

深入讲解AngularJS的基本概念及其背后的原理，包括完整的开发框架与最佳实践。
不仅抽丝剥茧地展现了AngularJS的诸多特性与技巧，还讲解了工程实践中容易陷入的“坑”，是从小工走向专家的必备参考。



雪狼 破狼 彭洪伟 编著

An in-Depth Analysis and Best Practices for AngularJS

AngularJS

深度剖析与最佳实践



机械工业出版社
China Machine Press



An in-Depth Analysis and Best Practices for AngularJS

AngularJS

深度剖析与最佳实践

雪狼 破狼 彭洪伟 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

AngularJS 深度剖析与最佳实践 / 雪狼, 破狼, 彭洪伟编著. —北京: 机械工业出版社, 2015.11

(实战)

ISBN 978-7-111-52096-2

I. A… II. ①雪… ②破… ③彭… III. 超文本标记语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2015) 第 270565 号

本书深入讲解 AngularJS 的基本概念及其背后的原理, 包括完整的开发框架与最佳实践。不仅抽丝剥茧地展现了 AngularJS 的诸多特性与技巧, 还讲解了工程实践中容易陷入的“坑”, 是从小工走向专家的必备参考。

本书首先从实战开始, 通过实战演练逐步带领读者体验 Angular 的开发过程, 并随着进度的推进, 引入所需的技术和概念。其次对在实战中提到的一些概念进行深入讲解: 包括这些概念怎么用, 什么时候用, 什么时候不用等。然后讲解这些概念背后的原理, 看看这些概念之间是如何协作的, 包括 AngularJS 的工作模式等。最后介绍最佳实践, 将主要介绍实战经验, 包括如何发掘一些不常用但很有用的 API, 如何把看起来平淡无奇的框架特性运用得出神入化等。此外, 作者还从实际工作中总结了一些开发技巧和容易陷入的“坑”, 以及常见的优秀工具及其使用经验, 这对于实际开发工作非常有参考价值。

AngularJS 深度剖析与最佳实践

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 吴 怡

责任校对: 殷 虹

印 刷: 北京诚信伟业印刷有限公司

版 次: 2016 年 1 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 21.5

书 号: ISBN 978-7-111-52096-2

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

这是一本具有强烈 ThoughtWorks 项目风格的书。书中打造的实战项目完全遵循了 ThoughtWorks 工程实践，一步一步从最初的框架通过快速迭代逐步丰富项目的骨肉，并在这个过程中抽丝剥茧地展现了 AngularJS 的诸多特性与技巧，如循循善诱的导师一步步指导着你从 AngularJS 的小工走向专家。

这里所谓的“专家”不仅仅是指你对 AngularJS 的诸多技巧尽皆了然于胸，能够挥洒自如地运用于项目开发中——若能如此，不过是“唯手熟尔”的工匠罢了。真正的专家需要从大处着手，挖掘这门技术背后隐含的设计思想与哲学，换言之，需要知其所以然，却又不偏废细节，锱铢必较每个变量函数的命名格式，使代码臻于完美，并从中提炼出能够推而广之的最佳实践。

从知其所以然入手，书中的第 3 章“背后的原理”加强了内容的深度，使得本书不至于沦落为一本 Example Step by Step。书中通过对 MVVM 模式的阐释，解释了 Angular JS 的设计原理与启动流程，并给出了 Angular JS 开发的注意事项。书中写道：

MVVM 模式的要点是：以领域 Model 为中，遵循“分离关注点”设计原则。这也是 Angular 的模型驱动思维与 jQuery 的 DOM 驱动思维的显著差异。所以我们在做 Angular 开发的时候应该谨记以下两点：

- ❑ 绝不要先设计你的页面，然后用 DOM 操作去改变它。
- ❑ 指令不是封装 jQuery 代码的“天堂”。

又例如细节之处，本书作者仿佛是踏着针尖在跳舞，刻绘的细节纤毫毕现；又佐以代码，论证有理有据；阅读时，真好像是你和雪狼、破狼在一起结对编程呢。例如书中在提及对服务访问对象（SAO）的封装时，给出了这样两段代码：

```
angular.module('com.ngnice.app').controller('ReaderCreateCtrl', function ReaderCreateCtrl($resource) {
```

```

var vm = this;
var Reader = $resource('/api/readers/:id', {id: '@id'});
vm.submit = function(form) {
    Reader.save(form);
};
});

