BUILD YOUR OWN ANGULARJS

Build Your Own AngularJS

ISBN 978-952-93-3544-2

This book is written for the working programmer, who either:

- Wants to learn AngularJS.
- Already knows AngularJS but wants to take their knowledge of its inner workings to the next level.
- Wants to get an idea of how a substantial JavaScript application framework can be built.

AngularJS is not a small framework. It has a large surface area with many new concepts to grasp. Its codebase is also substantial, with 35K lines of JavaScript in it. While all of those new concepts and all of those lines of code give you powerful tools to build the apps you need, they also come with a steep learning curve.

I hate working with technologies I don't quite understand. Too often, it leads to code that just happens to work, not because you truly understand what it does, but because you went through a lot of trial and error to make it work. Code like that is difficult to change and debug. You can't reason your way through problems. You just poke at the code until it all seems to align.

Frameworks like AngularJS, powerful as they are, are prone to this kind of code. Do you understand how Angular does dependency injection? Do you know the mechanics of scope inheritance? What exactly happens during directive transclusion? When you don't know how these things work, as I didn't when I started working with Angular, you just have to go by what the documentation tells you or what people have said on Stack Overflow. When that isn't enough, you try different things until you get the results you need.

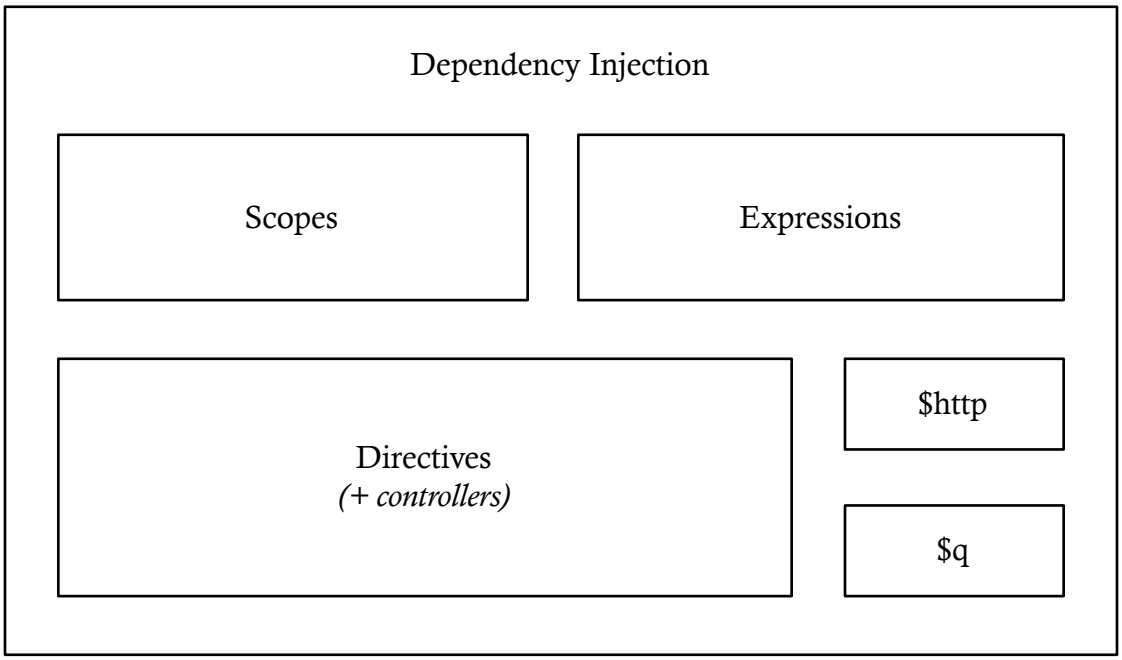
The thing is, while there's a lot of code in AngularJS, it's all *just regular JavaScript code*. It's no different from the code in your applications. Most of it is well-factored, readable code. You can study it to learn how Angular does what it does. When you've done that, you're much better equipped to deal with the issues you face in your daily application development work. You'll know not only what features Angular provides to solve a particular problem, but also how those features work, how to get the most out of them, and where they fall short.

