliebten Architekturstil für solche Webanwendungen. Wie ihr Name schon vermuten lässt, bestehen SPAs aus lediglich einer einzigen Seite, die ein Browser auf klassischem Weg abrufen und anzeigt. Alle weiteren Seiten und Daten bezieht die SPA bei Bedarf über direkte HTTP-Zugriffe per JavaScript.

Das populäre JavaScript-Framework Angular, das von Google entwickelt wird, hilft Ihnen beim Erstellen von Anwendungen, die diesen Architekturstil verfolgen. Angular bietet unter anderem Unterstützung für die Datenbindung, für das Validieren von Daten sowie das Arbeiten mit Vorlagen. Darüber hinaus stellt Angular Konzepte zur Verfügung, mit denen Sie den Quellcode strukturieren und wartbare, wiederverwendbare sowie testbare Programmteile erschaffen können. Das vorliegende Buch präsentiert die Möglichkeiten von Angular. Dabei beschränkt es sich nicht nur auf die Grundlagen, sondern geht auch auf die zugrunde liegenden Konzepte ein.

Zielgruppe

Das Buch richtet sich an Entwickler, die bereits grundlegende Erfahrungen mit HTML, CSS und JavaScript gesammelt haben und nun mit Angular SPAs entwickeln wollen. Dabei bezieht es sich auf die Version 12 von Angular, die beim Verfassen der Texte die aktuellste Version war. Aufgrund des evolutionären Charakters von Angular gehen wir jedoch davon aus, dass dieses Buch auch für viele darauffolgende Versionen geeignet ist.

Zielsetzung des Buchs

Mit diesem Buch verfolgen wir das Ziel, Ihnen anhand von Beispielen zu zeigen, wie Sie Angular zur Entwicklung von Single Page Applications nutzen können. Dabei gehen wir auf die Möglichkeiten von Angular ein und präsentieren auch Lösungen für Aspekte, die Angular nicht direkt unterstützt.

Quellcodebeispiele, Onlineservices und Errata

Über <http://www.ANGULARarchitects.io/leser> stellen wir Ihnen sämtliche in diesem Buch präsentierten Quellcodebeispiele sowie Web-APIs zur Verfügung, die die Beispiele zu Demonstrationszwecken nutzen und die Sie auch in eigene Projekte einbinden können. Darüber hinaus werde ich auf dieser Seite weitere Informationen zu Angular sowie gegebenenfalls Errata zum vorliegenden Buch veröffentlichen. Daneben bietet die Website Ihnen die Möglichkeit, mit mir als Autor direkt in Kontakt zu kommen.

Konventionen

Kursiv

Wird genutzt für neue Begriffe, URLs, Dateinamen und E-Mail-Adressen

Nichtproportionalschrift

Programmlistings und Codeelemente im Fließtext wie Methoden, Module o.Ä. werden in dieser Schrift dargestellt.



Dieses Symbol steht für Hinweise und allgemeinere Anmerkungen.



Dieses Symbol steht für Tipps.

Aufbau des Buchs

Das Buch besteht aus 19 Kapiteln. Die folgende Auflistung zeigt, was diese bieten:

- Kapitel 1, *Projekt-Setup*: In diesem Kapitel erfahren Sie, welche Schritte nötig sind, um mit Angular loszulegen. Dazu lernen Sie unter anderem das *Angular Command Line Interface (CLI)* kennen. Damit generieren wir auch schon das Grundgerüst für unsere erste Angular-Anwendung.

- Kapitel 2, *Erste Schritte mit TypeScript*: Die bevorzugte Sprache zum Entwickeln mit Angular ist TypeScript. Sie orientiert sich am ECMAScript-Standard und bietet zusätzlich ein statisches Typsystem, um Fehler frühzeitig erkennen zu können. In diesem Kapitel lernen Sie die wichtigsten Merkmale dieser JavaScript-Erweiterung kennen.
- Kapitel 3, *Eine erste Angular-Anwendung*: Dieses Kapitel zeigt Ihnen, wie Sie mit Angular eine erste einfache Anwendung erstellen können, die via HTTP Daten abrufen und darstellt. Darüber hinaus lernen Sie hier die Grundlagen zu Komponenten und Modulen kennen. Auch in das Thema Datenbindung wird eingeführt.
- Kapitel 4, *Komponenten und Datenbindung*: Angular-Anwendungen sind Komponenten, die aus weiteren Komponenten bestehen. In diesem Kapitel erfahren Sie, wie Sie eigene Komponenten erstellen können, die über Datenbindung mit anderen Komponenten kommunizieren.
- Kapitel 5, *Services und Dependency Injection*: Services sind unter Angular wiederverwendbare Codeeinheiten. Dank Dependency Injection können Sie sie austauschbar gestalten. Dieses Kapitel zeigt die Möglichkeiten zum Entwickeln solcher Services auf. Außerdem erfahren Sie hier, wie Sie Services global oder auch nur für bestimmte Komponenten registrieren können.
- Kapitel 6, *Pipes*: Pipes helfen dabei, Daten im Rahmen der Datenbindung zu transformieren. Sie kommen zum Beispiel zum Formatieren oder Umschlüsseln von Informationen zum Einsatz. Dieses Kapitel informiert über die in Angular enthaltenen Pipes und erklärt, wie Sie in Ihren Projekten eigene Pipes schreiben können.
- Kapitel 7, *Module*: Zum Strukturieren von Anwendungen setzt Angular auf Module. In diesem Kapitel lernen Sie eine typische Modulstruktur für Angular-Anwendungen kennen. Außerdem erfahren Sie, wie Sie eigene Module umsetzen können.
- Kapitel 8, *Routing*: Mit Routing lassen sich in einer Single Page Application mehrere Seiten simulieren und so Navigationsstrukturen etablieren. Dieses Kapitel geht auf den Angular-Router ein, der sich um diese Aufgabe kümmert.
- Kapitel 9, *Template-getriebene Formulare und Validierung*: In diesem Kapitel lernen Sie, Template-getriebene Formulare mit Angular aufzubauen. Außerdem erfahren Sie, wie Sie Eingaben validieren und eigene Formularsteuerelemente anbieten können.
- Kapitel 10, *Reaktive Formulare*: Reaktive Formulare bilden eine sehr mächtige Alternative zu den im vorangegangenen Kapitel präsentierten Template-getriebenen Formularen. Hier erfahren Sie, was es damit auf sich hat.
- Kapitel 11, *Reactive Extensions Library for JavaScript (RxJS)*: Angular nutzt die von der Bibliothek RxJS angebotenen Observables zur Darstellung asynchroner Operationen. In diesem Kapitel werfen wir einen etwas genaueren Blick auf