```

封装后:

```

angular.module('com.ngnice.app').controller('ReaderCreateCtrl', function
    Reader-CreateCtrl(Reader) {
    var vm = this;
    vm.submit = function(form) {
        Reader.save(form);
    };
});

```

寥寥几行代码的区别，却体现了作者对于代码可读性的执着追求。如此内容在书中俯拾皆是。作者对整洁代码的敏感度，就好像水银温度计对气温的感知一般，哪怕是一丝一毫都能准确感知，进而在展开的文字叙述中潜移默化地影响读者。尤其针对初学者，作者从一开始就展示了什么是 AngularJS 之美，什么是代码之美，什么是设计之美，就好似建立了 AngularJS 世界的“潜规则”，入了这个圈，你需如此这般，否则就得荆棘一路，步履蹒跚。而那些优秀的工程实践，例如测试驱动开发、面向模型编程、迭代的演化、一次只做一件事情的行为准则，则完全融化成本书的血液，通过简单朴实的词语，天然地流淌在整本书中，和风细雨，润物细无声。

我与本书的作者之一破狼相交甚深，虽然一直未有机会共同战斗在一个项目，却也有许多机会彼此沟通各自对设计的理解。在面向对象设计、领域驱动设计、架构设计等诸多方面，我们抱有相同的设计态度，可谓志同道合。问道技术，犹如饮酒论文，酒酣耳热时，得聆佳音，当浮一大白，人生乐趣大抵如此。虽然我对前端技术所知了了，但阅读此书，许多论点刚好击中我的肺腑，那种如寻觅到知己一般的快乐，真可以说是阅读之余的额外收获了。我喜欢此书的朴实，它没有去构架飘渺高深的理论，没有浮夸地吹嘘 AngularJS 如何优秀，在前端开发中所向披靡。技术人写文章，常常没有卖弄，只是踏实地表达对技术的一己之得，读者得到的却是字字铮铮的金石之音。

显然，这是一本工程师写给工程师阅读的书，我只可惜这本书的出版来得有点晚了。
是为序。

张逸

2015 年 8 月 7 日夜 旅行中，在斯坦福大学安静的校园

新时代

新挑战

时代已经不同了！

17 年前，当我的第一个作品推入市场的时候，互联网才刚刚传入中国。

那时候的软件不需要联网，每个用户也不需要知道其他用户的存在。

那时候只需要考虑 PC 运行环境，而需要考虑的屏幕分辨率也只有区区三种。

那时候的软件项目组多则十几人，少则一人，而发布周期常常会达到半年之久。

现在，一切都不同了。

现在，连一个手机手电筒软件都在偷偷联网，不能联网的游戏也已经是老古董的代名词。

现在，软件不但运行在 PC 上，还要运行在智能手机上，运行在各种 Pad 上，屏幕分辨率更是多到让研发和测试工程师发怵的地步。

现在，外界看到的产品其实只是冰山一角，它背后还有很多子系统紧密协作来提供支持，需求和架构的复杂度也暴增。

但最大的挑战恐怕还是来自发布周期——一期版本在一个月內上线已是常态，而修复 bug 的时间限制则往往以小时计，甚至以分钟计。

没错，这些都是新的挑战！好在，我们也有了新技术！

新技术

这 17 年间，软件业最大的技术革命，当然首推互联网。

互联网不但拓展了软件业的业务范围，更改变了程序员获取知识和解决问题的方式。

如今，一个不会 Google（以及翻墙），没上过 GitHub，不知道 Stackoverflow 的程序员

很难想象会有什么发展空间。

排在第二位的技术革命，当推移动互联网以及智能终端。这场革命不但把曾经的王者诺基亚打落凡尘，而且让苹果重新登上王位。

这两场技术革命让开源运动遍地开花，更催生了无数的新技术。

且不提 HTTP/HTML/JavaScript/CSS 这些耳熟能详的互联网基石，就连在互联网革命爆发之前已经就已经相当成熟的 OO 领域也有了很大的进展。

以 MVC 为例，它不但衍生出 MVP、MVVM 等很多变种，而且从后端领域扩展到了前端领域。而现在日益火爆的 Angular，正是 MVC 在前端领域的代表作之一。

