

AngularJS

Über die Autoren



Philipp Tarasiewicz ist im Web groß geworden und arbeitet als freiberuflicher Technologieberater, Autor, Sprecher und Coach. Seit einigen Jahren hat er sich auf den Bereich Enterprise JavaScript, insbesondere AngularJS, spezialisiert und unterstützt Unternehmen bei der Aus- und Fortbildung ihrer Mitarbeiter wie auch beim Ramp-up neuer Projekte. Gemeinsam mit Sascha Brink und Robin Böhm betreibt er das deutsche Portal zu AngularJS (AngularJS.DE).



Robin Böhm ist leidenschaftlicher Softwareentwickler, Berater und Autor im Bereich der Webtechnologien und speziell zu Enterprise JavaScript. Er beschäftigt sich seit einigen Jahren intensiv mit der Erstellung clientseitiger Webapplikationen und unterstützt Unternehmen sowohl bei der Aus- und Fortbildung von Mitarbeitern als auch bei der Umsetzung von Projekten. Außerdem ist er Mitgründer des Portals AngularJS.DE.

Philipp Tarasiewicz · Robin Böhm

AngularJS

**Eine praktische Einführung in das
JavaScript-Framework**



dpunkt.verlag

Philipp Tarasiewicz
philipp.tarasiewicz@angularjs.de

Robin Böhm
robin.boehm@angularjs.de

Lektorat: René Schönfeldt
Copy Editing: Annette Schwarz, Ditzingen
Satz: Da-TeX, Leipzig
Herstellung: Frank Heidt
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie 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 978-3-86490-154-6

1. Auflage 2014
Copyright © 2014 dpunkt.verlag GmbH
Wiebinger Weg 17
69123 Heidelberg

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.

5 4 3 2 1 0

*Philipp Tarasiewicz:
Für Thaddeus, Sylwia, Patrick und Laura*

*Robin Böhm:
Für Heike, Ernst, Kathi, Roswitha und Lisa*

Inhaltsverzeichnis

1	AngularJS Schnellstart	1
1.1	Zwei-Wege-Datenbindung: Boilerplate-Code war gestern	1
1.2	Direktiven: Eigene HTML-Elemente und Attribute	5
1.3	Filter: Formatierte Ausgaben im Handumdrehen	12
2	Grundlagen und Konzepte des Frameworks	19
2.1	Leitkonzepte	19
2.1.1	Model-View-Controller oder Model-View-ViewModel? ..	19
2.1.2	Die Zwei-Wege-Datenbindung und Scopes	23
2.1.3	Inversion of Control und Dependency Injection	25
2.1.4	Testbarkeit	27
2.2	Anwendungsbausteine	29
2.2.1	Module	29
2.2.2	Controller	30
2.2.3	Models	31
2.2.4	Routen	31
2.2.5	Ansichten, Templates und Expressions	33
2.2.6	Filter	36
2.2.7	Services	42
2.2.8	Direktiven	49
3	Das BookMonkey-Projekt	75
3.1	Auf geht's, ab geht's: Projekt- und Prozessvorstellung	75
3.2	Voraussetzungen	76
3.3	Die Projektumgebung aufsetzen	77
3.4	Projektstart: Detailansicht eines Buches	81
3.4.1	Das Template für die Detailansicht mit Expressions	83
3.4.2	Die ngHref-Direktive	84
3.4.3	Das Template mit der ngBind- und ngBindTemplate-Direktive	84
3.4.4	Das Anwendungsmodul definieren	86
3.4.5	Die erste Route	87
3.4.6	Ein eigener Scope mit dem BookDetailsCtrl-Controller ..	89
3.4.7	Der erste Test	90

