

The
Pragmatic
Programmers

Seven Web Frameworks in Seven Weeks

Adventures in Better Web Apps

七周七 Web开发框架



[美] Jack Moffitt Fred Daoud 著

张霄翀 邱俊涛 孙镌宸 顾宇 译

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

The
Pragmatic
Programmers

Seven Web Frameworks in Seven Weeks

Adventures in Better Web Apps

七周七 Web开发框架



[美] Jack Moffitt Fred Daoud 著

张霄翀 邱俊涛 孙镌宸 顾宇 译

人 民 邮 电 出 版 社

北 京

图书在版编目 (C I P) 数据

七周七Web开发框架 / (美) 墨菲特 (Moffitt, J.),
(美) 达乌德 (Daoud, F.) 著 ; 张霄翀等译. -- 北京 :
人民邮电出版社, 2015. 8
ISBN 978-7-115-38843-8

I. ①七… II. ①墨… ②达… ③张… III. ①网页制
作工具—程序设计 IV. ①TP393.092

中国版本图书馆CIP数据核字(2015)第121154号

版权声明

Simplified Chinese-language edition Copyright © 2015 by Posts & Telecom Press. All rights reserved.
Copyright © 2014 The Pragmatic Programmers, LLC. Original English language edition, entitled Seven Databases
in Seven Weeks.

本书中文简体字版由 The Pragmatic Programmers, LLC 授权人民邮电出版社独家出版。未经出版者书面许可,
不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

-
- ◆ 著 [美] Jack Moffitt Fred Daoud
译 张霄翀 邱俊涛 孙镌宸 顾 宇
责任编辑 陈冀康
责任印制 张佳莹 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>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 19
字数: 329 千字 2015 年 8 月第 1 版
印数: 1—3 000 册 2015 年 8 月北京第 1 次印刷

著作权合同登记号 图字: 01-2014-5440 号

定价: 59.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

内容提要

本书带领读者认识和学习 7 种影响现代 Web 应用并改变了 Web 开发方式的框架，以期给 Web 开发者带来启发和思考。

本书延续了同系列的畅销书《七周七语言》和《七周七数据库》的体例和风格。全书共 8 章，前 7 章介绍了 Sinatra、CanJS、AngularJS、Ring、Webmachine、Yesod 和 Immutant 共计 7 种 Web 开发框架，最后一章总结回顾了所有的知识点。书中对每一种框架的介绍，都为构建 Web 应用带来了独特而强大的思路。除此之外，书中还提供了一系列代码示例和在线资源以供参考。

本书适合有一定基础的 Web 开发人员阅读，能够帮助读者拓宽思路，激发更多的设计灵感。

序

2003 年，我带着家人乘坐去往科罗拉多州杜兰戈市的火车。在寸土寸金的年代里，狭窄的红色砂崖中狭窄的轨距曾经运转良好。而现在，火车变成了历史的遗迹，已经完全被更安全高效的汽车、飞机所取代。逝者如斯。

今天也一样，我们见证了变革。单核的计算机已死，或者正在死去。不错，作为它们的继承者，多核技术是一个奇迹。同时，它们也是一个巨大的技术挑战。我们已经熟知且依赖的编程语言不再按照预期的工作了，结果就是我们看到了新一代的编程语言正在诞生。到目前为止，还没有好心人宣布获胜者。

在这个背景下，我在 2010 年编写了《七周七语言》这本书。说实话，我没有期望会卖多少本。毕竟那只是一个在 Java 世界中讨论各种语言的书，也是一本在一切都被面向对象所控制的世界中讨论多种编程范式的书而已。但是程序员逐渐意识到了技术停滞带来的危险，并且接受学习编程语言是为了让人更聪明，从而能够更好地应对变化的理念。从这个方面来看，那本书是一个巨大的成功。

三年过去了，虽然函数式编程已经开始获得一些关注，但是仍然没有领导者。我们发现，多年之前扔进我们虚拟池中的多核芯片开始荡起一些涟漪。

仅仅将去年的窄轨道拓宽是不够的。在本书中，Fred 和 Jack 会向你展示前沿的人们如何将 Web 开发朝着它本应该的方向上发展。你会看到一个叫作 Sinatra 的传统的面向对象框架，然后你会朝着客户端进发，那里 JavaScript 正在编织传奇。你将会通过 CanJS 和 AngularJS 来学习如何实现完整的、富客户端的开发。接下来，你将进入服务器端，看看函数式编程的强大能力。你会遇到两个 Clojure 的框架：极简主义的 Ring 和以健壮著称的 Immutable。你会看到 Erlang 中基于状态机的 Webmachine。如果这还不足以颠覆你的想法，你会发现强大到不可思议的 Haskell 框架 Yesod。

“七周七某某”这套书就是用来拓展你的思维的。我非常自豪地为你带来这个系

列的下一本：《七周七 Web 开发框架》。衷心地希望本书能带你超越任何阻碍。

谨致问候

Bruce Tate

首席技术官，icanmakeitbetter.com

作者访谈

Q: 你们为什么要写这本书?