The purpose of this book is to help you demystify the inner workings of AngularJS. To take it apart and put it back together again, in order to truly understand how it works.

A true craftsman knows their tools well. So well that they could in fact make their own tools if needed. This book will help you get there with AngularJS.

How To Read This Book

During the course of the book we will be building an implementation of AngularJS. We'll start from a completely blank slate, and then in each chapter extend the implementation with new capabilities.



In "Part 1" we begin by implementing the *dirty-checking* and *eventing* features provided by Angular Scopes. We'll get intimately familiar with the inner workings of `$watch`, `$watchCollection`, `$digest`, `$apply`, `$evalAsync`, `$emit`, `$broadcast` et al.

In "Part 2", we extend our implementation by adding support for *expressions* - the strings we put inside `{{curly braces}}` in templates and occasionally use with `$watch`. This seemingly simple feature actually involves a deep and exciting journey into the world of programming language parsing, lexing, and abstract syntax trees. In this part we also look at *filters*, a feature that is intimately integrated with expressions.

In "Part 3", we dive into *dependency injection*. Here we'll get to know how exactly Angular *providers*, *services*, and *factories* are implemented and how Angular *modules* work. We'll also retrofit the code from parts 1 and 2 to the DI system, so that we get something that resembles an integrated framework.

In "Part 4" we write a couple of utilities that are central to almost every Angular application: The *promises* implementation in `$q` and the *HTTP client* implementation in `$http`. These are not only useful for application developers to understand, but also a key building block for directives.

In the final part, "Part 5", we finally get into *directives*. This is the most advanced part of the book, as it implements the most feature-rich and complex subsystem of AngularJS: The directive compiler. We'll get acquainted with DOM *compilation* and *linking*, the `Attributes` object, *isolate scopes*, *controllers*, *transclusion*, and *interpolation*. In the final chapter, we'll put everything together and see how Angular actually *bootstraps and runs* an application.

While there are certain areas of functionality in Angular that are largely independent, most of the

code you'll be writing builds on things implemented in previous chapters. That is why a sequential reading will help you get the most out of this book.

The format of the book is simple: Each feature is introduced by discussing what it is and why it's needed. We will then proceed to implement the feature following test-driven development practices: By writing failing tests and then writing the code to make them pass. As a result, we will produce not only the framework code, but also a test suite for it.

It is highly encouraged that you not only read the code, but also actually type it in and build your own Angular while reading the book. To really make sure you've grasped a concept, poke it from different directions: Write additional test cases. Try to intentionally break it in a few ways. Refactor the code to match your own style while keeping the tests passing.

If you're only interested in certain parts of the framework, feel free to skip to the chapters that interest you. While you may need to reference back occasionally, you should be able to poach the best bits of Angular to your own application or framework with little difficulty.

Source Code

The source code and test suite implemented in this book can be found on GitHub, at https://github.com/teropa/build-your-own-angularjs/.

To make following along easier, commits in the repository are ordered to match the order of events in the book. Note that this means that during the production of the book, the history of the code repository may change as revisions are made.

There is also a Git tag for each chapter, pointing to the state of the codebase at the end of that chapter. You can download archives of the code corresponding to these tags from https://github.com/teropa/build-your-own-angularjs/releases.

Contributors

Since I released the first prerelease of the book in early 2014, hundreds of GitHub issues have been submitted by a great number of people, each one pointing out a bug, typo, or improvement idea for the book. These have been immeasurably helpful in getting the book to the state it is in now, and I want to thank everyone who has submitted feedback.

