



C o m m u n i t y E x p e r i e n c e D i s t i l l e d

Learning AngularJS for .NET Developers

Build single-page web applications using frameworks that help you work efficiently and deliver great results

Alex Pop

[PACKT] open source*
PUBLISHING community experience distilled

Learning AngularJS for .NET Developers

Build single-page web applications using frameworks that help you work efficiently and deliver great results

Alex Pop

[PACKT] open source 
PUBLISHING
community experience distilled
BIRMINGHAM - MUMBAI

Learning AngularJS for .NET Developers

Copyright © 2014 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: July 2014

Production reference: 1180714

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78398-660-6

www.packtpub.com

Cover image by Rajnish Jha (rajnishlalitjha@gmail.com)

Credits

Author

Alex Pop

Project Coordinator

Neha Bhatnagar

Reviewers

Andrew C Cunliffe

Anton Kropp

Pascal Precht

Proofreaders

Maria Gould

Ameesha Green

Commissioning Editor

Usha Iyer

Indexers

Hemangini Bari

Tejal Soni

Acquisition Editor

Richard Brookes-Bland

Graphics

Valentina D'silva

Content Development Editor

Arvind Koul

Production Coordinator

Conidon Miranda

Technical Editor

Sebastian Michael Rodrigues

Cover Work

Conidon Miranda

Copy Editors

Insiya Morbiwala

Aditya Nair

About the Author

Alex Pop is a professional software developer with a university degree in Computer Engineering and 12 years of commercial experience building .NET applications. He has worked for ISVs, building enterprise resource planning applications, content management systems, and insurance and financial software products, and he is currently working in the higher education sector as a web application developer. His developer blog at <http://alexvpop.blogspot.co.uk> contains technical articles about .NET, JavaScript, and various software engineering topics.

I would like to salute the efforts of the open source developer community at large that made this book possible. I would like to thank my ever-so-understanding wife and daughter who patiently supported me in writing this book.

About the Reviewers

Andrew C Cunliffe is a technical consultant; he focuses on projects that use open source technologies and process improvements for development teams of all sizes. He delivers integrated, end-to-end solutions through the use of continuous delivery techniques, with more than 15 years of experience writing software for corporate companies, ranging from company tools such as intranets to large bespoke solutions used by governments and banks. Andrew has worked with a large number of software languages, predominantly JavaScript (AngularJS and Node.js), Java, and PHP. He always ensures the highest levels of quality through test-driven techniques. In his limited free time, he contributes to projects on GitHub and reviews books for publications. Andrew is the owner of SYSEN Ltd, where he holds the position of Director; SYSEN offers professional services in the software development and support landscape. The company has delivered solutions for a number of businesses; for more information, see the website, www.sysen.co.uk.

I would not have been able to do all of this without the ongoing support of my family and friends, and as such, I would like to thank them all, especially my wife and our three children.

Anton Kropp is an ex-Amazon engineer, who is now Senior Software Engineer at Practice Fusion, where he works on web-based electronic health records. He has extensive experience with F#, C#, AngularJS, TypeScript, realtime video, and OCR, and he also runs the blog www.onoffswitch.net. Anton is currently based in Seattle and spends his free time riding his bicycle and drinking local beer.

Pascal Precht is a frontend engineer and has over 7 years of professional experience in web development. He loves semantic markup, evolving technologies, and the open Web. Pascal is very active in the community and has contributed to many open source projects, such as AngularJS and AngularUI, to name a few. He is also the creator of the popular angular-translate module for the AngularJS framework. Pascal has also reviewed *ng-book*, *Ari Lerner*, *Fullstack.io*, which is one of the most-read books about AngularJS. When he's not busy preparing for the next workshop or conference talk, you might find him outside with his skateboard.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Introducing AngularJS	9
Presenting AngularJS with examples	10
A jQuery example	10
An AngularJS example	13
Introducing the AngularJS controller	16
An overview of the AngularJS architecture	20
The Model-View-Controller pattern	20
The structure of an AngularJS application	21
Bootstrapping an AngularJS application	21
Defining module dependencies	23
The JavaScript patterns and practices used in AngularJS applications	24
Immediately-invoked Function Expression	24
The revealing module pattern	25
The strict mode of JavaScript	27
Dependency injection	27
Introducing AngularJS services	29
Directives	34
The ngRepeat directive	34
The ngInclude directive	36
Creating a custom directive	37
Filters	40
Summary	43