一个“极客”总是痴迷于各种“漂亮”的技术，而 Angular 当之无愧地是其中之一，它可供借鉴的地方很多：

- ❑ 如何弥补语言的先天不足。
- ❑ 如何干净漂亮地解耦。
- ❑ 如何设计“小而美”的类/代码块。

所以，即使你还没有下决心把 Angular 应用到项目中，也可以在学习 Angular 的过程中获得一些启迪，帮助你重构现有项目。

面对技术的快速进步，有人会感到恐慌，有人会盲目地追踪一切新技术，而真正的极客会看到“新”技术中那些“不变”的元素，会在“新挑战”中看到“新机遇”，并且把握。

新机遇

一方面出现了前所未有的挑战，另一方面出现了前所未有的技术，这样的机遇并不多，“极客”们欢呼雀跃。

对于公司，它将影响产品形态、开发速度和产品品质，也会对团队的组织架构带来改变。比如，伴随着设备的多样化，网络服务的访问入口变得多样化：不但需要有供电脑访问的网站，还需要供手机访问的网站、供 Pad 访问的网站，对于一些追求极致用户体验的公司来说，还会提供给安卓设备用的 App、给苹果设备用的 App。

作为开发人员，也许你会看到或正在经历一个工作量暴增的时代，但是，不要紧张，事情没那么坏。在新时代，有一项重要且迅速成长的技术革新，那就是“前后端分离架构”，它可以有效遏制工作量的暴增。“前后端分离架构”正是伴随着“前端 MVC”的成长而成长的。

它的原理很简单：虽然多出了很多访问入口，但是其背后的业务逻辑并没有本质性变化，那么，我们是否可以让这一套业务逻辑为多种不同形态的终端服务呢？答案是肯定的，

那就是让后端只提供跟业务逻辑紧密相关的那部分 API，而用户交互等非核心逻辑则交给前端程序来完成。

这样，我们的工作量并不会成倍增长，而是可以先着重开发一个版本，让后端 API 和一种形态的前端应用变得成熟，然后再去开发其他形态的前端应用。而这些其他形态的前端程序的工作量和风险都比较容易得到控制。

但是，根据康威定律，在新的程序架构下，项目的组织架构甚至整个公司的组织架构都将发生相应的变化，而最显著的变化就是出现了专门的前端工程师。前端工程师往往不是零基础开始的，一小部分来自原来负责切图或写 JavaScript 特效的工程师，不过大部分是从以前开发 Web 应用的程序员转型而来的。

无论对于公司还是个人，“前端 MVC”以及相应的“前后端分离架构”都是一个新的机遇。不思进取的王者终会没落，勤奋好学的新星将会崛起，希望本书能有幸成为你的助力。

致读者

写给想转职或兼修前端 Web 工程师

本书面向的读者，第一大群体是 Web 工程师。“前后端分离架构”出现之前，在大多数 Web 应用中，无论是核心的业务逻辑，还是表现层的交互逻辑，都是完全运行在服务端的。写这类程序的程序员就是这里所说的 Web 工程师。

随着“前后端分离架构”的普及，原来的开发方式将主动或被迫转变。本书将通过实例引导你完成到“前后端分离架构”的思维转变，以及与此相关的技术。

如果你是个 Web 工程师，在读本书的时候请留意用户交互逻辑是如何完全移交给前端程序的，而后端程序又做了哪些精简，特别要注意体会模块职责的单一化、专业化趋势。

对于部分转职过来的 Web 工程师，除了转换思维以外，还有一大挑战是前端庞杂的知识体系：HTML/CSS/JavaScript/ 前端工具链 / 浏览器兼容性等，每一个领域都相当庞大。

在本书中，我们无法对此展开讲解，但这些知识对于做实际项目又是必需的。所以，我们只能在附录中提供一些重要的技术要点和“坑”，并且给出一些在线学习资源和书单。这些大部分都是从我们开展培训时所使用的课件改编而来的，具有很强的实战性、实用性。希望可以为你提供一些第三方资料，作为进一步学习的起点。

写给想进阶为专业前端的切图师

在很多开发组中，切图师往往由初级程序员或美工担任，有没有想过自己将来向哪里

发展？除了面向对象、项目管理等必学的基础技能之外，还可以学习数据库、后端框架、安全技术等，转职为后端工程师。也可以学习 HTML/CSS/JavaScript、用户体验、交互设计、前端框架等，转职为前端工程师。