I would like to especially thank the following people for their valuable ideas and help during the writing of this book:

- Iftach Bar
- Xi Chen
- Wil Pannell

- Pavel Pomyerantsyev
- Mauricio Poppe
- Mika Ristimäki
- Jesus Rodriguez
- Scott Silvi

Errata & Contributing

Your feedback about corretions and improvement ideas is more than welcome. If you come across errors, or just feel like something could be improved, please file an issue to the book's Errata on GitHub: https://github.com/teropa/build-your-own-angularjs/issues. To make this process easier, there's also a link to the issue tracker on the footer of each page.

Contact

Feel free to get in touch with me by sending an email to tero@teropa.info or tweeting at @teropa.

Table of Contents

How To Read This Book 3
Source Code 5
Contributors 5
Errata & Contributing 6
Contact 6
Version History 15
Setting up 19
Install Node and NPM 20
Create The Project Directories 20
Create package.json for NPM 20
“Hello, World!” 21
Enable Static Analysis With JSHint 21
Enable Unit Testing With Jasmine, Sinon, and Karma 23
Integrate Browserify 25
Include Lo-Dash And jQuery 27
Summary 28
Scopes 29
Scopes and Dirty-Checking 31
Scope Objects 32
Watching Object Properties: $watch And $digest 33
Checking for Dirty Values 35
Initializing Watch Values 38
Getting Notified Of Digests 39
Keeping The Digest Going While It Stays Dirty 40
Giving Up On An Unstable Digest 42
Short-Circuiting The Digest When The Last Watch Is Clean 44
Value-Based Dirty-Checking 48
NaNs 50
Handling Exceptions 51
Destroying A Watch 53
Summary 59
Scope Methods 60
$eval - Evaluating Code In The Context of A Scope 61
$apply - Integrating External Code With The Digest Cycle 62
$evalAsync - Deferred Execution 63
Scheduling $evalAsync from Watch Functions 66
Scope Phases 68
Coalescing $apply Invocations - $applyAsync 72
Running Code After A Digest - $$postDigest 78
Handling Exceptions 80
Watching Several Changes With One Listener: $watchGroup 83
Summary 92

Scope Inheritance 93
The Root Scope 94
Making A Child Scope 94
Attribute Shadowing 98
Separated Watches 99
Recursive Digestion 100
Digesting The Whole Tree from $apply, $evalAsync, and $applyAsync 104
Isolated Scopes 108
Substituting The Parent Scope 115
Destroying Scopes 117
Summary 119
Watching Collections 120
Setting Up The Infrastructure 122
Detecting Non-Collection Changes 123
Detecting New Arrays 127
Detecting New Or Removed Items in Arrays 129
Detecting Replaced or Reordered Items in Arrays 130
Array-Like Objects 133
Detecting New Objects 136
Detecting New Or Replaced Attributes in Objects 138
Detecting Removed Attributes in Objects 141
Preventing Unnecessary Object Iteration 143
Dealing with Objects that Have A length 145
Handing The Old Collection Value To Listeners 146
Summary 150
Scope Events 151
Publish-Subscribe Messaging 152
Setup 153
Registering Event Listeners: $on 153
The basics of $emit and $broadcast 155
Dealing with Duplication 157
Event Objects 158
Additional Listener Arguments 159
Returning The Event Object 160
Deregistering Event Listeners 161
Emitting Up The Scope Hierarchy 163
Broadcasting Down The Scope Hierarchy 165
Including The Current And Target Scopes in The Event Object 166
Stopping Event Propagation 171
Preventing Default Event Behavior 172
Broadcasting Scope Removal 174
Disabling Listeners On Destroyed Scopes 175
Handling Exceptions 176
Summary 177

