

TURING

图灵程序设计丛书

涵盖最新版本1.2.x

ng-book

The Complete Book on AngularJS

AngularJS

权威教程

[美] Ari Lerner 著
赵望野 徐飞 何鹏飞 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

ng-book

The Complete Book on AngularJS

AngularJS

权威教程

[美] Ari Lerner 著
赵望野 徐飞 何鹏飞 译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

AngularJS权威教程 / (美) 勒纳 (Lerner, A.) 著 ;
赵望野, 徐飞, 何鹏飞译. — 北京 : 人民邮电出版社,
2014.8

(图灵程序设计丛书)
ISBN 978-7-115-36647-4

I. ①A… II. ①勒… ②赵… ③徐… ④何… III. ①
超文本标记语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第167267号

内 容 提 要

本书是学习 AngularJS 的公认经典, 内容全面, 讲解通俗, 适合各层次的学习者。作者拥有丰富的 AngularJS 开发和教学经验, 也是一位全栈工程师。全书 35 章, 由浅入深地讲解了 AngularJS 的基本概念和基本功能, 包括模块、作用域、控制器、表达式、指令、路由、依赖注入等, 重要的是书中对每一个概念的讲解都配合了恰如其分的示例和代码, 让读者通过动手实践, 切身体会到这些概念的含义和价值。本书后半部分深入到 AngularJS 应用开发, 系统地讨论了服务器通信、事件、架构、动画、本地化、安全、缓存、移动应用等主题。

本书适合各个层次的 AngularJS 开发人员学习, 无论是出于工作需要, 还是好奇心的驱使, 只要你想彻底理解 AngularJS, 本书都会让你满载而归。

-
- ◆ 著 [美] Ari Lerner
 - 译 赵望野 徐 飞 何鹏飞
 - 责任编辑 李松峰
 - 执行编辑 李 静 许林玉
 - 责任印制 焦志炜

- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
- 邮编 100164 电子邮件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 北京隆昌伟业印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
- 印张: 29.75
- 字数: 760 千字
- 印数: 1—4 000 册

2014 年 8 月第 1 版

2014 年 8 月北京第 1 次印刷

著作权合同登记号 图字: 01-2014-5140号

定价: 99.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号



版 权 声 明

Original edition, entitled *The Complete Book on AngularJS Machines*. Copyright © 2013 by Ari Lerner.

Simplified Chinese translation copyright © 2014 by Posts & Telecom Press.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without permission in writing from W. W. Norton & Company, Inc.

本书简体中文版由 Ari Lerner 授权人民邮电出版社独家出版。未经出版者许可，不得以任何方式复制本书内容。

仅限于中华人民共和国境内（中国香港、澳门特别行政区和台湾地区除外）销售发行。

版权所有，侵权必究。

在微博上分享这本书

请帮Ari Lerner在新浪微博（<http://weibo.com/>）上宣传这本书。

推荐本书的微博：

#ngbook#我刚买了《AngularJS权威教程》！我准备构建高级、现代的Webapp！@图灵教育

点击下面这个链接，在新浪微博上搜索其他人对本书的评价：

<https://huati.weibo.com/k/ngbook>

献词

我把这本书献给我的父母，Lisa Lerner和Nelson Lerner，因为没有他们的支持和鼓励就不可能有这本书。

特别感谢

感谢可爱的Q，感谢你一直以来的激励，以及你在编辑方面的过人天赋。感谢我的共同创始人兼朋友Nate Murray。

译者序

2012年上半年，我所在的公司正在开发一个二次开发平台，它的目标是从数据库开始，能自由、方便地定制业务数据、规则、流程、服务接口，还有展现层。在对展现层的实现部分，我思考了很久，对其中部分技术细节还是缺乏好的思路，于是把眼光转到开源社区，无意中发现了AngularJS这样一个框架，详细考察之后，我认为它在很大程度上满足了我们的需求，继而投入了不小的精力进行研究。

在这两年里，我差不多遍历了它的源码，了解了很多细节的实现机制，并且与当时研究得较深的几位朋友，比如angularjs.cn的作者严清，资深开发者王宇鹏等进行了交流，获得了很多有益的信息，与此同时，也跟Avalon的作者司徒正美有过一些讨论，对前端MV*有了更深入的认识。后来，团队中的大漠穷秋翻译的《用AngularJS开发下一代Web应用》由电子工业出版社出版。作为国内第一本关于AngularJS的译著，它带动了学习和了解AngularJS框架的浪潮，也因此与朴灵的《深入浅出Node.js》一起，成为前端开发人员拓展思维和技能的两本最受欢迎图书。

