



Switching to Angular 2

迈向Angular 2: 基于TypeScript的高性能SPA框架

AngularJS与Angular 2之父Miško Hevery作序力荐

[保] Minko Gechev 著
大漠穷秋 熊三 译



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Switching to Angular 2

迈向Angular 2

基于TypeScript的高性能SPA框架

[保]Minko Gechev 著

大漠穷秋 熊三 译

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书由 Angular 之父 Miško Hevery 作序，作者为 Angular 开发团队成员之一。基于对 Angular 2 架构和设计方面的深入理解，本书选材精准、内容实用。全书从一个小例子开始引导读者快速上手，详细介绍 Angular 2 带来的所有新特性，包括组件、指令、TypeScript、新的路由机制、管道、服务端渲染等。

本书读者对象包括 Angular 1.x 的开发者、想直接从 Angular 2 开始入手的学习者，以及前端技术架构师等。

Copyright © 2016 Packt Publishing. First published in the English language under the title ‘Switching to Angular 2’.

本书简体中文版专有版权由Packt Publishing授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有版权受法律保护。

版权贸易合同登记号 图字：01-2016-5249

图书在版编目（CIP）数据

迈向 Angular 2: 基于 TypeScript 的高性能 SPA 框架 / (保) 明科·基彻 (Minko Gechev) 著; 大漠穷秋, 熊三译. —北京: 电子工业出版社, 2016.8

书名原文: Switching to Angular 2

ISBN 978-7-121-29409-9

I. ①迈… II. ①明… ②大… ③熊… 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 (2016) 第 165737 号

责任编辑: 张春雨

印 刷: 北京中新伟业印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787 × 980 1/16

印张: 15.25

字数: 319.64 千字

版 次: 2016 年 8 月第 1 版

印 次: 2016 年 8 月第 1 次印刷

定 价: 69.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819 faq@phei.com.cn。

译者序

谢谢我的粉丝们。

Angular 2 的发布标志着这门框架已经走向成熟和稳定，正如官方申明所说：Angular 2 的核心架构已经定型，不会再做颠覆性的设计。

然而，鉴于你们前端界“每隔 6 个月就重写一遍”的现状，我本人对以上申明持观望的态度。纵观当前市场上的所有前端框架，只有 jQuery 和 ExtJS 大体上做到了“核心架构不变”。

但是与其他所有前端框架相比，Angular 2 有一点非常具有前瞻性，它是基于 TypeScript 开发的。JavaScript 之父 Brendan Eich 对他自己发明的语言有一段经典的评价：

与其说我爱 Javascript，不如说我恨它。它是 C 语言和 Self 语言一夜情的产物。18 世纪英国文学家约翰逊博士说得好：它的优秀之处并非原创，它的原创之处并不优秀。

对于这段评价，只要写过 JavaScript 的人，一定会表示不能同意更多。JavaScript 的灵活性、动态性，以及目前各种新标准的剧烈变革直接把学习成本推高了 N 个数量级，对于大规模的开发团队尤其如此。因此，Angular 2 直接基于静态类型的 TypeScript 来进行开发，无疑是合适的，直接从语言层面上就把一些坑给填了，这是非常具有战略眼光的决策。对于有 Java、C++，或者 ActionScript 背景的开发来说，TypeScript 的学习成本几乎为零，你只要花 30 分钟浏览一下大致的语言特性就能写得风生水起了。

在当前这个时间节点上，各种前端技术层出不穷，让人有目不暇接之感。从学习的角

度而言，这是一种负担，但是这种状况同时也意味着大量的工作机会，毕竟沧海横流方显英雄本色。

本书所有代码都在 GitHub 上，原作者会不断更新：

<https://github.com/mgechev/switching-to-angular2>

大漠穷秋

2016-06-06

推荐序

Angular 2 依然是 Angular，只是更强大而已。它依然构建在那些相同的原则之上，这些原则也是你们热爱 AngularJS 的原因：构建单页应用的一种快速而强大的解决方案。在 Angular 2 中，应用将会运行得更加快速，对 SEO 和移动设备更加友好，并且是天然跨平台的。虽然 Angular 2 已经对 AngularJS 中的很多概念做出了大幅度的演进，但是仍然保留了上一个版本的设计哲学。

《迈向 Angular 2：基于 TypeScript 的高性能 SPA 框架》一书完全可以佐证以上观点。所以，Minko 的这本书将会成功地帮你把思维模式从 AngularJS1.x 切换到 Angular 2。从你与 Angular 2 的初次邂逅直到最后，Angular 的核心概念将会贯穿始终。这本指南将会帮你切换到用 Angular 做事的新方式上去。Minko 将会引导你贯通所有变更的内容，包括引入的所有新特性：组件、指令、TypeScript、新的路由机制，以及开始使用 Angular 2 开发你的下一个项目时所需要的一切内容。