Expressions and Filters 178
A Whole New Language 179
What We Will Skip 180
Literal Expressions 182
Setup 183
Parsing Integers 187
Parsing Floating Point Numbers 194
Parsing Scientific Notation 196
Parsing Strings 199
Parsing true, false, and null 207
Parsing Whitespace 210
Parsing Arrays 211
Parsing Objects 218
Summary 226
Lookup and Function Call Expressions 227
Simple Attribute Lookup 228
Parsing this 233
Non-Computed Attribute Lookup 233
Locals 238
Computed Attribute Lookup 241
Function Calls 246
Method Calls 250
Assigning Values 254
Ensuring Safety In Member Access 260
Ensuring Safe Objects 265
Ensuring Safe Functions 272
Summary 275
Operator Expressions 276
Unary Operators 277
Multiplicative Operators 285
Additive Operators 288
Relational And Equality Operators 290
Logical Operators AND and OR 295
The Ternary Operator 299
Altering The Precedence Order with Parentheses 301
Statements 303
Summary 304
Filters 306
Filter Registration 308
Filter Expressions 310
Filter Chain Expressions 318
Additional Filter Arguments 319
The Filter Filter 320

Filtering With Predicate Functions 322
Filtering With Strings 323
Filtering With Other Primitives 327
Negated Filtering With Strings 330
Filtering With Object Criteria 331
Filtering With Object Wildcards 338
Filtering With Custom Comparators 342
Summary 345
Watching Expressions 346
Integrating Expressions to Scopes 347
Literal And Constant Expressions 350
Optimizing Constant Expression Watching 361
One-Time Expressions 364
Input Tracking 370
Stateful Filters 387
External Assignment 389
Summary 394
Modules and Dependency Injection 396
Modules and The Injector 398
The angular Global 399
Initializing The Global Just Once 400
The module Method 401
Registering A Module 402
Getting A Registered Module 404
The Injector 406
Registering A Constant 407
Requiring Other Modules 412
Dependency Injection 414
Rejecting Non-String DI Tokens 416
Binding this in Injected Functions 417
Providing Locals to Injected Functions 418
Array-Style Dependency Annotation 419
Dependency Annotation from Function Arguments 421
Strict Mode 426
Integrating Annotation with Invocation 428
Instantiating Objects with Dependency Injection 429
Summary 432
Providers 434
The Simplest Possible Provider: An Object with A $get Method 435
Injecting Dependencies To The $get Method 437
Lazy Instantiation of Dependencies 438
Making Sure Everything Is A Singleton 441
Circular Dependencies 441

Provider Constructors 445
Two Injectors: The Provider Injector and The Instance Injector 447
Unshifting Constants in The Invoke Queue 457
Summary 458
High-Level Dependency Injection Features 460
Injecting The $injectors 461
Injecting $provide 463
Config Blocks 464
Run Blocks 469
Function Modules 472
Hash Keys And Hash Maps 475
Function Modules Redux 482
Factories 483
Values 486
Services 489
Decorators 492
Integrating Scopes, Expressions, and Filters with The Injector 496
Making a Configurable Provider: Digest TTL 516
Summary 519
Utilities 520
Promises 522
Promises 523
Promise Implementations 524
Promises in AngularJS 525
Further Reading 525
The $q Provider 525
Creating Deferreds 526
Accessing The Promise of A Deferred 528
Resolving A Deferred 529
Preventing Multiple Resolutions 532
Ensuring that Callbacks Get Invoked 533
Registering Multiple Promise Callbacks 534
Rejecting Deferreds And Catching Rejections 536
Cleaning Up At The End: finally 540
Promise Chaining 541
Exception Handling 545
Callbacks Returning Promises 546
Chaining Handlers on finally 548
Notifying Progress 555
Immediate Rejection - $q.reject 561
Immediate Resolution - $q.when 562
Working with Promise Collections - $q.all 565
ES2015-Style Promises 571