当然，如果你足够聪明和有足够的进取心，你也可以两者兼修，成为一名全栈工程师。不过，相对来说，前端这条路径可能更加平缓。而且，这几年前端职位正逐渐火爆，从个人职业发展来说，这也是个不错的选择，过一段时间后未必再有这样好的机遇。从切图师到前端，这条路并非荆棘重重。事实上，没有传统 Web 工程师的思维定势，这反倒会是个优点。在笔者的编程、咨询和教学过程中，曾接触过一些对 Angular 感兴趣的人，总体上说，转变思维比导入新思维的难度更大。圈子里还常流传一些无稽之谈，比如，Angular 是 Google 开发的，面向的是 Google 中那些妖怪级程序员。那都是乱传的，没那么恐怖。

我写下这些，是希望你们可以轻装上阵，Angular 的很多设计都是遵循“最小意外”原则的，靠直觉就可以掌握，“高估难度”有害无益。

不过，难度仍然是有的。读本书之前，你至少应该已经熟悉了 JavaScript 语法，对 Angular 的各种概念有了大致的了解。如果你对很多新名词不知所云，那么建议先去翻阅一下附录中的书籍，浏览一下网上关于 Angular 入门的文章。

对于切图师来说，MVC 方面的基础往往会成为短板，而 JavaScript 中一些诡异的特性也常常带来困扰。所以，本书会穿插一些这方面的简短知识。但是，对于一个立志成为“极客”的初级程序员来说，这仍然是不够的，所以，在附录中我们还提供了一些网址和书单，希望本书能帮你开启职业生涯的新阶段。

2.0 要来了，本书会过时吗？

Angular 2.x 已经进入了 Alpha 测试阶段，那么，不免有人担心，等到 2.x 推出的时候，本书会过时吗？从实现细节上来讲，会的。从思想上来讲，不会。从实用性上来讲，不会。

1.x 和预计 2016 年推出的 2.x 在语法甚至一些底层实现上是截然不同的。

据目前得到的消息，2.x 将使用 TypeScript 和 ES6 作为主体语言，那时候，本书的很多代码将不再适用于 2.x。而由于 2.x 彻底抛弃了 IE11 之前的低版本浏览器，它可以借助最新的浏览器特性进行底层实现，不用为了向后兼容而使用“脏检查”等技术来弥补浏览器的不足。在这些细节上，1.x 和 2.x 几乎没有共通之处。这一点一直被人诟病，也是一些人对 Angular 的前途深表担忧的原因。不过，从另一个角度来看，2.x 的这种改进也是一种勇敢的改革，可以让它轻装前进，更有利于长远发展。

好消息是，2.x 不是 1.x 的替代品。官方已经宣布，即使 2.x 推出，也仍然会对 1.x 进行

长期维护。这就有点类似于 Query 2.x 不再兼容 IE8，而 Query 1.x 仍然兼容 IE8 并继续向前发展一样。这种版本策略可以防止 Angular 背上向老旧浏览器兼容的包袱。

1.x 和 2.x 在编程模型上并没有太大的差异，它们都基于 MVVM 模型，都具有双向绑定功能（即使底层实现方式已经变了），都具有相同的设计哲学——利用高内聚的小模块组合出最终程序。而这些在我们的书中都有所体现。在目录结构、指令的分类等方面，本书也从 2.x 中引入了很多更好的实践。

从实用性上来说，本书更不会过时。2.x 的浏览器兼容性起点就是 IE11，这不是因为细节层面的问题，而是从底层原理上就不可能——它依赖太多的新特性。而在国内市场上，彻底抛弃 IE11 以下的版本恐怕还会是一个长期的历程——即使最乐观的估计，至少也需要两年。当然，手机端的浏览器版本更新要快得多，所以，预计 2.x 最早会被用在手机版上。

固然，2.x 是个高大上的版本，但目前在国内还是个屠龙之术。如果要在现实中使用，还是先学好 1.x 吧。按照本书的指引，你可以提前领略 2.x 的优点，而不用付出兼容性的代价。当然，等本书的 2.x 版推出时，这种熟悉的味道也会让你有一个更高的起点。

阅读指南

Angular 的学习曲线大概是这样的：入门非常容易，中级的时候会发现需要深入理解很多概念，高级的时候需要掌握 Angular 的工作原理，而想成为专家则很难，需要经过很多工程实践的磨练。

