

**Gregor Woiwode · Ferdinand Malcher · Danny Koppenhagen ·
Johannes Hoppe**

Angular

**Grundlagen, fortgeschrittene Techniken und
Best Practices mit TypeScript – ab Angular 4,
inklusive NativeScript und Redux**



dpunkt.verlag

X EDITION

Inhaltsverzeichnis

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

I Einführung

1 Haben Sie alles, was Sie benötigen?	3
1.1 Visual Studio Code	3
1.2 Google Chrome mit Augury	6
1.3 Paketverwaltung mit Node.js und NPM	6
1.4 Codebeispiele in diesem Buch	9
2 Schnellstart	11
2.1 Die erste Angular-Anwendung aufsetzen	11
2.2 HTML-Grundgerüst erstellen	12
2.3 Den Modul-Loader konfigurieren	13
2.4 Die Startdatei für das Bootstrapping anlegen	15
2.5 Das zentrale Angular-Modul anlegen	16
2.6 Eine erste Komponente anlegen	17
2.7 Den Webserver starten	18
2.8 Retrospektive	19
3 Angular CLI: Der Codegenerator für unser Projekt	21
3.1 Vorstellung	21
3.2 Installation	22
3.3 Die wichtigsten Befehle	23

II TypeScript

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 Getter und Setter	31
4.4 Die wichtigsten Basistypen	32

4.5	Klassen	34
4.6	Interfaces	38
4.7	Operatoren und Funktionen	40
4.8	Dekoratoren	42

III BookMonkey 2: Schritt für Schritt zur App 45

5	Projekt- und Prozessvorstellung	47
5.1	Unser Projekt: Der BookMonkey 2	47
5.2	Projekt mit Angular CLI initialisieren	51
5.3	Style-Framework Semantic UI einbinden	59
6	Komponenten & Template-Syntax: Iteration I	63
6.1	Komponenten: Die Grundbausteine der Anwendung	63
6.1.1	Komponenten	64
6.1.2	Komponenten in der Anwendung verwenden	69
6.1.3	Template-Syntax	70
6.1.4	Den BookMonkey erstellen	79
6.2	Property Bindings: Mit Komponenten kommunizieren	88
6.2.1	Komponenten verschachteln	88
6.2.2	Eingehender Datenfluss mit Property Bindings	89
6.2.3	Andere Arten von Property Bindings	93
6.2.4	DOM-Propertys in Komponenten auslesen	95
6.2.5	Den BookMonkey erweitern	96
6.3	Event Bindings: Auf Ereignisse in Komponenten reagieren ...	101
6.3.1	Native DOM-Events	101
6.3.2	Eigene Events definieren	104
6.3.3	Den BookMonkey erweitern	106
7	Powertipp: Styleguide	115
8	Services & Routing: Iteration II	117
8.1	Dependency Injection: Code in Services auslagern	117
8.1.1	Abhängigkeiten anfordern	119
8.1.2	Eingebaute Abhängigkeiten	120
8.1.3	Abhängigkeiten bereitstellen	120
8.1.4	Den BookMonkey erweitern	124
8.2	Routing: Durch die Anwendung navigieren	128
8.2.1	Routen konfigurieren	129
8.2.2	Routing-Modul einbauen	130
8.2.3	Komponenten anzeigen	132
8.2.4	Root-Route	133
8.2.5	Routen verlinken	134