Chapter 2: Creating an AngularJS Client-side Application in Visual Studio	45
Using Visual Studio with NuGet to manage AngularJS and related libraries	47
Introducing NuGet	49
Using Visual Studio extensions for AngularJS	52
Prototyping client-side components	54
Organizing client-side code	56
Integrating AngularJS with JavaScript libraries	58
Native JavaScript libraries and AngularJS	59
Third-party user interface libraries and AngularJS	60
Routing	60
Creating a bicycle rental management application	62
Creating the application layout	63
Implementing the Bicycles views	65
The Bicycles index view	65
The Bicycles new view	66
The Bicycles edit view	70
Implementing the Customers views	72
Implementing the Rentals views	73
Summary	74
Chapter 3: Creating .NET Web Services for AngularJS	75
RESTful web services	76
Choosing a RESTful web services framework	78
Why ServiceStack is a great fit for AngularJS	79
ServiceStack overview	80
Setting up the ServiceStack starter project	80
Creating ServiceStack web services	82
Using the ServiceStack C# client	84
The ServiceStack infrastructure	84
Routing	85
Request and response filters	86
Dependency injection	87
Automatic validation	88
Creating web service methods for the sample application	91
ServiceStack project structure	91
Implementing Bicycles-related web services	92
The GET web service methods	92
The POST web service method	93
The PUT web service method	96
Merging PUT and POST request DTOs	97

Calling RESTful web services in AngularJS using the ngResource module	98
Using the \$resource service	98
Resource class methods	99
Resource instance methods	101
Custom resource methods	102
Summary	102
Chapter 4: Creating an AngularJS, ASP.NET MVC, ServiceStack Application	103
Using ASP.NET MVC as the application backbone	104
Why you should use ASP.NET MVC	104
Adding ASP.NET MVC to the main sample application	105
Integrating ASP.NET MVC with ServiceStack	106
Routing considerations for ASP.NET MVC and AngularJS	107
Using ASP.NET bundling and minification features for AngularJS files	108
Securing the AngularJS application	109
Using the ServiceStack.Mvc library	110
The building blocks of the ServiceStack security infrastructure	112
Introducing the AuthFeature plugin	112
Implementing authentication for web services	114
Implementing authentication for ASP.NET MVC controllers	115
Pushing initial data from the server-side application to the AngularJS application	118
How to ensure that AngularJS works well with the backend	120
Handling the waiting time for long operations	120
Handling exceptions and implementing generic animations	122
Adding a \$http interceptor	122
Implementing spinner animations for all web service calls	124
Adding database persistence	125
Adding database persistence to the AuthFeature plugin	126
Adding database persistence for web services	128
Persisting the Bicycle class	129
Changing BicycleRepository to use the OrmLite API	131
Summary	132
Chapter 5: Testing and Debugging AngularJS Applications	133
Overview of testing AngularJS applications	134
Introducing Node.js	135
Setting up the Node.js environment	136

Unit testing AngularJS components	138
Installing Karma	139
Configuring Karma	140
Creating AngularJS unit tests with Jasmine	141
Jasmine essentials	142
Using the ngMock module for the first test	143
Mocking remote server calls with \$httpBackend	145
Testing an AngularJS application end to end	146
Installing Protractor	146
Configuring Protractor	147
Writing Protractor tests	148
Testing web service endpoints	150
Configuring self-hosted ServiceStack web services	151
Adding BDD tests with xBehave.net	153
Unit testing .NET code	156
Techniques to debug AngularJS applications	158
Support for HTML5 validation and older browsers such as Internet Explorer 8 and 9	160
Summary	162
Chapter 6: Advanced AngularJS Topics	163
Internationalization and localization	164
Using the ngLocale module	164
Using the angular-dynamic-locale module	166
Other internationalization-related topics	168
Using AngularJS animations	168
Working with remote web services	170
Using JSONP	171
Using CORS	173
Template caching	176
Summary	178
Index	179

Preface

In the last couple of years, the web development landscape has changed dramatically following the mainstream adoption of smartphones and tablets. Mobile web traffic has increased substantially in 2013 alone, reaching almost a third of the total web traffic and the increase is expected to continue in the near future. Websites are now expected to work on a wide range of devices and form factors and they have to support an unprecedented diversity of browser clients. Web applications based on browser plugins such as Java, Adobe Flash, and Silverlight have become problematic as these plugins are not available or not supported on all devices.