Jack: 解决问题的方法有很多种, 而程序员不断地开发出新技术, 构建出新语言以找到更好的解决方案。在编程的不同方面存在众多不同的想法, 多年来, 在感受过多次大开眼界的震撼之后, 我很想与人分享自己的经历。

构建一个 Web 应用的方式不止一种, 甚至连我自己经手过的项目都没有哪两个用的是同样的方式开发的。Web 编程拥有数百种可供选择的框架和库, 而传统的 GUI 编程却只有屈指可数的几种选择。如此多的可能性, 却没有哪个是完美的, 所以我想探索那些最有意思的思路和技术, 然后分享给更多的读者。

Fred: 不论是通常所说的编程还是具体的 Web 开发都在以惊人的速度进化。我觉得本书是一个绝佳的机会来打破主流, 探索新的想法并发现 Web 开发中的不同方式。

选择这些框架的目的并不是想比较它们, 也不是为了让你在下个项目中选择它们。这本书更像是一次对 Web 开发中各种不同特性的框架的探索。

Q: 为什么选了这 7 个框架呢?

Jack: 我们希望能挑选出那些有着独特而强大思想的框架, 而不仅仅是那些已经备受关注的框架。我们在本书中探索了极简主义、可组合、静态类型、状态机和声明式语法等。

在某些情况下, 我们选择一些框架是源于我们想要探索的一些想法, 而另一些则只是同类框架中最简明的例子。而且我们选择了一些互相重叠最少的框架, 以确保覆盖尽可能多的思路。

Fred: 在受够了那些过于复杂的所谓“企业级”框架之后, 探索学习那些可以通

过恰如其分的代码来进行编码的框架可谓是令人心旷神怡。

我们还设想引入客户端的框架，因为客户端的 JavaScript 已经不再是代码碎片的大杂烩了，这些框架给了我们编写模块化和组织良好代码的机会。

Q：你们还想引入哪些框架？

Jack：我觉得 Play 这个框架和 Yesod 旗鼓相当，它使用 Scala 来支持静态类型，并且还有一些其他特性。我还想将基于 Elixir 的框架加进来，但是目前还没有任何一个成熟的框架，因为 Elixir 本身也还在不停的修改中。

在另一个平行宇宙中，整本书可能都是关于前端框架的，仅仅前端就足以带来足够有意思的思想。一方面，一些非常优秀且独特的 ClojureScript 框架正在不断涌现，比如 Webfui 和 Om；另一方面，Meteor 和 Derby 也在实时和协作应用中大放异彩。

Fred：Node.js 是一个本应该被包含进来的有趣的服务器端框架，但是我们更想引入一些客户端框架。当然，我们可以写一本完全使用 JavaScript 的书，但这并不是我们的目标。

Q：你们开始写这本书之后，有没有出现一些有趣的新框架呢？

Jack：我的 RSS 阅读器里存的全是关于各种新框架的文章，而且每天都会有更新。目前我打算研究的几个包括 Revel（一个使用 Go 编写的框架）和 Om（一个构建在 core.async 之上的 ClojureScript 框架）。虽然 Elixir 是一个很新的语言，但是基于它的 Web 框架已经开始不断涌现，比如 Sugar 和 Dynamo，这些我也都要尝试一下。

Fred：新的框架不断涌现，我们都快跟不上了。由于 Clojure 是我最喜爱的编程语言，我想要探索的两个框架分别为 total.js 和 Pedestal。

Q：学习这些都需要什么？

你需要 Windows、MacOSX 或者 Linux，还有你最喜欢的浏览器。每一章都会告诉你下载哪些工具以及对应的编程语言版本。

前言

一般而言，在开发 Web 应用时，往往在一段时间之后我们会想要用另外一种方式来实现，或者使用更好的工具来实现。虽然没有完美的框架，但是探索其他框架时展开的思路，不仅可以满足应用本身的需求，而且还可以在极大程度上帮助我们使用现有工具以不同的方式来解决问題。