8.2.6	Routenparameter	135
8.2.7	Verschachtelung von Routen	137
8.2.8	Routenweiterleitung	139
8.2.9	Aktive Links stylen	141
8.2.10	Route programmatisch wechseln	141
8.2.11	Den BookMonkey erweitern	143
9	Powertipp: Chrome Developer Tools	157
10	HTTP & reaktive Programmierung: Iteration III	169
10.1	HTTP: Ein Server-Backend anbinden	169
10.1.1	Das HTTP-Modul einbinden	170
10.1.2	Einfache Requests mit der Http-Klasse	171
10.1.3	Benutzerdefinierte Anfragen mit der Request-Klasse ..	173
10.1.4	Request erweitern: Zusätzliche Header	173
10.1.5	Den BookMonkey erweitern	175
10.2	RxJS: Reaktive Programmierung	186
10.2.1	Observables	187
10.2.2	Operatoren	190
10.2.3	Den BookMonkey erweitern	193
11	Powertipp: Augury	203
12	Formularverarbeitung & Validierung: Iteration IV	207
12.1	Angulars Ansätze für Formulare	208
12.2	Template Driven Forms	209
12.2.1	Template Driven Forms verwenden	209
12.2.2	NgModel und NgModelGroup	211
12.2.3	Eingaben validieren	214
12.2.4	Den BookMonkey erweitern	215
12.3	Reactive Forms	227
12.3.1	Reactive Forms verwenden	227
12.3.2	Den FormBuilder verwenden	231
12.3.3	Vorhandene Validatoren nutzen	232
12.3.4	Den BookMonkey erweitern	233
12.4	Custom Validators	245
12.4.1	Validatoren für einzelne Formularfelder	245
12.4.2	Validatoren für Formulargruppen und -Arrays	247
12.4.3	Asynchrone Validatoren	249
12.4.4	Den BookMonkey erweitern	252

- 13 Pipes & Direktiven: Iteration V 259**
 - 13.1 Pipes: Daten im Template formatieren 259
 - 13.1.1 Pipes verwenden 259
 - 13.1.2 Die Sprache einstellen 260
 - 13.1.3 Integrierte Pipes für den sofortigen Einsatz 261
 - 13.1.4 Eigene Pipes entwickeln (Custom Pipes)..... 271
 - 13.1.5 Pipes in Komponenten nutzen..... 273
 - 13.1.6 Den BookMonkey erweitern 274
 - 13.2 Direktiven: Das Vokabular von HTML erweitern 282
 - 13.2.1 Was sind Direktiven? 282
 - 13.2.2 Eine erste eigene Direktive schreiben 283
 - 13.2.3 Attributdirektiven..... 285
 - 13.2.4 Strukturdirektiven 290
 - 13.2.5 Den BookMonkey erweitern 293
- 14 Module & fortgeschrittenes Routing: Iteration VI 301**
 - 14.1 Die Anwendung modularisieren: Das Modulkonzept von Angular 301
 - 14.1.1 Module in Angular 301
 - 14.1.2 Grundaufbau eines Moduls..... 302
 - 14.1.3 Bestandteile eines Moduls deklarieren 302
 - 14.1.4 Anwendung in Feature-Module aufteilen 304
 - 14.1.5 Aus Modulen exportieren: Shared Module 307
 - 14.1.6 Den BookMonkey erweitern 309
 - 14.2 Lazy Loading: Angular-Module asynchron laden..... 318
 - 14.2.1 Warum Module asynchron laden? 318
 - 14.2.2 Lazy Loading verwenden..... 319
 - 14.2.3 Module asynchron vorladen: Preloading 321
 - 14.2.4 Den BookMonkey erweitern 322
 - 14.3 Guards: Routen absichern 327
 - 14.3.1 Grundlagen zu Guards 328
 - 14.3.2 Guards implementieren 328
 - 14.3.3 Guards verwenden..... 330
 - 14.3.4 Den BookMonkey erweitern 331
 - 14.4 Resolver: Asynchrone Daten vorladen..... 337
 - 14.4.1 Warum Resolver verwenden? 337
 - 14.4.2 Resolver aufsetzen 338
 - 14.4.3 Resolver in Routen verwenden 339
 - 14.4.4 Daten in einer Komponente abrufen 340
 - 14.4.5 Den BookMonkey erweitern 341