到了2014年，我离开工作9年的地方，来到新的工作环境——苏宁云商，本来心里权衡过，很可能不再使用AngularJS的业务场景了，不曾想到入职之后面对的几个项目都属于云产品，正适合使用这类框架，因此又继续了对它的深入研究。

在此期间，图灵公司的李松峰老师发布了本书招募译者的消息，我心里一动就联系了他。经过沟通之后，我与另外两名译者，豌豆荚的@赵望野和腾讯的@basecss，合作翻译本书，每人负责1/3的内容。第一次正式翻译图书，我很忐忑，翻译过程中也遇到了一些困难。此前我虽翻译过一些技术文章，其中一篇恰好与AngularJS有关，但翻译图书跟翻译文章的差异很大，有很多东西要考虑一致性和连贯性。

本书内容丰富，从零开始向读者讲述AngularJS，首先介绍AngularJS的基本概念，以及在一些场景下的简单应用。接着，本书花很大篇幅讲解AngularJS的周边体系。我们使用这样一个框架，自然需要对前端的架构有一些考虑，包括代码的组织，一些第三方库的选择，甚至还有项目的建立、开发、测试、发布等各环节的综合考虑，这不再是一个简单的编码过程，而是一整套工程化的流程。

另外，我们也可能需要为这样一套前端的技术栈选择相应的后端服务，比如，可以使用Node.js自己建立，或者是利用互联网上已有的一些强大平台（比如Amazon等），在这些平台的帮助下，我们的AngularJS应用将如虎添翼，到达新的高度。

使用一个框架却不去深入了解它的原理，就会一直流于表面，当面对比较复杂的场景时，就找不到优化方案。因此，本书的后面部分也深入剖析了AngularJS的一些原理和拓展主题，比如国际化、移动开发、调试、性能优化等。

无论是零基础的入门级开发者，还是有过一定经验的中高级开发人员，都能从本书中受益。

目前，前端MV*框架百花齐放，AngularJS只是其中较流行的一种。这些框架孰优孰劣，其实并无定论，每个框架都会有它的适用场景，都有它优秀的一面，也没有哪个框架能够通吃所有业务场景，如果因为对一个框架的喜爱，而把它引入到不适合的产品中，一定是有害无益。

因此，我们希望读者在阅读本书时，能够多思考，愿大家在学习本书过程中都能收获满满。这样的话，作为本书译者的我们也将感同身受，与大家一同分享其中的喜悦和满足感。

在本书的出版过程中，除了我们三名译者之外，图灵公司的编辑李静也付出了很大的努力，支付宝的玉伯、51JS版主宝玉、百度的berg提出了不少宝贵意见，对此，一并表示衷心感谢。

徐飞

2014年7月

何鹏飞的个人致谢

感谢我所在的团队，给我提供非常好的环境，让我能学习成长。

感谢我在腾讯的导师@TooBug，让我接触到很多新东西，当然包括这本书所讲述的AngularJS。在翻译的过程中他也为我提供了很多帮助。

感谢图灵公司的出版团队，本书的出版离不开他们的努力和帮助。

最后，还要感谢没有在这里一一列出的帮助过我的每一个人。

赵望野的个人致谢

Web技术日新月异，每天早上翻看各种技术博客，都有一种逆水行舟不进则退的危机感，而这两年来前端MV*框架无疑是Web前端开发领域最热门的话题之一。之前已经读过本书英文版，其中涵盖了AngularJS开发的全部细节，示例代码清晰易懂，因此接下了中文版的翻译工作，并迫不及待地推荐给所有想学习AngularJS的朋友。感谢李松峰老师的帮助，感谢图灵出版团队的辛勤工作，感谢在翻译本书过程中给予帮助的所有朋友。

引言

序

似乎每天都有新的JavaScript库或框架发布，对此我多少已经有些麻木了。有能力从众多的库或框架中进行筛选是件好事，但至少在我看来，一个应用程序中如果包含了太多的脚本，对于维护来说却是件坏事。随着应用程序中脚本数量的增加，脚本间会产生依赖关系，所以我一直期待能有那么一到两个脚本，就提供我需要的所有核心功能。