Parallel with these changes, all major browser vendors have accelerated the work to define and implement web standards such as HTML5 and CSS3 and JavaScript standards such as ECMAScript 5.1 and the upcoming ECMAScript 6. Although not all these standards are final or complete, a succession of browser releases from all vendors have continuously improved standards support and the performance of client-side content and code. The wider development community also contributed to the standardization effort by providing various tools and libraries. These have extended web standards support for older browsers and unified different browser implementations of similar features under a common programming model.

A new web development ecosystem based on open standards, tools, and libraries emerged. It was used as a solid base to build modern web applications that embrace HTML, CSS, and JavaScript rather than abstracting them away. It also allowed for applications based on browser plugins to be gradually replaced by new versions based on HTML, CSS, and JavaScript. The focus on leveraging client-browser capabilities for rendering and manipulating content led to the adoption of a new type of application called a single-page application. It was made popular by the Google Mail and Google Maps web applications, which were very responsive, easy to use, and had a look and feel similar to desktop applications.

A single-page application typically has initial resources and content generated server side on the first application request, and for all subsequent requests, it will use client-side code to load any required resources and display content. A single-page application can also have one or more static pages that serve as physical entry points. Microsoft and Google made this type of application a first-class citizen in the software development kits for their operating systems in the form of Windows Store apps and Chrome apps. Cross-platform mobile development has also embraced HTML, CSS, and JavaScript in frameworks such as PhoneGap/ Apache Cordova. Single-page applications have transitioned from consumer-focused applications to enterprise-focused applications in development models like the ones introduced by Microsoft with apps for Office and apps for SharePoint in Office 2013 and SharePoint 2013.

The building of modern web applications that include single-page applications involves the adoption of development approaches such as responsive web design (when the application adapts seamlessly to the browser-client resolution and orientation) and progressive enhancement (when the application provides different content based on the browser-client capabilities and the available network connection). A need for rapid prototyping, rich client features, and the reality of having to maintain large JavaScript code bases are other aspects that make the development process challenging and costly when compared with typical desktop application development.

The rise and mass adoption of jQuery and its plugin ecosystem after 2006 addressed the problem of rich client features. The increasingly complex JavaScript code bases remained a challenge that started to be addressed by the adoption of test frameworks, such as QUnit in 2008 and Jasmine in 2010, and new application frameworks, such as Backbone.js and Knockout.js in 2010. The application frameworks were implementing derivatives of a software design pattern common in server-side web application frameworks such as ASP.NET MVC: the Model-View-Controller (MVC) pattern. This pattern makes a clear distinction between components such as application data and business rules represented by the Model and the presentation of the application data in any shape or format represented by the View. The Controller component sits in the middle, passing data to or from the Model and updating the View or responding to View changes that are passed to the Model. This abstraction brings benefits like better code reuse (the same Model can be used by multiple Views), testability (the Controller can be tested in isolation from the View), and a more efficient development workflow (for example, while the developer works on a Controller, the designer can modify the View).

In the same timeframe, another JavaScript framework called AngularJS was being developed at Google. This framework was also based on the MVC pattern, and its strong points were that it could enhance and extend HTML by adding new elements and declarative behavior, and it could be tested from the start using modular, composable code units. With the 1.0 release in 2012, and since 2013, its popularity has grown more and more, with the high points being the 1.2 release in late 2013 and its first conference, called ng-conf, in January 2014.

Reflecting the popularity of single-page applications, Microsoft has started to provide support for this new application paradigm with Visual Studio 2012. With Visual Studio 2013, this support was enhanced, and a couple of recent Visual Studio extensions provide excellent support for AngularJS. Now is a great time to learn AngularJS if you are a .NET developer.