Promises Without $digest Integration: $$q 574
Summary 582
HTTP 584
What We Will Skip 585
The Providers 585
Sending HTTP Requests 587
Default Request Configuration 595
Request Headers 596
Response Headers 606
Allow CORS Authorization: withCredentials 609
Request Transforms 611
Response Transforms 616
JSON Serialization And Parsing 622
URL Parameters 629
Shorthand Methods 645
Interceptors 648
Promise Extensions 659
Request Timeouts 661
Pending Requests 665
Integrating $http and $applyAsync 666
Summary 668
Directives 671
DOM Compilation and Basic Directives 674
Creating The $compile Provider 675
Registering Directives 676
Compiling The DOM with Element Directives 681
Recursing to Child Elements 688
Using Prefixes with Element Directives 690
Applying Directives to Attributes 691
Applying Directives to Classes 695
Applying Directives to Comments 696
Restricting Directive Application 699
Prioritizing Directives 703
Terminating Compilation 708
Applying Directives Across Multiple Nodes 712
Summary 719
Directive Attributes 720
Passing Attributes to the compile Function 721
Introducing A Test Helper 724
Handling Boolean Attributes 725
Overriding attributes with ng-attr 727
Setting Attributes 728
Setting Boolean Properties 732

Denormalizing Attribute Names for The DOM 733
Observing Attributes 737
Providing Class Directives As Attributes 742
Adding Comment Directives As Attributes 746
Manipulating Classes 747
Summary 749
Directive Linking and Scopes 751
The Public Link Function 752
Directive Link Functions 754
Plain Directive Link Functions 761
Linking Child Nodes 762
Pre- And Post-Linking 765
Keeping The Node List Stable for Linking 770
Linking Directives Across Multiple Nodes 771
Linking And Scope Inheritance 774
Isolate Scopes 779
Isolate Attribute Bindings 786
One-Way Data Binding 791
Two-Way Data Binding 797
Expression Binding 807
Summary 811
Controllers 813
The $controller provider 814
Controller Instantiation 815
Controller Registration 818
Global Controller Lookup 821
Directive Controllers 822
Locals in Directive Controllers 827
Attaching Directive Controllers on The Scope 828
Controllers on Isolate Scope Directives 830
Requiring Controllers 846
Requiring Multiple Controllers 850
Requiring Multiple Controllers as an Object 851
Self-Requiring Directives 854
Requiring Controllers in Multi-Element Directives 855
Requiring Controllers from Parent Elements 856
Accessing Required Controllers from The Directive Controller 867
The ngController Directive 869
Attaching Controllers on The Scope 872
Looking Up A Controller Constructor from The Scope 873
Summary 874
Directive Templates 876
What We Will Skip 877
Basic Templating 877

Disallowing More Than One Template Directive Per Element 880
Template Functions 881
Isolate Scope Directives with Templates 882
Asynchronous Templates: templateUrl 883
Template URL Functions 893
Disallowing More Than One Template URL Directive Per Element 894
Linking Asynchronous Directives 897
Linking Directives that Were Compiled Earlier 904
Preserving The Isolate Scope Directive 906
Preserving Controller Directives 909
Summary 911
Directive Transclusion 912
Basic Transclusion 914
Transclusion And Scopes 920
Transclusion from Descendant Nodes 929
Transclusion in Controllers 935
The Clone Attach Function 936
Transclusion with Template URLs 943
Transclusion with Multi-Element Directives 947
The ngTransclude Directive 948
Full Element Transclusion 951
Requiring Controllers from Transcluded Directives 965
Summary 968
Interpolation 970
The $interpolate service 971
Interpolating Strings 973
Value Stringification 978
Supporting Escaped Interpolation Symbols 981
Skipping Interpolation When There Are No Expressions 982
Text Node Interpolation 984
Attribute Interpolation 990
Optimizing Interpolation Watches With A Watch Delegate 1000
Making Interpolation Symbols Configurable 1006
Summary 1012
Components 1014
Registering Components 1016
Basic Components 1017
Component Scopes and Bindings 1019
Component Templates 1024
Component Transclusion 1029
Requiring from Components 1030
The $onInit Lifecycle Hook 1031
The $onDestroy Lifecycle Hook 1033
The $postLink Lifecycle Hook 1034

The $onChanges Hook 1036
Summary 1053
Bootstrapping 1056
The ngClick Directive 1057
Bootstrapping Angular Applications Manually 1061
Bootstrapping Angular Applications Automatically 1068
Building The Production Bundle 1073
Running An Example App 1075
Summary 1077

Version History

2016-05-30: Chapter 23: Components

Added the chapter for Angular 1.5 components. Fixed a few errata.