本书的主体结构也是针对这样的学习曲线设计的。

首先，初级阶段，实战演练

我们会带你实战中逐步体验 Angular 的开发过程，并随着进度的推进，逐步引入所需的技术和概念。

然后，中级阶段，概念介绍

在实战中提到的一些概念不会就地展开，而是只做简介，到了这个阶段，会对概念进行深入讲解：是什么，为什么，怎么用，什么时候用，什么时候不用等。

接下来，高级阶段，工作原理

学习了这些概念，我们还要把它们串起来，向你揭示 Angular 的工作原理，看看这些概念之间是如何协作的。

最后，专家阶段：最佳实践，技巧

前面主要是入门和理论，而这部分将主要以实战经验为主。

只把 Angular 用熟了是不够的，我们还要把它整合进更宏观的开发过程中，不但要考虑开发，更要考虑维护。我们要如何开发容易维护的 Angular 程序？请看第 4 章。

专家还需要掌握一些技巧去把复杂问题简单化，发掘一些不常用但很有用的 API，看起来平淡无奇的框架特性运用得出神入化，第 5 章将集中展现这一点。

坑

在前面的章节中零零散散提到了一些需要注意的地方，但是这样不方便查阅，所以我们把需要注意的地方作为独立的一章，把我们帮别人解决过的一些典型问题收集在一起。当然，我们也会在读者社区继续维护并更新这些“坑”，而不是等再版时才发布。我们希望能把这本书做成“活的”，让这本书更加物超所值，不辜负读者对我们的信任。

工具

工欲善其事，必先利其器。充分发挥工具的力量是开发人员的重要素质，日常用到的工具你真的用熟练了吗？有没有更好的工具？我们会把实战中觉得对自己帮助最大的工具及其使用经验分享给你。

更多

在实战中，有很多需求是不显眼但很重要的，比如 SEO、访问统计等，在实际项目中，这些往往是不能忽视的。我们会专门通过一章来讲解如何结合 Angular 和第三方软件干净漂亮地解决这些问题。

Hybrid 应用和手机 Web 越来越普及，手机版开发的需求也越来越高，在 Angular 的基础上，开发手机版变得容易多了。而且，已经有了比较成熟的工具和框架，我们会简要讲解一下手机版开发的方法和框架。

附录

软件开发需要很多综合技能，但本书容量有限，我们也不可能是每个领域的专家。因此，我们会“授人以渔”，给出一些在线资源和书单，供大家深入学习或作为备查资料。

关于随书代码

书中所摘录的只是全部代码的一小部分，大部分代码都放在了 GitHub 上。地址是 <https://github.com/ng-nice/code>。

如果你查看 GitHub 历史，会发现总的提交数并不多。这是因为要方便教学，所以在提交前进行了合并。所保留的这些提交大都和书中的主要进度有关，略去了细节提交。所以，本书中代码的提交粒度不能代表实际项目中的提交粒度，在实际项目中，其提交粒度通常比本书中所示范的更小。阅读代码时请记住这一点，以免养成“大粒度提交”的坏习惯。

另外，文中的 JavaScript 代码（包括摘引的 Angular 源码）全都使用了两格缩进模式，这主要是考虑到图书排版问题，希望少一些不必要的换行。你们在现实项目中愿意用两格或四格均可，只要项目组内保持一致即可。

关于内容的重复

仔细阅读，可能会发现有些内容会在多个不同的章节中重复讲解，这当然不是凑字数，而是尽可能符合人的记忆规律——把重要的内容在不同的场景下重复，对于深入掌握重点是很有帮助的。

关于写作风格

这是一本多人协作的书，虽然我们进行了后期统稿，但在语言风格等方面仍难免会有不一致的地方，我们期待你们的反馈，以便将来改进。

你的好，我永远记得！

双狼的感恩

双狼的本次合作起于机械工业出版社编辑吴怡的邀请。作为 ThoughtWorks 的 Tech Lead，双狼都有很多工作任务，原定 6 个月的写书计划，被拖到了 8 个月，感谢吴怡的耐心与推动。

还有很多 ThoughtWorker 为本书做出了贡献：

- ❑ 张逸，资深 ThoughtWorker，很多技术书籍的作者或译者。一直在鼓励我们，并给了我们很多帮助。
- ❑ 彭洪伟，本书的第三作者。在交稿压力最大的时候，承担了“工具”篇的撰写工作，保障了本书的尽早交稿。
- ❑ 陈嘉，幕后的贡献者，全栈式工程师。帮我们设计了“双狼说”微信公众号的 Logo，从技术的角度帮我们审稿，并提了一些非常有用的建议。