The main focus of this book is on how to rapidly prototype and build modern web applications with AngularJS in the context of .NET development tools and frameworks. The book assumes the reader has already built websites, web applications, or web services using Microsoft technologies such as ASP.NET, ASP.NET MVC, Silverlight, or WCF and knows the fundamentals of HTML5, CSS3, JavaScript, jQuery, and ASP.NET MVC.

What this book covers

Chapter 1, Introducing AngularJS, teaches the reader how to manipulate HTML using AngularJS. The reader will also learn about the core parts of AngularJS. A series of examples will gradually showcase the main AngularJS features. After a high-level overview of the AngularJS architecture, a more complex example will be shown, allowing the reader to see how data binding works and how to respond to user events. The chapter will end with an overview of the JavaScript patterns that are essential for AngularJS applications, which will be followed by an overview of AngularJS building blocks.

Chapter 2, Creating an AngularJS Client-side Application in Visual Studio, teaches the reader how to use Visual Studio to build an AngularJS application. A walkthrough to set up the development environment using the AngularJS NuGet packages will be followed by a presentation of the workflow involved in building an AngularJS application and some of the best practices in organizing the project structure. The final topic will cover building a complex AngularJS application.

Chapter 3, Creating .NET Web Services for AngularJS, teaches the reader about RESTful web services, how to use a .NET web service framework that is a great fit for AngularJS, and how to create web service resources for AngularJS. A brief presentation of the REST principles will be followed by a discussion on why RESTful web services work very well with AngularJS. An overview of the .NET web services framework, ServiceStack, will evolve into a walkthrough to create the required web service resources that will be used in the AngularJS example from the previous chapter.

Chapter 4, Creating an AngularJS, ASP.NET MVC, ServiceStack Application, teaches the reader how to integrate an AngularJS application with ASP.NET MVC and ServiceStack and how to create a production-ready application. An overview of ASP.NET MVC will be followed by a discussion about when you need to combine ASP.NET MVC and AngularJS. A walkthrough of how to set up routing with ASP.NET MVC and how to secure AngularJS applications will be next and the chapter will conclude with presenting additional tasks needed for AngularJS and backend integration.

Chapter 5, Testing and Debugging AngularJS Applications, teaches the reader how to test and debug AngularJS components, test web services, and how to implement end-to-end testing. The next topics are a discussion about why it is important to test web service endpoints, how to test them, and what additional tasks are needed for IE 8 support.

Chapter 6, Advanced AngularJS Topics, teaches the reader about advanced topics such as internationalization, animations, JSONP and CORS considerations, and template caching.

What you need for this book

The minimum software requirement is Visual Studio 2013 Express for Web and the recommended one is Visual Studio 2013 Professional. For testing and debugging AngularJS applications, you also need to install Node.js, Node.js Tools for Visual Studio, and Java SE Development Kit 7 (JDK 7).

Who this book is for

The book is targeted at .NET developers who have already built web applications or web services and have a fundamental knowledge of HTML, JavaScript, and CSS.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "It will match some regular HTML tags such as `input` and `select` if they contain ng- prefixed attributes or CSS classes."

A block of code is set as follows:

```
var myAppModule = angular.module('myApp', []);
myAppModule.controller('ExampleController', function ($scope) {
  $scope.name = "Alex Pop";
  $scope.previousName = "";
  $scope.onNameFocused = function() {
    $scope.previousName = $scope.name;
  };
});
```


When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:


```
var myAppModule = angular.module('myApp', []);
myAppModule.controller('ExampleController', ['$scope', function
($scope) {
  $scope.name = "Alex Pop";
  $scope.previousName = "";
  $scope.onNameFocused = function() {
    $scope.previousName = $scope.name;
  };
}]);
```

Any command-line input or output is written as follows:

```
Install-Package AngularJS.Core -Version 1.2.15
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "The main menu for our application will contain three links to the three main views: **Rentals**, **Customers**, and **Bicycles**."

[ Warnings or important notes appear in a box like this.]

[ Tips and tricks appear like this.]

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or the code – we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