如今的 Web 开发领域日新月异，Angular 2 接受了这一挑战，并且构建在 AngularJS 的遗产之上。因此，对 Angular 社区来说，出现高质量的学习资料是极其重要的，而 Minko 的这本书将会帮助 Angular 开发者迈出面向未来的第一步。

Miško Hevery
AngularJS 与 Angular 2 之父

关于作者

Minko Gechev 是一名软件工程师，他坚定地信仰开源软件。他开发过许多项目，包括 Angular JS 1. x 和 Angular 2 style guides、Angular 2-seed、Angular 2 项目静态代码分析器、aspect.js、angular-aop，以及很多其他项目。他开办了 JavaScript、Angular 等 web 技术培训课程。

Minko 热衷于进行各种计算机科学概念实验并把它们投入到实战中去。他在 ng-vegas、AngularConnect、ITWeekend Kiev、Angular JS-SF 及 Angular Berlin 等全球论坛上进行过关于 Angular 与软件开发相关话题的演讲。

感谢 Misko Hevery 在软件工程和技术审查方面对本书做出的巨大贡献。他竭尽全力为我提供了精确的内容。为了让书中的实例代码运行起来更简单，我开发了 Angular 2-seed 这个项目(这个项目提交在 github 上——译者注)。Ludovic Hénin 是该项目的核心贡献者，他的工作让这个项目远远超越了 Angular 2 入门级的水平。我还要感谢 Daniel Lamb、Radoslav Kirov 和 Tero Parviainen，他们给我提供了极其宝贵的反馈意见！

没有 Packt 出版团队的全心工作，我不可能完成此书。

最后，我想感谢谷歌的 Angular 团队，感谢他们为大家带来这样一个框架，也为我带来源源不断的灵感。

关于审校者

Miško Hevery 是 AngularJS 框架之父。他酷爱化繁为简。他目前就职于 Google，此前他在 Adobe、Sun Microsystems、Intel 以及 Xerox 工作过，这些经历让他成为了使用 Java、JavaScript、Flex 以及 ActionScript 这些 web 技术开发 web 应用方面的专家。

Daniel Lamb 是一位资深的软件开发专家，同时也是一位乐于分享知识的作家，他专攻大规模架构和 web 前端开发。在 16 年职业生涯中，他支持了数十亿次的访问，让数亿人能够连接到一起互相交流。

序言

AngularJS 是一个 JavaScript 开发框架，致力于让 web 应用开发变得更加简单。目前，它已经被应用在大规模、大流量的网站中，这些网站饱受性能低下、移植性差的困扰，同时还面临 SEO 不友好、复杂度大的问题。Angular 2 改变了这一切。

它是一款非常现代的框架，可以利用它构建性能更高、健壮性更强的 web 应用。《迈向 Angular 2：基于 TypeScript 的高性能 SPA 框架》是掌握 Angular 2 最快捷的方式，它将引领你进入 Angular 2 的全新世界。

读完本书，你将会具备利用 Angular 2 所提供的一系列新特性来快速有效地构建应用的能力。

本书内容

第 1 章 Angular 2 快速上手：开启 Angular 2 新世界的旅程。这一章描述了框架设计决策背后的一些主要因素。我们将会看到形成这门框架的两种主要驱动力——web 的当前状态以及前端框架的进化。

第 2 章 Angular 2 应用的基础构件：简要介绍 Angular 2 引入的一些核心概念。我们将会探讨 AngularJS 1.x 所提供的基础构件与最近一个主版本之间的差异。

第 3 章 TypeScript 速成：Angular 2 是语言无关的，但是 Google 推荐大家利用 TypeScript 所带来的静态类型特性。在这一章中，你将会学习利用 TypeScript 开发 Angular 2 应用的所有必要语法！

第 4 章 Angular 2 组件和指令入门：解释开发用户界面所需要的核心构件——Directive 以及 Component。我们将会深度解析视图封装、内容投影、输入输出、脏值检测等概念。同时还会讨论一些高级主题，例如：模板引用，以及使用不可变数据类型加快应用的运行速度。

第 5 章 Angular 2 中的依赖注入：这一章将会全面解析这门框架中最强大的特性之一：依赖注入机制。它最初是由 AngularJS 1.x 引入的。我们可以利用这一特性来编写更加便于维护、测试，并且更易于理解的代码。在这一章结束的时候，我们将会理解如何在 Service 中定义业务逻辑，然后利用 DI（依赖注入）机制把它和 UI 黏合到一起。我们还会解析一些更加高级的主题，例如：注射器的层级结构、配置 provider 等。