还有很多 ThoughtWorker 和社区朋友帮助我们从技术层面和语言层面进行修改。他们有的是 Angular 专家，有的是新手，给了我们比较全面的反馈。能将枯燥、乏味的技术平易近人地展现在这本书中，一定要感谢他们所作出的奉献。他们是（排名不分先后）：冯尔东、朱本威、李科伟、杨琛、彭琰、叶志敏、ng 群 as。

还要感谢 Angular 中文社区 QQ 群和关注“双狼说”微信号的网友们，是你们的鼓励给

了我们写作的信心和动力！

雪狼的感恩

开始写书的时候，刚刚认识我的女友娜娜，今天，我们即将走进婚姻的殿堂。我这样一个负情商的程序员，生活有多么枯燥乏味，不问可知。感谢你点亮了我的人生。你的好，我永远记得！

能专注开发 17 年，要感谢我父母和弟弟的支持。人到中年，本应是最纠结的时代，特别是我这样的前“单身狗”。我无法经常回家，是弟弟经常回去探望父母。父母的乐观与健康，让我可以心无旁骛地工作。你们的好，我永远记得！

能走入软件开发这一行，要感谢我的伯乐何战涛和王勇的帮助。还记得那个沉默而不自信的“小汪”吗？当初，他什么也不会，犯过很多错误；如今，他在尽力为别人的职业生涯提供帮助。你们的好，我永远记得！

在这 17 年间，我尝试过很多角色，从写文档、测试到小公司的 CTO，走过了丰富多彩的人生。特别需要感谢的是林先生和余姐等前同事，我们追随林先生走过 8 年创业之旅，这一过程让我具备了更开阔的视野和更坚韧的性格。虽然因为生活压力不得不离开，但，你们的好，我永远记得！

最后，感谢 ThoughtWorks！作为“敏捷”的倡导者，ThoughtWorks 处处体现着敏捷思想，这是一个把“敏捷”变成“文化”并渗透到骨子里的公司。这是一个很“奇葩”的公司。这一点从“全球 CEO 和中国区程序员撞衫”事件就可见一斑。这是一个很“简单”的公司。引用我们一位 HR 的说法：“进了 ThoughtWorks，我感觉自己的情商都下降了！”这是一个很“技术”的公司。每年两期的技术雷达都来自全球近 3000 名工程师的实践总结。这里几乎会涉足每一项前沿技术：大数据、React、Scala 等。仅以 Angular 为例，我们在工程实践中使用它是在 4 年前，那时候 Angular 还是 0.6 版呢。

还差一句话就变成招聘软文了，索性补上吧：ThoughtWorks，你的好，进来才知道！

破狼的感恩

在写作本书之际，我作为 ThoughtWorks 高级敏捷咨询师、架构师，因为项目曾辗转多地，没有太多时间投入到这次的写作之中。在此之前，已经有很多的出版社联系我写本国人的深度解析 Angular 的书籍，但都被我婉言拒绝了。直到雪狼告诉我他希望写一本关于 Angular 的书籍的时候，恰巧吴怡编辑也跟我提起了写书的事。这次我们应承下来了。在这里首先感谢吴怡的知遇之情和写书过程之中的包容与耐心。还有雪狼大叔的最终推动。

我是一个具有承诺“强迫症”的人，一旦应承下来的事情，我就会尽自己最大的努力把它做好。特别在快截稿的几个月里，每天写作到凌晨 1 ~ 2 点，次日还需要准备与客户 7:00 的站会。和老婆结婚已经一年多了，可是由于工作原因，我也出差在外一年多了。加上写作此书的时候，连陪伴父母和老婆的时间也被我挤出来写作本书了。所以在此，首先要感谢我的父母和老婆，感谢你们的支持和包容，在这本书的背后也包含着你们对我的一份宝贵的爱。爸妈、老婆：我也爱你们！

还要感谢我大学的导师刘继光老师和柳翠寅老师，是你们让我学到了软件开发的技能，以及帮我寻找到一份实习的机会，给了我比别人多 3 年的实战机会，因此我才能独自闯荡成都这座大城市，并开始了软件开发之路。还有你们对知识的追求和坚强的毅力，深深感染了我。至此坚持了 6 年多的博客写作和回馈开源社区，因此才有了本书的顺利完成。你们是我一辈子的良师益友，谢谢你们的付出！