当我第一次听说AngularJS时，它就立刻引起了我的注意，因为它只通过一个独立的框架就可以构建动态、交互密集型的客户端应用。通过进一步的研究，我确信这个第一判断是正确的，于是开始迷上了这个框架。AngularJS提供了一系列健壮的功能，以及将代码隔离成模块的方法，这对提高可复用性、可维护性和可测试性都是非常有益的。它的核心功能包括DOM操作、动画、模板、双向数据绑定、路由、历史管理、Ajax和测试，等等。

基于一个核心框架进行开发虽然很方便，但是学习它却充满挑战。一开始学习AngularJS时，我迷失在各种不同的主题中，并很快变得有些沮丧，甚至开始怀疑它到底是不是我想要的。服务是什么？它和工厂相比有什么区别？作用域服务是怎么同整个系统融合在一起的？指令是什么，我为什么要使用它？将这些零碎的知识点拼在一起形成大局观是我最初要克服的障碍。如果能有一些简明的参考资料，对于降低学习难度大有裨益。

很幸运，你已经有了这样一本优秀的参考资料，就是你手上的这本《AngularJS权威教程》，它将帮助你提升学习效率。本书作者将他掌握的AngularJS知识倾囊相授，并以非常容易理解和学习的方式呈现给大家。如果你想更深入地了解数据绑定、实时模板的工作原理、测试AngularJS应用的流程、服务和工厂的作用以及作用域和控制器如何协同工作等知识，那么这本书就是你所需要的。使用功能强大的AngularJS进行开发是一件非常有趣的事情，本书的示例将帮助你快速掌握这个框架。祝你的AngularJS项目一切顺利！

Dan Wahlin, 瓦林咨询公司^①

致谢

首先，我要感谢一直鼓励我完成这本书的每一个人。那些说写书很容易的人，一定没有亲自写过。

^① 个人博客和Twitter页面网址为<http://weblogs.asp.net/dwahlin>和<http://twitter.com/DanWahlin>。

我还要亲自感谢Q Kuhns对本书语法方面不厌其烦的修改和支持,感谢Erik Trom耐心地对细节进行修订,以及Nate Murray的清晰思路和乐观精神。

非常感谢Hack Reactor^①的全体成员在2013年的暑期课程中给了我一个探索如何在正式场合讲授AngularJS的机会。

同时也要感谢我在30x500的校友们,Sean Iams、Michael Fairchild、Bradly Green、Misko Hevery和整个AirPair团队。

最后,感谢那些对这本书的预览版提供帮助的人。我们从社区获得了非常棒的帮助和支持。特别要感谢以下三位:

- Philip Westwell
- Saurabh Agrawal
- Dougal MacPherson

关于本书

本书包含了能让你成为AngularJS[®]高手的解决方案。AngularJS是由Google[®]开发的先进前端框架,借助它你可以快捷高效地开发富交互应用。

本书提供了一系列前沿工具,使你在很短的时间内就可以上手创建令人印象深刻的Web体验。它能帮助你解决棘手的问题,并提供了一些可以立刻投入使用的实用技术。

本书涵盖的主题可以帮助你构建专业的Web应用,并能够非常顺利地执行。这些主题包括:

- 与RESTful风格的Web服务交互;
- 创建可复用的自定义组件;
- 测试;
- 异步编程;
- 创建服务;
- 提供先进的视觉效果;
- 其他更多内容。

本书的目标不仅是让你深刻了解AngularJS的运行原理,而且同时也提供了专业的代码片段,你可以对它们进行修改,从而构建你自己的应用。

借助这些工具和测试,你可以着手使用AngularJS开发自己的动态Web应用了,并且确信你的应用是可扩展的。

本书读者对象

本书写给那些从未使用AngularJS开发过Web应用,并且对如何开始使用这个优秀的框架心存好奇的读者。我们假定读者已经掌握了HTML和CSS,并且熟悉JavaScript(或者其他JavaScript

① <http://www.hackreactor.com>

② <http://angularjs.org>

③ <http://google.com>

框架)的基础知识。

本书组织结构

首先,本书涵盖了入门的基础知识,目的是帮助你很快上手使用AngularJS开发动态Web应用。

接下来会介绍AngularJS的工作原理,以及它与其他流行的JavaScript框架的差异。我们会深入讨论AngularJS应用内部的工作流程。

最后,我们将应用所学的知识开发一个相对复杂的应用程序。

其他资源

我们会引用AngularJS^①官方网站的文档。官方文档是非常好的学习资源,我们会经常用到它。

建议你先看一下AngularJS的API文档,通过它,你可以直接获得开发AngularJS应用的推荐方法。同时,这个文档肯定也是最新的。

本书排版约定

本书使用如下排版规范来表示不同类型的信息。

单行代码是这样的: `<h1>Hello</h1>`

代码块如下所示:

```
var App = angular.module('App', []);

function FirstController($scope) {
    $scope.data = "Hello";
}
```