14.5	Routing: Wie geht's weiter?	347
14.5.1	Routenparameter asynchron laden	347
14.5.2	Mehrere RouterOutlets verwenden	348
14.5.3	Darstellung der URLs ändern: LocationStrategies	349
15	Internationalisierung: Iteration VII	351
15.1	i18n: Mehrere Sprachen und Kulturen anbieten	351
15.1.1	Was bedeutet Internationalisierung?	351
15.1.2	Nachrichten mit dem i18n-Attribut markieren	352
15.1.3	Nachrichten extrahieren und übersetzen	353
15.1.4	Feste IDs vergeben	354
15.1.5	JIT: Die App mit Übersetzungsdatei laden	354
15.1.6	AOT: Die App statisch mit Übersetzungsdatei bauen .	356
15.1.7	Den BookMonkey erweitern	357
16	Power Tipp: POEditor	363
17	Qualität fördern mit Softwaretests	369
17.1	Softwaretests	369
17.1.1	Arten von Tests	369
17.1.2	Test-Framework Jasmine	371
17.1.3	Test-Runner Karma	373
17.1.4	E2E-Test-Runner Protractor	373
17.2	Tests mit Karma	376
17.2.1	Die Testbibliothek von Angular	376
17.2.2	Isolierte Unit-Tests (Services und Pipes testen)	376
17.2.3	Shallow Unit-Tests: Einzelne Komponenten testen ...	380
17.2.4	Integrationstests: Mehrere Komponenten testen	382
17.2.5	Abhängigkeiten durch Stubs ersetzen	383
17.2.6	Abhängigkeiten durch Mocks ersetzen	386
17.2.7	HTTP-Requests testen	388
17.2.8	Komponenten mit Routen testen	391
17.2.9	Asynchronen Code testen	394
17.2.10	Fazit	396
17.3	Tests mit Protractor	397
17.3.1	Auf die Balance kommt es an	397
17.3.2	Protractor verwenden	398
17.3.3	Elemente selektieren: Locators	399
17.3.4	Aktionen durchführen	400
17.3.5	Asynchron mit Warteschlange	400
17.3.6	Redundanz durch Page Objects vermeiden	402
17.3.7	Eine Angular-Anwendung testen	403
17.3.8	Fazit	404

IV	Das Projekt ausliefern: Deployment	407
18	Das Projekt ausliefern: Deployment	409
18.1	Umgebungen konfigurieren	409
18.1.1	Umgebungen am Beispiel: BookMonkey	412
18.2	Produktivmodus aktivieren	414
18.3	Build erzeugen	414
18.4	Die Templates kompilieren	418
18.4.1	Just-In-Time-Kompilierung (JIT)	419
18.4.2	Ahead-Of-Time-Kompilierung (AOT)	420
18.5	Webserver konfigurieren und die App ausliefern	423
18.6	Ausblick: Automatisches Deployment	427
V	Weiterführende Themen	429
19	NativeScript: Mobile Anwendungen entwickeln	431
19.1	Mobile Apps entwickeln	431
19.2	Was ist NativeScript?	432
19.3	Warum NativeScript?	433
19.4	Hinter den Kulissen	434
19.5	Plattformspezifischer Code	435
19.6	Widgets und Layouts	437
19.7	Styling	438
19.8	NativeScript und Angular	439
19.9	Angular als Native App	440
19.10	Den BookMonkey mit NativeScript umsetzen	441
20	Power Tipp: Android-Emulator Genymotion	457
21	Redux: Den Application State verwalten	461
21.1	Was ist Redux?	462
21.2	Was sind Reducer?	463
21.3	Reducer verwalten den Anwendungsstatus	463
21.4	Actions	465
21.5	Angular-Services und Actions	467
21.6	Redux im Projekt einbinden	468
21.7	Components mit Redux verwenden	470
21.8	Container und Presentational Components	473
21.9	Weiterführendes	473
21.10	Den BookMonkey mit Redux umsetzen	475
22	Power Tipp: Redux DevTools	495

23	Wissenswertes	499
23.1	Plattformen und Renderer	499
23.2	Lifecycle-Hooks	500
23.3	Change Detection	503
23.4	Transklusion: Inhalt des Host-Elements verwenden	513
23.5	Eigenes Two-Way Binding	514
23.6	Else-Block für die ngIf-Direktive	516
23.7	Upgrade von Angular 1.x	519
24	Nachwort	525
	Anhang	527
A	Befehle der Angular CLI	529
B	Matcher von Karma	535
C	Abkürzungsverzeichnis	537
D	Linkliste	539
Index		545
Literaturverzeichnis		551