3.5	Listenansicht für Bücher	98
3.5.1	Als Erstes der Test	99
3.5.2	Die Infrastruktur für die Listenansicht	104
3.5.3	Der BookListCtrl-Controller	105
3.5.4	Die ngRepeat-Direktive: Ausgabe eines Arrays im Template	105
3.5.5	Der orderBy-Filter: Sortierung festlegen	110
3.5.6	Der filter-Filter: Daten bequem filtern	115
3.6	Navigieren innerhalb der Anwendung	123
3.6.1	Die Standardroute mit \$routeProvider.otherwise()	123
3.6.2	Als Erstes der Test	124
3.6.3	Navigation mittels Hashbang-URLs	126
3.6.4	Die ngClick-Direktive: Auf Klick-Events reagieren	130
3.6.5	Der \$location-Service: Interaktionen mit der Adresszeile	131
3.7	Der erste Service	134
3.7.1	Als Erstes der Test	134
3.7.2	Der BookDataService: Datenzugriffe kapseln	141
3.7.3	Den BookDataService einbinden	146
3.7.4	Der \$routeParams-Service: URL-Parameter auslesen ...	148
4	Die Anwendung erweitern	153
4.1	Der Administrationsbereich	153
4.1.1	Als Erstes der Test	154
4.1.2	Die CRUD-Operationen des BookDataService	158
4.1.3	Die ngShow- und ngHide-Direktive: Inhalte bedingt anzeigen und ausblenden	162
4.1.4	Formularverarbeitung und Validierung mit dem Form- Controller und NgModelController	165
4.1.5	Templates mit der ngInclude-Direktive einbinden	178
4.1.6	Die Funktion zum Editieren eines Buches	181
4.1.7	Die Funktion zum Löschen eines Buches	182
4.2	Kategorisierung durch Tags	187
4.2.1	Das Datenmodell um Tags erweitern	188
4.2.2	Als Erstes der Test	190
4.2.3	Die Tokenfield-Direktive: Tags anlegen	198
4.2.4	Die Tags-Direktive: Tags anzeigen	207
4.3	Einen REST Web Service anbinden	211
4.3.1	Das BookMonkey-Backend	212
4.3.2	HTTP-Kommunikation mit dem \$http-Service	213
4.3.3	Als Erstes der Test	216
4.3.4	\$http im BookDataService nutzen	221
4.3.5	Die Anwendung wiederinstandsetzen	224

5	Projektverwaltung und Automatisierung	233
5.1	Node.js: Die Ablaufumgebung für die Werkzeuge	233
5.2	Frontend-Abhängigkeiten mit Bower verwalten	237
5.2.1	Bower konfigurieren	241
5.2.2	Eigene Pakete mit Bower verwalten	242
5.2.3	Ein privates Register erstellen	242
5.2.4	Mögliche Probleme mit Proxy-Servern	243
5.3	Aufgaben mit Grunt automatisieren	244
5.3.1	Aufgaben konfigurieren	245
5.3.2	Sinnvolle Pakete für die Entwicklung	255
5.4	Tests mit Karma automatisiert ausführen	259
5.4.1	Konfiguration	260
5.4.2	Die wichtigsten Parameter	261
5.4.3	Initiale Karma-Konfiguration generieren	263
5.4.4	Karma-Erweiterungen nutzen	265
5.4.5	Tests direkt in WebStorm ausführen	265
5.4.6	Test-Frameworks	267
5.4.7	Continuous Integration	268
5.5	Yeoman: Ein definierter Workflow	270
5.5.1	Was ist Yeoman?	270
5.5.2	Yeoman installieren	271
5.5.3	Anwendungsbausteinen generieren	271
5.5.4	Yo für AngularJS-Projekte	272
5.5.5	Generierte Grunt-Konfiguration	279
6	Debugging	283
6.1	Chrome Developer Tools	283
6.1.1	Der Elemente-Tab	284
6.1.2	Die Konsole	285
6.1.3	Der Sources-Tab: JavaScript-Code debuggen	287
6.2	Batarang: Einsicht in die laufende AngularJS-Anwendung	289
6.2.1	Scopes untersuchen	290
6.2.2	Ausführungszeiten von Funktionen vermessen	292
6.2.3	Serviceabhängigkeiten untersuchen	293
6.3	Die WebStorm-IDE	295
7	Antworten auf häufig gestellte Fragen	299
7.1	AngularJS-Module: Wie strukturieren wir Anwendungen mit Modulen?	299
7.1.1	Module in AngularJS	299
7.1.2	Wann ist Modularisierung sinnvoll?	301
7.1.3	Ordnerstruktur	304

- 7.2 Promises: Wie gehen wir mit Asynchronität um? 306
 - 7.2.1 Asynchronität und nichtblockierende Aufrufe 306
 - 7.2.2 Was sind Promises? 307
 - 7.2.3 Promises in AngularJS 311
 - 7.2.4 Promises in AngularJS testen 315
- 7.3 AngularJS und RequireJS: Ist diese Kombination sinnvoll? 318
 - 7.3.1 Was ist RequireJS? 318
 - 7.3.2 Ein Beispiel mit RequireJS..... 319
 - 7.3.3 AngularJS und RequireJS 322
 - 7.3.4 Testen mit RequireJS 327
 - 7.3.5 Die Antwort 331
- 7.4 Mobile: Unterstützt AngularJS mobile Endgeräte? 333
 - 7.4.1 Touch-Events unterstützen 334
 - 7.4.2 Die Swipe-Direktiven 335
 - 7.4.3 Der \$swipe-Service 336
- Index 341**