这本书记录了一些我们在研究开发应用程序的新思路时的一些探索。我们希望你从这个现代，而又没有经过太多探索的 Web 编程世界找到乐趣。

为什么是 7 个框架

你很可能已经有一个用来完成手头工作的框架，或者已经习惯于某个框架。你可能喜欢它，也可能讨厌它，但是你还想知道世界上有没有更好的框架。即使你并不想换一个框架或者学习一门新语言，我们认为探索一下其他开发者的好想法也会积极地影响你的工作和思维方式。

我们是对新想法和编程充满热情的终生学习者。在有如此众多的 Web 框架和编程语言的今天，非常容易就可以获得很多快乐，学习有趣而新鲜的事物，你不大可能会觉得无聊。我们在自己的职业生涯中经历了许多框架，其中一些变成了我们的新宠，另外一些则给了我们启发，还有少数则给我们本来就熟悉的领域带来了新的思路。

这本书想要带给你对 7 种截然不同的 Web 框架的感性认识，既为你展现这些框架的关键思路，又引起你的好奇心和探索精神。我们探索的每个框架都具有某些独一无二的特质。与主流框架相比，它们是更少有人走的路，而且这些路上充满了欢乐和惊喜、探索和激励。

关于本书

这本书是 Pragmatic Bookshelf 丛书的“七周七某某”系列中的一本，这个系列还包括《七周七语言》、《七周七数据库》。本书中的每一章都讨论了一个不同的 Web 框架，而且大部分都是不同的编程语言，目的是为你提供一些开发现代 Web 应用程序的全新理念、风格和技术。

每一章都是独立的，并且都被组织成了 3 天的形式。其中我们会介绍框架，展现其独一无二的特性。章节之间并没有特别的顺序，阅读时也不需要按照特定的顺序，可以直接跳到自己感兴趣的章节开始阅读。

每个框架都因为某些独一无二的特质而被选中，当然这与它们的流行程度并没有太大关系。你可能会发现一些不论是语言本身还是框架你都没听过，但是有时候这正是那些最好的想法的藏身之所。

我们从第 1 章开始，介绍了 Ruby 世界中最为简单的框架之一 Sinatra。我们在探索这个小巧而精致的框架的过程中，会创建并测试一个书签应用。

在第 2 章，我们会学习 Web 应用的新趋势：使用基于 JavaScript 的客户端框架 CanJS，并使用 Sinatra 作为后端。我们重新实现一个书签应用，并在过程中展示可以被观察和响应的动态模型。

在第 3 章，我们介绍 AngularJS。这是另外一个客户端的 JavaScript 框架，但是使用了一种完全不同的风格。AngularJS 是声明式的，并且可以与你的 HTML 集成在一起。你只需要告诉它你想要什么，而不是如何完成。

Lisp 程序员常常会说“代码即数据”，我们在第 4 章介绍 Ring 时，你会看到 Web 应用程序也是数据。Ring 应用运行在一个既复杂又简单的抽象层上，并从函数式编程中受益。

你之前对 Web 应用程序如何工作的理解将会在第 5 章受到挑战，这一章介绍了 Webmachine，这个基于 Erlang 的框架将 HTTP 建模为一个状态机。这可以让你充分利用到这个协议的优势，而这一点恰好是其他框架所欠缺的。

第 6 章，介绍了 Yesod。通过 Haskell 的强类型、静态类型系统，避免了很多 Web

应用程序错误。如果你有损坏的连接或者没有很好地处理用户生成的内容，那么编译就会失败。

最后即第 7 章，介绍了 Immutant 框架，通过用 Clojure 包装 JBoss 系统，以及删除一些不必要的组件来重塑企业级 Java Web 框架。其结果就是产生一个你会欣然使用的企业级功能的组合。

这本书不是什么

在一本书中引入如此多的想法很难做到公正，于是我们不得不缩减一些你可能想要看到的功能，那些功能需要专门介绍该语言或者框架的书来覆盖。

不是一本 Web 编程教程

我们假设你已经对 Web 应用开发比较熟悉了，我们没有解释 HTML、CSS 以及一些 Web 应用程序的基础知识。你可能之前已经做过一两个 Web 应用了，如果没有，也不用担心，我们假设的 Web 知识其实都非常基础。

不是一本编程语言教程

我们用 5 个不同的编程语言介绍了 7 种不同的 Web 框架。其中有些语言你可能已经很熟悉了，比如 Ruby、JavaScript，另外一些则非常奇怪。限于篇幅，本书中并没有引入语言的简介，但是我们也考虑到了第一次接触这些语言的读者。即使你之前没有接触过这些语言，也应该很容易获取框架想要表达的思路。这些思路适用于任何语言。