同时还要感谢我在 ThoughtWorks 的同事们。是我的 Sponsor 张逸一直鼓励我写书，是熊节每年的阅读数量激励我更加坚定地持续获取更多的知识来武装自己。还有很多的 twer，同样在这里感谢你们长久以来对我的帮助和鼓励。

我常常告诉自己：要么读书，要么旅行，身体和心灵总有一个在路上。与君共勉。

彭洪伟的感恩

首先，我要感谢我父母！即使是在家境最困难的时候，他们也时时刻刻鼓励我、支持我，让我能坚持走自己的路。

我第一次接触到计算机是在 2003 年。那时刚上初中，正是 CS 和传奇火爆的时候，想自己申请一个 QQ 号都感觉很困难。直到 2006 年初中毕业，莫名其妙地开始了信息学奥林匹克竞赛的培训之路，用 C 语言写出了人生的第一行 Hello World 代码。

感谢张宗弋老师的悉心教导和高中三年对 C 语言、算法、图论、数论的培训，还记得大家都叫您小弋（误读作：Ge）子。是您激发了我的“程序员梦”。以至于后来在报志愿的时候，第一志愿写的都是计算机信息科学与技术，虽然那时候并不知道这个专业是干什么的。

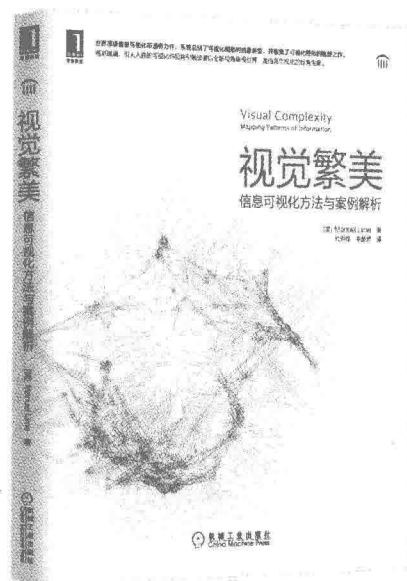
在大学里，特别要感谢的是邓芳老师，是您鼓励我们，除了学好课堂上的东西，更要积极去探索自己感兴趣的東西。您“授人以鱼，不如授人以渔”的教学思想，让我逐渐养成了自学的习惯，至今让我受益匪浅。后来，您帮我引荐了两位让我受益良多的导师：陈宇副教授和王海军副教授。是他们把我带到了物联网的领域，让我对计算机的领域产生了更浓厚的兴趣。三位老师的指引，让我接触到了更多课堂上没有的东西，终于有幸加入

ThoughtWorks 这个大家庭中。

在这里，我认识了有黑客范儿的马伟和段子手 Jojo（周哲武），是他们给了我建设性的意见，让我感觉到了 ThoughtWorks 的不同。不得不提的还有 Jojo 帮我推荐的 Sponsor——破狼，是他不断地在开源的世界里给我指出新机会，鼓励我大胆尝试。不然我也不能坚持翻译完 Scala 构建工具 SBT：<http://www.scala-sbt.org/0.13/tutorial/zh-cn/index.html> 的使用文档。如果没有破狼的引荐，我也不会认识前端牛人雪狼和参与本书的编写。

最后，如果没有未婚妻韩盼盼的爱、支持与鼓励，我也难以完成这么多富有挑战性的工作。谢谢，我爱你！

推荐阅读



视觉繁美：信息可视化方法与案例解析

作者：（美）Manuel Lima 书号：978-7-111-42077-4 定价：79.00元

世界顶级信息可视化布道者力作，
系统总结了可视化图形的抽象类型和著名作品。
精彩斑斓、引人入胜的可视化作品将引领读者以全新视角审视世界，
是信息可视化的经典宝库。

《视觉繁美》为我们展现了艺术、设计与科学三个领域如何相互交融。其中既有略显稚嫩的信息图，也不乏意味深远、视觉精美的佳作。书中的信息图引人入胜，展现了数据可视化的无限可能性。

——《纽约时报》书评论

从中世纪挂毯之中的家谱到分形图背后的数学原理，《视觉繁美》的作者展示了设计师如何把复杂模式转变为引人注目的艺术作品。

——《连线》杂志书评

Web前端开发&设计经典

框架篇