命令行中的命令如下所示:

```
$ ls -la
```

Chrome(开发过程中使用的主要浏览器)开发者控制台中的命令如下所示:

```
> var obj = {message: "hello"};
```

新术语使用楷体。

重点文字将会加粗。

提示和技巧用如下图标标示:



这个图标表示提示。

提醒和陷阱用警告图标标示:



这个图标表示警告。

^① <http://angularjs.org>

错误信息用如下图标标示：



这个图标表示错误。

重要的补充内容使用如下图标标示：



信息框。

需要讨论的主题用如下图标标示：



这是一个讨论框。

开发环境

为了开发AngularJS应用，首先需要有一个顺手的开发环境。在整个学习过程中，我们会将精力主要放在两个环境中：编辑器和浏览器。

本书提到编辑器时指的是你使用的文本编辑器，而浏览器就是你使用的浏览器。强烈建议你下载Google的Chrome浏览器，因为它提供了一个非常强大的开发环境，可以使用开发者工具。

开始之前，我们还需要安装一些库。为了运行测试，我们需要Karma和Node.js。最好也装上git，但不强求。

本书不会介绍如何安装NodeJS。可以访问nodejs.org^①来获得更多信息^②。

虽然我们大部分工作都是在浏览器中完成的，但本书的部分内容也会重点介绍如何在服务器端通过构建RESTful风格的API来服务前端。

^① <http://nodejs.org>

^② 读者还可以参考图灵公司的《深入浅出Node.js》一书。——编者注

目 录

第 1 章 初识 AngularJS	1	7.2 表单验证	29
1.1 浏览器如何获取网页	1	第 8 章 指令简介	43
1.2 浏览器是什么	2	8.1 指令：自定义 HTML 元素和属性	44
1.3 AngularJS 是什么	2	8.2 向指令中传递数据	50
1.3.1 AngularJS 有什么不同	3	第 9 章 内置指令	56
1.3.2 许可	3	9.1 基础 ng 属性指令	56
第 2 章 数据绑定和第一个 AngularJS Web 应用	4	9.1.1 布尔属性	56
2.1 AngularJS 中的数据绑定	5	9.1.2 类布尔属性	58
2.2 简单的数据绑定	6	9.2 在指令中使用子作用域	59
2.3 数据绑定的最佳实践	8	第 10 章 指令详解	72
第 3 章 模块	10	10.1 指令定义	72
3.1 参数	11	10.1.1 restrict (字符串)	74
3.1.1 name (字符串)	11	10.1.2 优先级 (数值型)	75
3.1.2 requires (字符串数组)	11	10.1.3 terminal (布尔型)	75
第 4 章 作用域	12	10.1.4 template (字符串或函数)	76
4.1 视图和\$scope 的世界	12	10.1.5 templateUrl (字符串或 函数)	76
4.2 就是 HTML 而已	13	10.1.6 replace (布尔型)	77
4.3 作用域能做什么	14	10.2 指令作用域	77
4.4 \$scope 的生命周期	14	10.2.1 scope 参数 (布尔型或对象)	78
4.4.1 创建	15	10.2.2 隔离作用域	80
4.4.2 链接	15	10.3 绑定策略	81
4.4.3 更新	15	10.3.1 transclude	82
4.4.4 销毁	15	10.3.2 controller (字符串或函数)	84
4.5 指令和作用域	15	10.3.3 controllerAs (字符串)	86
第 5 章 控制器	16	10.3.4 require (字符串或数组)	86
5.1 控制器嵌套 (作用域包含作用域)	18	10.4 AngularJS 的生命周期	87
第 6 章 表达式	20	10.4.1 编译阶段	87
6.1 解析 AngularJS 表达式	20	10.4.2 compile (对象或函数)	88
6.2 插值字符串	21	10.4.3 链接	89
第 7 章 过滤器	24	10.5 ngModel	90
7.1 自定义过滤器	29	10.5.1 自定义渲染	92
		10.5.2 属性	92
		10.6 自定义验证	93