第 6 章 Angular 2 中的路由和表单：这一章将会探索在开发实际应用的过程中如何使用用来维护 form 的新模块。我们将会实现一个页面，可以显示 form 中输入的值。最后，我们会利用基于组件的路由把各个单独的页面黏合成一个完整的应用。

第 7 章 详解管道以及与 RESTful 服务端之间的通信：深度解析路由和表单模块。这一章将会解析如何开发数据模型驱动型表单，以及如何定义参数化路由、子路由。我们还会解释 HTTP 模块，以及如何开发有状态和无状态管道。

第 8 章 开发体验与服务端渲染：探索开发 Angular 2 应用中的一些高级主题，例如：在 WebWorker 中运行应用，以及服务端渲染。在这一章中的第二部分，我们将会学习一些全新的开发工具，从而让日常开发工作更加轻松一些，例如：angular-cli、angular2-seed，以及解释热重载的概念等。

阅读准备

对于本书中的绝大多数实例，你需要：一个简单的文本编辑器或者 IDE、装好 Node.js 和 TypeScript、能上网，以及一个浏览器。

对于每一章所提供的实例代码，对应章节都会解释需要安装的软件。

目标读者

你想深入学习 Angular 2 吗？或许你想先评估一下最新的变更再决定是否跟进？如果是，那么《迈向 Angular 2：基于 TypeScript 的高性能 SPA 框架》就是为你量身定制的。

为了能够充分理解本书内容，需要对 AngularJS 1.x 有基本的理解，同时需要对

JavaScript 非常熟悉。阅读本书不需要预先了解 Angular 2 所引入的变更。

排版约定

在本书中，根据不同的内容会采用各种不同的文字风格。下面解释其中一些文字风格的含义。

文本中的代码单词、数据库表名、目录名、文件名、文件后缀、路径名、伪 URL 链接、用户输入，以及 Twitter 路径将会这样排版：“你将会看到相同的结果，但是 `test.js` 这个文件不会存储到硬盘上。”

代码块将会这样显示：

```
@Injectable()
class Socket {
  constructor(private buffer: Buffer) {}
}

let injector = Injector.resolveAndCreate([
  provide(BUFFER_SIZE, { useValue: 42 }),
  Buffer,
  Socket
]);

injector.get(Socket);
```

当需要注意代码块中的某个部分时，对应的行或者项目将会加粗显示：

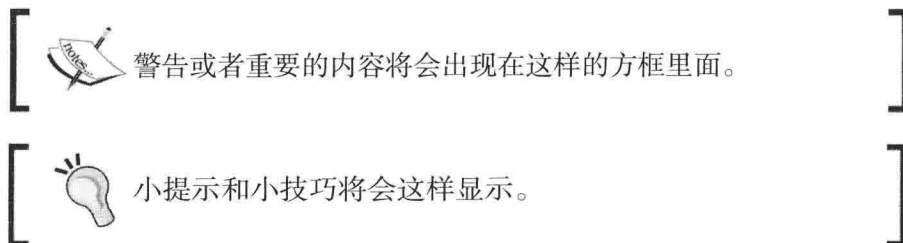
```
let injector = Injector.resolveAndCreate([
  provide(BUFFER_SIZE, { useValue: 42 }),
  Buffer,
  Socket
]);
```

对于代码库中的代码，本书对应的实例代码之前都会有一行注释，用来说明代码对应的文件路径，所有路径都相对于 `app` 目录：

```
// ch5/ts/injector-basics/forward-ref.ts

@Injectable()
class Socket {
  constructor(private buffer: Buffer) {...}
}
```

新术语和重要的词语都会加粗显示。你可以在屏幕上看到的单词，例如菜单和对话框中的文本，将会这样显示：“当标签渲染到屏幕上之后，用户唯一能看到的内容就是这样一段文本：**Loading...**”。



代码下载

可以从 GitHub 下载本书中的所有实例代码：<https://github.com/mgechev/switching-to-angular2>。

下载步骤如下：

1. 在浏览器地址栏中输入以上 URL 路径。
2. 单击屏幕右侧中间的“Download ZIP”按钮开始下载。

也可从 <http://www.broadview.com.cn> 下载所有已购买的博文视点书籍的示例代码文件。

勘误提交

虽然我们已经尽力谨慎地确保内容的准确性，但错误仍然存在。如果你发现了书中的错误，包括正文和代码中的错误，请告诉我们，我们会非常感激。这样，你不仅帮助了其他读者，也帮助我们改进后续的出版。如发现任何勘误，可以在博文视点网站相应图书的页面提交勘误信息。一旦你找到的错误被证实，你提交的信息就会被接受，我们的网站也会发布这些勘误信息。可以随时浏览图书页面，查看已发布的勘误信息。