不是一本安装或者部署手册

安装语言和框架变得越来越容易了，为了保持每一章都聚焦，我们并没有引入太多的安装或者部署的细节。大多数情况下，包管理器 and 构建工具会搞定这些烦琐的工作，如果你遇到了问题，请在搜索引擎中查找相关的在线文档。

示例代码及体例

我们想要在每一章中都覆盖尽可能多的内容，但是在有些情况下，我们省略了一些与正在解释的主题无关的代码，而这些代码又是正常运行应用程序所必需的。有时这些代码是由脚手架自动生成的，有时你需要自己下载代码包。每章应用里的代码都提供完整的代码包下载，请放手在下载的代码中进行修改，那样就不用什么都要从头手敲一遍了。

对于每种语言，我们都尽可能地使用了当时社区中最为流行的约定和工具。

在线资源

书中的应用程序和实例都可以在“Pragmatic Programmer”网站¹上找到。你还可以找到一个论坛和勘误表单，可以报告发现的问题，或者为将来的版本提一些建议。

我们希望你享受学习这 7 个不同的框架的探索之路，也希望这些框架中的好想法能给你更多的启发。

Jack Moffitt 和 Fred Daoud

2014 年 12 月

¹ <http://pragprog.com/book/7web/seven-web-frameworks-in-seven-weeks>

对本书的赞誉

如果只看标题，你会以为这本书只是对不同技术进行了广度优先的分析。但是令人惊讶的是它同时在深度上的展开，对每一个不同主题的核心都做了足够的强调。如果你是一个多语言狂人，或者准备成为一个，那么这本书正是写给你的。

Jim Crossley

Immutable 团队核心成员，Red Hat 首席软件设计师

客观且清晰！不仅仅是一个介绍，更是一个好的起点。任何一个现代程序员都需要这样的广度和深度，我强烈推荐本书。

Pablo Aguiar

软件工程咨询师

这本书非常有意思，对每个框架的介绍都快速而清晰，作者不仅快速地带你学习了每个框架，而且又令人惊奇地详细品味了每个框架的主要功能：设计哲学、实现、测试以及对进一步探索的提示。本书包含两个 JavaScript 框架、一个 Ruby 框架、一个 Haskell 框架、两个 Clojure 框架和一个 Erlang 框架。如果你喜欢 Web 开发，则一定会喜欢上这本书的。

Giles Bowkett

高级开发，知名博主

我非常喜欢读这本书。事实上，Yesod 这一章给了我很多关于如何向非 Haskell 程序员介绍强类型语言的强大之处的全新想法。

Michael Snoyman

Yesod 的发明者，FP Complete 公司首席软件工程师

致 谢

我们要感谢使本书得以出版的“Pragmatic Bookshelf”团队。特别感谢我们的编辑 Jackie Carter，是她以自己的专业性和不懈的努力让本书变得更好，并支持到了最后。感谢 Bruce Tate，我们两个都是他的书的忠实读者，我们因能追随他的脚步而感到荣幸。感谢 Andy Hunt 和 Dave Thomas，他们创建了一个我们都感到非常兴奋的技术主题。

感谢下面这些技术审阅者，他们为每个框架都贡献了自己专业的建议：Konstantin Haase (Sinatra), David Luecke (CanJS), Miš ko Hevery (AngularJS), James Reeves (Ring), Justin Sheehy (Webmachine), Michael Snoyman (Yesod), Jim Crossley 和 Toby Crawley (Immutable)。此外，还有为多个章节都贡献了自己的评论和建议的审阅者：Kimberly Hagen、Kevin Wiley、Pablo Aguiar、Mick Thompson、Christopher Zorn、Nathaniel Schutta，以及 Aaron Bedra。

如果没有框架的作者们，这些如此有创新意义的框架也就不会存在，感谢 Blake Mizerany、Justin Meyer、Miš ko Hevery、Adam Abrons、Justin Sheehy、Andy Gross、Mark McGranaghan、James Reeves、Jim Crossley、Toby Crawley、Michael Snoyman，以及他们的团队和贡献者。

感谢那些在本书还在 beta 版本就帮助勘误的读者，他们使得本书变得更好。

Jack: 我要感谢我的妻子 Kim，她鼓励我编写本书，我的各种想法都先试讲给她，她还花费了很多时间来审阅本书。感谢我的两个孩子 Beatrix 和 Jasper，他们带给了我很多欢乐。感谢 Sean Johnson，他介绍了 Bruce 给我，而 Bruce 启动了整个项目。