第 11 章 AngularJS 模块加载.....	95	第 15 章 同外界通信: XHR 和服务器通信.....	125
11.1 配置.....	95	15.1 使用\$http.....	125
11.2 运行块.....	96	15.2 设置对象.....	128
第 12 章 多重视图和路由.....	98	15.3 响应对象.....	130
12.1 安装.....	98	15.4 缓存 HTTP 请求.....	131
12.2 布局模板.....	99	15.5 拦截器.....	132
12.3 路由.....	99	15.6 设置\$httpProvider.....	133
12.4 \$location 服务.....	103	15.7 使用\$resource.....	134
12.5 路由模式.....	105	15.8 安装.....	134
12.5.1 HTML5 模式.....	105	15.9 应用\$resource.....	135
12.5.2 路由事件.....	106	15.9.1 基于 HTTP GET 方法.....	135
12.5.3 关于搜索引擎索引.....	107	15.9.2 基于非 HTTP GET 类型的方法.....	136
12.6 更多关于路由的内容.....	107	15.9.3 \$resource 实例.....	137
12.6.1 页面重新加载.....	107	15.9.4 \$resource 实例是异步的.....	138
12.6.2 异步的地址变化.....	107	15.9.5 附加属性.....	138
第 13 章 依赖注入.....	108	15.10 自定义\$resource 方法.....	138
13.1 推断式注入声明.....	109	15.11 \$resource 设置对象.....	139
13.2 显式注入声明.....	110	15.12 \$resource 服务.....	141
13.3 行内注入声明.....	110	15.13 使用 Restangular.....	142
13.4 \$injector API.....	111	15.14 Restangular 简介.....	142
13.4.1 annotate().....	111	15.15 安装 Restangular.....	143
13.4.2 get().....	111	15.16 Restangular 对象简介.....	144
13.4.3 has().....	111	15.17 使用 Restangular.....	145
13.4.4 instantiate().....	112	15.17.1 我的 HTTP 方法们怎么办.....	146
13.4.5 invoke().....	112	15.17.2 自定义查询参数和头.....	147
13.5 ngMin.....	112	15.18 设置 Restangular.....	147
13.5.1 安装.....	113	第 16 章 XHR 实践.....	153
13.5.2 使用 ngMin.....	113	16.1 跨域和同源策略.....	153
13.5.3 工作原理.....	113	16.2 JSONP.....	153
第 14 章 服务.....	114	16.3 使用 CORS.....	154
14.1 注册一个服务.....	114	16.3.1 设置.....	154
14.2 使用服务.....	116	16.3.2 服务器端 CORS 支持.....	155
14.3 创建服务时的设置项.....	118	16.3.3 简单请求.....	155
14.3.1 factory().....	119	16.3.4 非简单请求.....	156
14.3.2 service().....	119	16.4 服务器端代理.....	157
14.3.3 provider().....	120	16.5 使用 JSON.....	157
14.3.4 constant().....	122	16.6 使用 XML.....	158
14.3.5 value().....	122	16.7 使用 AngularJS 进行身份验证.....	159
14.3.6 何时使用 value()和 constant().....	123	16.7.1 服务器端需求.....	159
14.3.7 decorator().....	123	16.7.2 客户端身份验证.....	160
		16.8 和 MongoDB 通信.....	165

第 17 章 promise 168

- 17.1 什么是 promise 168
- 17.2 为什么使用 promise 169
- 17.3 Angular 中的 promise 170
- 17.4 链式请求 173
 - 17.4.1 all(promises) 174
 - 17.4.2 defer() 174
 - 17.4.3 reject(reason) 174
 - 17.4.4 when(value) 174

第 18 章 服务器通信 175

- 18.1 自定义服务器端 175
- 18.2 安装 NodeJS 175
- 18.3 安装 Express 176
- 18.4 调用 API 178
- 18.5 使用 Amazon AWS 的无服务器应用 181
 - 18.5.1 DynamoDB 181
 - 18.5.2 简单通知服务 (SNS) 181
 - 18.5.3 简单队列服务 (SQS, Simple Queue Service) 182
 - 18.5.4 简单存储服务 (S3) 182
 - 18.5.5 安全令牌服务 (STS) 182
- 18.6 AWSJS + Angular 182
- 18.7 开始 182
- 18.8 介绍 184
- 18.9 安装 184
- 18.10 运行 185
- 18.11 用户认证/鉴权 186
- 18.12 UserService 190
- 18.13 迁移到 AWS 上 191
- 18.14 AWSService 194
- 18.15 在 Dynamo 上开始 196
- 18.16 \$cacheFactory 196
- 18.17 保存 currentUser 197
- 18.18 上传到 S3 199
- 18.19 处理文件上传 201
- 18.20 查询 Dynamo 203
- 18.21 在 HTML 显示列表 204
- 18.22 出售我们的作品 205
- 18.23 使用 Stripe 206
- 18.24 使用 Firebase 的无服务器应用 209
- 18.25 使用 Firebase 和 Angular 的三方数据绑定 210
- 18.26 从 AngularFire 开始 211
 - 18.26.1 注册并创建一个 Firebase 211