RxJS und die zugrunde liegenden Konzepte. Wir machen uns mit den verschiedenen Gruppen von Operatoren vertraut und lernen populäre Vertreter dieser Gruppen kennen. Außerdem beschäftigen wir uns mit den Schritten des reaktiven Entwurfs, mit Hot und Cold Observables bzw. Multicasting sowie mit Fehlerbehandlung, Flattening und dem Kombinieren verschiedener Datenflüsse.

- Kapitel 12, *Testautomatisierung*: Wie die Qualität von Angular-Code mit automatisierten Tests geprüft werden kann, erfahren Sie in diesem Kapitel. Wir gehen dazu auf das populäre Testing-Framework Jasmine in Kombination mit den Angular Testing Utilities ein und zeigen Ihnen, wie Sie damit Tests für Ihren Angular-Code schreiben können. Begriffe und Techniken wie Mocks und Spys werden ebenfalls thematisiert.
- Kapitel 13, *Performancetuning*: In diesem Kapitel betrachten wir zwei elementare Stellschrauben, die die Performance von Angular-Anwendungen beeinflussen, und lernen dabei auch die Art und Weise, wie das Framework arbeitet, besser kennen. Zum einen betrachten wir die Datenbindungsstrategie OnPush und nutzen sie gemeinsam mit Observables und Immutables, zum anderen beschäftigen wir uns mit Lazy-Loading- und Preloading-Strategien.
- Kapitel 14, *Querschnittsfunktionen*: Querschnittsfunktionen sind meist technische Anforderungen, die es immer und immer wieder zu berücksichtigen gilt. Beispiele dafür sind unter anderem Authentifizierung, Protokollierung und die Behandlung von Fehlern. In diesem Kapitel beschäftigen wir uns mit Mechanismen zur Implementierung solcher Querschnittsfunktionen, unter anderem mit Guards, Resolver und HttpInterceptoren.
- Kapitel 15, *Authentifizierung und Autorisierung*: Die wenigsten Anwendungen kommen ohne Authentifizierung und Autorisierung aus. Bei Single Page Applications bieten sich hierzu neben Cookies auch tokenbasierte Verfahren an. Dieses Kapitel beschreibt beide Optionen. Für Letztere geht es auf die populären Standards OAuth 2 und OpenID Connect ein und zeigt, wie Sie damit ein Security-Token anfordern können. Anschließend erfahren Sie, wie sich dieses Vorgehen in Ihren Angular-Anwendungen umsetzen lässt.
- Kapitel 16, *Internationalisierung*: In diesem Kapitel zeigen wir Ihnen, wie das Anpassen von Angular-Anwendungen für Benutzer verschiedener Länder und Sprachen funktioniert. Die dafür in den Angular-Compiler integrierten Möglichkeiten betrachten wir dabei genauso wie den Einsatz der populären Bibliothek *ngx-translate*.
- Kapitel 17, *Reaktive Zustandsverwaltung mit NGRX (Redux)*: State Management hilft beim Verwalten lokal vorgehaltener Daten, auf die unter anderem mehrere Komponenten zugreifen. Die wohl populärste Bibliothek, die dabei unterstützt, ist NGRX. In diesem Kapitel erfahren Sie, was sich dahinter verbirgt und wie Sie diese Bibliothek in Ihrer Angular-Anwendung nutzen können.

- Kapitel 18, *Details zu Komponenten und Direktiven*: Obwohl seit der Angular-Einführung in Kapitel 3 ständig Komponenten zum Einsatz kommen, gibt es doch noch einige Details, die unter anderem bei der Entwicklung wiederverwendbarer Komponenten nützlich sind. Diese lernen Sie hier kennen. Außerdem erfahren Sie, wie sich mit Direktiven wiederkehrende Aufgaben umsetzen lassen.
- Kapitel 19, *Wiederverwendbare Bibliotheken und Monorepos*: In diesem Kapitel erfahren Sie, wie sich Ihre Angular-Anwendungen mittels Bibliotheken in einem mit der Angular CLI generierten Monorepo untergliedern lassen, aber auch, wie sie diese Bibliotheken als wiederverwendbare npm-Pakete für andere Projektteams veröffentlichen können.

Schulungen und Beratung

Der Autor bietet Schulungen und Beratung zu Angular an. Informationen dazu finden Sie unter <http://www.ANGULARarchitects.io>.

Danksagungen

Meinen Dank für ihre Mitwirkung an diesem Buch möchte ich aussprechen an

- Ariane Hesse, Lektorat, und Sibylle Feldmann, Korrektorat, die das Manuskript bearbeitet haben.
- Hans-Peter Grahl, der sehr sorgfältig und unter vollstem Einsatz seines berechtigten Adlerblicks äußerst wertvolles inhaltliches sowie fachliches Feedback zu den Texten und den Beispielen gegeben hat.