Fred: 感谢我的妻子 Nadia，她在各个方面都给我以支持，生活中有她真好。感谢 Lily 和 Ruby，他们为这个家增添了如此多的欢乐和精彩。

很多年后，我将带着叹息在一个未知的远方向人诉说：林中曾有一条两岔路，而我选择了那条更少人走的路，它改变了所有的一切。

Robert Frost

目 录

第 1 章 Sinatra1	
1.1 简单的领域特定语言2	
1.2 第 1 天: 构建一个书签应用.....2	
1.2.1 你好, Sinatra.....3	
1.2.2 用 RSpec 来测试.....4	
1.2.3 REST 风格的 API.....6	
1.2.4 数据持久化6	
1.2.5 创建和读取书签8	
1.2.6 编写自动化测试9	
1.2.7 更新和删除书签10	
1.2.8 我们在第 1 天学到的...11	
1.3 第 2 天: 创建视图.....12	
1.3.1 ERB 简介12	
1.3.2 Mustache 介绍.....17	
1.3.3 Slim 介绍.....20	
1.3.4 我们在第 2 天学到的...23	
1.4 第 3 天: 添加新功能24	
1.4.1 校验24	
1.4.2 块参数26	
1.4.3 过滤器27	
1.4.4 为书签打上标签28	
1.4.5 添加标签的 API 支持...31	
1.4.6 使用正则表达式来匹配 路由33	
	1.4.7 我们在第 3 天学到的... 34
	1.5 总结 35
	1.5.1 Sinatra 的强项 35
	1.5.2 Sinatra 的弱项 35
	1.5.3 最后的思考..... 36
第 2 章 CanJS 37	
2.1 CanJS 的独一无二之处 37	
2.2 第 1 天: 创建对象和同步 变化 39	
2.2.1 你好, CanJS 40	
2.2.2 构建和扩展对象..... 41	
2.2.3 观察属性的变化..... 44	
2.2.4 使用 CanJS 创建一个 前端书签应用..... 46	
2.2.5 连接模型与服务器..... 47	
2.2.6 渲染视图..... 48	
2.2.7 动态绑定..... 50	
2.2.8 我们在第 1 天学到的... 50	
2.3 第 2 天: 创建控制器..... 51	
2.3.1 将控制器绑定到页面 元素上..... 52	
2.3.2 监听 UI 事件 53	
2.3.3 使用 data()方法从页面 获取数据模型..... 54	

2.3.4 使用观察者实现控制器 之间的沟通	55	3.3.2 利用数据双向绑定的 优势	97
2.3.5 创建一个表单控制器	57	3.3.3 创建书签表单	98
2.3.6 我们在第 2 天学到的	60	3.3.4 关于作用域的重要注意 事项	100
2.4 第 3 天: 与模型的协作	61	3.3.5 端到端的自动化测试	102
2.4.1 添加校验	62	3.3.6 我们在第 2 天学到的	104
2.4.2 实现标签的处理	64	3.4 第 3 天: 创建过滤器和 路由	105
2.4.3 过滤书签	66	3.4.1 为书签添加标签	106
2.4.4 创建一个标签列表	69	3.4.2 构建一个标签列表	107
2.4.5 使用路由管理浏览器的 位置	71	3.4.3 通过过滤器操作数据	108
2.4.6 我们在第 3 天学到的	73	3.4.4 定义路由	112
2.4.7 对 CanJS 的创造者 Justin B. Meyer 的采访	74	3.4.5 我们在第 3 天学到的	114
2.5 总结	76	3.4.6 对 AngularJS 创建者 Miško Hevery 的采访	115
2.5.1 CanJS 的强项	76	3.5 总结	116
2.5.2 CanJS 的弱项	76	3.5.1 AngularJS 的强项	116
2.5.3 最后的思考	76	3.5.2 AngularJS 的弱项	117
第 3 章 AngularJS	77	3.5.3 最后的思考	117
3.1 概览	77	第 4 章 Ring	118
3.2 第 1 天: 使用依赖注入	79	4.1 Ring 简介	118
3.2.1 你好, AngularJS	80	4.2 第 1 天: 基础组件	120
3.2.2 创建服务	83	4.2.1 起步	121
3.2.3 换个角度来看我们的 书签应用前端	86	4.2.2 Hello, World!	121
3.2.4 使用资源服务	87	4.2.3 用 Korma 处理数据	124
3.2.5 为服务写自动化测试	89	4.2.4 用 Hiccup 把数据转化 为 HTML	130
3.2.6 我们在第 1 天学到的	93	4.2.5 使用 Compojure 处理 路由	133
3.3 第 2 天: 创建控制器和 视图	93	4.2.6 我们在第 1 天学 到的	136
3.3.1 创建控制器和使用视图 指令	95		