- 18.26.2 包含 Firebase 和 AngularFire 库 212
- 18.26.3 把 Firebase 作为依赖项添加 212
- 18.26.4 绑定模型到 Firebase URL 212
- 18.26.5 数据同步 213
- 18.27 在 AngularFire 中排序 214
- 18.28 Firebase 事件 215
- 18.29 显式同步 215
- 18.30 用 AngularFire 进行认证 216
- 18.31 认证事件 217
 - 18.31.1 \$logout() 218
 - 18.31.2 \$createUser() 218
- 18.32 使用 Firebase 托管部署你的 Angular 应用 218
 - 18.32.1 安装 Firebase 工具 218
 - 18.32.2 部署你的 Web 站点 219
- 18.33 除了 AngularFire 之外 219

第 19 章 测试 220

- 19.1 为什么要做测试 220
- 19.2 测试策略 220
- 19.3 开始测试 220
- 19.4 AngularJS 测试的类型 221
 - 19.4.1 单元测试 221
 - 19.4.2 端到端测试 222
- 19.5 开始 222
- 19.6 初始化 Karma 配置文件 223
- 19.7 配置选项 226
- 19.8 使用 RequireJS 231
- 19.9 Jasmine 233
 - 19.9.1 细则套件 233
 - 19.9.2 定义一个细则 233
- 19.10 预期 234
 - 19.10.1 内置的匹配器 234
 - 19.10.2 安装和卸载 237
- 19.11 端到端的介绍 238
 - 19.11.1 选项输入 244
 - 19.11.2 重复循环元素 244
- 19.12 模拟和测试帮助函数 245
- 19.13 模拟\$httpBackend 246
- 19.14 测试一个应用 251
 - 19.14.1 测试路由 252
 - 19.14.2 测试页面内容 255
 - 19.14.3 测试控制器 257

19.14.4 测试服务和工厂	259	22.5.2 交错 CSS 动画	302
19.14.5 测试过滤器	263	22.5.3 什么指令支持交错动画	302
19.14.6 测试模板	264	22.6 使用 JavaScript 动画	302
19.14.7 测试指令	266	22.7 微调动画	303
19.15 测试事件	269	22.8 DOM 回调事件	304
19.16 对 Angular 的持续集成	270	22.9 内置指令的动画	304
19.17 Protractor	270	22.9.1 ngRepeat 动画	304
19.18 配置	272	22.9.2 ngView 动画	306
19.19 配置选项	273	22.9.3 ngInclude 动画	308
19.20 编写测试	275	22.9.4 ngSwitch 动画	310
19.21 测试实践	278	22.9.5 ngIf 动画	312
19.21.1 我们的应用	278	22.9.6 ngClass 动画	314
19.21.2 测试的策略	279	22.9.7 ngShow/ngHide 动画	316
19.22 建立我们的第一个测试	279	22.10 创建自定义动画	318
19.23 测试输入框	281	22.10.1 addClass()	319
19.23.1 测试列表	282	22.10.2 removeClass()	320
19.23.2 测试路由	284	22.10.3 enter()	321
19.24 页面对象	285	22.10.4 leave()	322
第 20 章 事件	287	22.10.5 move()	323
20.1 什么是事件	287	22.11 与第三方库集成	324
20.2 事件传播	287	22.11.1 Animate.css	324
20.2.1 使用\$emit 来冒泡事件	288	22.11.2 TweenMax/TweenLite	324
20.2.2 使用\$broadcast 向下传递事件	288	第 23 章 digest 循环和\$apply	326
20.3 事件监听	289	23.1 \$watch 列表	326
20.4 事件对象	289	23.2 脏值检查	327
20.5 事件相关的核心服务	290	23.3 \$watch	328
20.5.1 核心系统的\$emitted 事件	290	23.4 \$watchCollection	330
20.5.2 核心系统的\$broadcast 事件	290	23.5 页面中的\$digest 循环	330
第 21 章 架构	292	23.6 \$evalAsync 列表	331
21.1 目录结构	292	23.7 \$apply	332
21.2 模块	293	23.8 何时使用\$apply	332
21.3 控制器	294	第 24 章 揭秘 Angular	334
21.4 指令	296	24.1 视图的工作原理	335
21.5 测试	296	24.1.1 编译阶段	335
第 22 章 Angular 动画	297	24.1.2 运行时	336
22.1 安装	297	第 25 章 AngularJS 精华扩展	337
22.2 它是如何运作的	297	25.1 AngularUI	337
22.3 使用 CSS3 过渡	298	25.2 安装	337
22.4 使用 CSS3 动画	300	25.3 ui-router	337
22.5 交错 CSS 过渡/动画	301	25.3.1 安装	337
22.5.1 交错 CSS 过渡	301	25.3.2 事件	342
		25.3.3 \$stateParams	343
		25.3.4 \$urlRouterProvider	344