2016-05-15: Added One-way Bindings and Require as object

Shipping the first Angular 1.5 features

2016-05-01: Production Release

Added epub and mobi formats. New typeset and layout. Revised preface. A large number of other improvements here and there.

2015-11-05: Chapter 22: Bootstrapping Angular

Added the final chapter. Migrated the project code to Browserify, and removed the dependency to Grunt in favor of plain NPM build scripts. Switched test runner back to Karma.

Also fixed a number of errata.

2015-08-24: Maintenance Release

Added a number of Angular 1.4 updates and fixed errata.

2015-07-31: Chapter 8: Filters

Added Chapter 8 and added filter support to Chapters 9 and 12.

2015-07-23: Expression Parser Rewrite

Revisited Part 2 of the book to cover the Angular 1.4+ expression parser.

2015-06-15: Chapter 21: Interpolation

Added Chapter 21.

2015-06-08: Chapter 20: Directive Transclusion

Added Chapter 20 and fixed a few errata.

2015-04-27: Chapter 19: Directive Templates

Added Chapter 19 and fixed a few errata.

2015-04-05: Chapter 14: $http

Added Chapter 14 and fixed a number of errata.

2015-02-23: Chapter 13: Promises

Added Chapter 13.

2015-01-03: Chapter 16: Controllers

Added Chapter 16 and fixed a number of errata.

2014-12-24: Chapter 15: Directive Linking And Scopes

Added Chapter 15.

2014-11-16: Angular 1.3 updates

Added coverage of several smaller changes in Angular 1.3 across all chapters. Fixed a few errata.

2014-10-26: Added $applyAsync

Added coverage of Angular 1.3 `$applyAsync` to Chapters 1 and 2. Fixed a few errata.

2014-10-12: Added One-Time Binding and Input Tracking

Reworked the Expressions part of the book for Angular 1.3, with its new features - both internal and user-facing. Fixed a few errata.

2014-09-21: Added $watchGroup

Coverage of the new Angular 1.3 `$watchGroup` feature in Chapter 1 and fixed a number of errata.

2014-08-16: Chapter 13: Directive Attributes

Added Chapter 13.

2014-08-09: Maintenance Release

Fixed a number of errata and introduced some features new to AngularJS.

2014-08-04: Chapter 12: DOM Compilation and Basic Directives

Added Chapter 12.

2014-06-14: Chapter 11: High-Level Dependency Injection Features.

Added Chapter 11 and fixed a number of errata.

2014-05-24: Chapter 10: Providers

Added Chapter 10.

2014-05-18: Chapter 9: Modules And The Injector

Added Chapter 9 and fixed a number of errata.

2014-04-13: Chapter 7: Operator Expressions

Added Chapter 7. Also added coverage of literal collections in expressions to Chapters 5 and 6, and fixed some errata.

2014-03-29: Maintenance Release

Fixed a number of errata and introduced some features new to AngularJS.

2014-03-28: Chapter 6: Lookup And Function Call Expressions

Added Chapter 6.

2014-03-11: Part II Introduction and Chapter 5: Scalar Literal Expressions

Added Chapter 5 and fixed some minor errata.

2014-02-25: Maintenance Release

Fixed a number of errata.

2014-02-01: Chapter 4: Scope Events

Added Chapter 4 and fixed a number of errata.

2014-01-18: Chapter 3: Watching Collections

Added Chapter 3 and fixed a number of errata.

2014-01-07: Digest Optimization in Chapter 1

Described the new short-circuiting optimization for digest.

2014-01-06: Initial Early Access Release

First public release including the Introduction, the Setup chapter, and Chapters 1-2.