目录

序言	XV
第 1 章 Angular 2 快速上手	1
Web 的进化——新框架时代	2
ECMAScript 的进化	2
Web Component	3
WebWorker	4
从 AngularJS 1.x 中学到的经验	5
Controller	6
Scope	7
依赖注入	7
服务端渲染	8
大规模应用	9
模板	10
脏值检测	12
本章小结	12
第 2 章 Angular 2 应用的基础构件	14
Angular 2 概念性简介	15

脏值检测	17
认识 Angular 2 中的组件	19
组件实战	20
Angular 2 中的组件	22
管道	23
定义管道	24
脏值检测	25
传统的脏值检测	26
AngularJS 1.x 中的脏值检测	27
增强 AngularJS 1.x 的脏值检测	29
理解服务	30
理解新的基于组件的路由机制	33
Angular 2 中定义路由的语法	35
本章小结	36
 第 3 章 TypeScript 速成	 37
TypeScript 简介	37
编译时类型检查	38
文本编辑器和 IDE 的支持更好	38
TypeScript 的更多特性	39
TypeScript 用法	39
用 npm 安装 TypeScript	40
运行我们的第一个 TypeScript 程序	40
TypeScript 从 ES2015 和 ES2016 中引入的语法和特性	41
ES2015 中的箭头函数	41
使用 ES2015 和 ES2016 中的类	43
定义在块级作用域中可见的变量	45
使用 ES2016 装饰器进行元编程	46
使用可配置的装饰器	47
使用 ES2015 编写模块化的代码	48
使用 ES2015 中的模块语法	49

利用隐式异步行为	50
使用别名	50
导入所有导出的模块	50
默认导出	51
ES2015 模块加载器	52
ES2015 和 ES2016 总结	52
发挥静态类型的优势	53
使用显式类型定义	53
理解原生类型	54
理解 Object 类型	55
定义类	58
使用访问修饰符	59
定义接口	61
使用 TypeScript 装饰器提升表现力	64
使用类型参数编写泛型代码	64
使用泛型函数	66
多重泛型	66
利用 TypeScript 的类型推断机制简化代码	67
最常见的类型	67
与上下文有关的类型推断	68
使用外部类型定义	68
使用预定义的外部类型定义	68
自定义外部类型	70
定义 ts.d 文件	72
本章小结	72
第 4 章 Angular 2 组件和指令入门	74
Angular 2 Hello world!	75
配置开发环境	78
初始化项目仓库	78
Angular 2 和 TypeScript 上手试玩	79

首页代码深度解析	80
Angular 2 指令用法	81
ngFor 指令	83
改进了指令语法的语义	83
在模板内部定义变量	85
在模板里面使用语法糖	85
定义 Angular 2 指令	85
设置指令的输入	87
理解指令的构造函数	87
封装指令的更好方式	88
Angular 2 内置指令的用法	89
组件视图封装简介	90
实现组件的控制器	90
处理用户交互	92
指令的输入和输出	93
找到指令的输入和输出	94
定义组件的输入和输出	95
传递输入与使用输出结果	97
事件冒泡	99
重命名指令的输入与输出	100
定义输入输出参数的另一种语法	101
详解 Angular 2 中的内容投影	102
Angular 2 中的内容投影入门	102
投射多块内容	103
组件嵌套	105
ViewChildren 和 ContentChildren 的用法	106
ViewChild 与 ContentChild	108
挂钩到组件的生命周期上	112
执行的顺序	114
用 TemplateRef 定义通用视图	115
理解并优化脏值检测机制	118

脏值检测器的执行顺序	118
脏值检测策略	120
利用不可变数据和 OnPush 策略提升性能	121
在 Angular 中使用不可变数据结构	122
本章小结	124
第 5 章 Angular 2 中的依赖注入	125
为什么要依赖注入?	125
Angular 2 中的依赖注入	126
Angular 2 中 DI 的优点	127
配置注射器	127
使用生成的元数据解析依赖关系	129
初始化注射器	129
前向引用简介	130
配置 provider	132
定义实例化服务的工厂	134
子注射器以及可见性	136
构建注射器层级结构	137
配置依赖关系	138
在组件和指令中使用 DI	144
元素注射器简介	145
在 ES5 中使用 Angular 的 DI 机制	149
本章小结	152
第 6 章 Angular 2 中的路由和表单	153
开发一个名为“码农仓库”的应用	153
探索 Angular 2 中的路由	156
定义根组件并启动应用	157
PathLocationStrategy 的用法	158
使用 @RouteConfig 配置路由	158
routerLink 和 router-outlet 的用法	160