25.3.5 创建一个导航程序	345	27.12 编译新语言	378
25.4 ui-utils	346	27.13 改变语言	379
25.4.1 安装	347	第 28 章 缓存	381
25.4.2 mask	347	28.1 什么是缓存	381
25.4.3 ui-event	347	28.2 Angular 中的缓存	381
25.4.4 ui-format	348	28.2.1 \$cacheFactory 简介	381
第 26 章 移动应用	350	28.2.2 缓存对象	382
26.1 响应式 Web 应用	350	28.3 \$http 中的缓存	382
26.2 交互	350	28.3.1 默认的 \$http 缓存	382
26.2.1 安装	350	28.3.2 自定义缓存	383
26.2.2 ngTouch	351	28.4 为 \$http 设置默认缓存	384
26.2.3 \$swipe 服务	352	第 29 章 安全性	385
26.2.4 angular-gestures 和多点 触控手势	353	29.1 严格的上下文转义: \$sce 服务	385
26.2.5 安装 angular-gestures	354	29.2 URL 白名单	387
26.2.6 使用 angular-gestures	354	29.3 URL 黑名单	388
26.3 Cordova 中的原生应用程序	355	29.4 \$sce API	388
26.4 Cordova 入门	356	29.4.1 getTrusted	388
26.4.1 Cordova 开发流程	359	29.4.2 parse	389
26.4.2 平台	359	29.4.3 trustAs	389
26.4.3 插件	359	29.4.4 isEnabled	390
26.4.4 构建	360	29.5 配置 \$sce	390
26.4.5 模拟和运行	360	29.6 可信赖的上下文类型	390
26.4.6 开发阶段	360	第 30 章 AngularJS 和 IE 浏览器	391
26.4.7 Angular 中的 Cordova 服务	361	30.1 Ajax 缓存	393
26.5 引入 Angular	362	30.2 AngularJS 中的 SEO	393
26.6 使用 Yeoman 构建	363	30.3 使 Angular 应用可被索引	393
26.6.1 修改 Yeoman 以便使用 Cordova	364	30.4 服务端	393
26.6.2 装配 Yeoman 构建	365	30.4.1 hashbang 语法	394
26.6.3 构建移动部分	365	30.4.2 HTML5 路由模式	394
26.6.4 处理引导程序	367	30.5 服务端处理 SEO 的选项	394
第 27 章 本地化	369	30.5.1 使用 Node/Express 中间件	395
27.1 angular-translate	369	30.5.2 使用 Apache 重写 URL	395
27.2 安装	369	30.5.3 使用 Nginx 代理 URL	396
27.3 教你的应用一种新语言	370	30.6 获取快照	396
27.4 多语言支持	371	30.7 使用 Zombie.js 获取 HTML 快照	397
27.5 运行时切换语言	372	30.8 使用 grunt-html-snapshot	398
27.6 加载语言	373	30.9 Prerender.io	399
27.7 angular-gettext	374	30.10 <noscript> 方法	400
27.8 安装	374	第 31 章 构建 Angular Chrome 应用	401
27.9 用法	375	31.1 了解 Chrome 应用	401
27.10 字符串提取	375	31.1.1 manifest.json	401
27.11 翻译字符串	377	31.1.2 背景脚本	401