

O'REILLY®

TURING

图灵程序设计丛书



# JSON 实战

JSON at Work

涵盖JSON基础知识、操作实践与案例  
全面掌握JSON强大功能的明智之选

[美] 汤姆·马尔斯 著  
邵钊 译

 中国工信出版集团

 人民邮电出版社  
POSTS & TELECOM PRESS

TURING

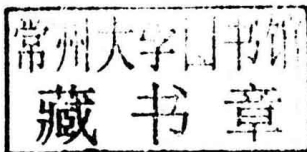
图灵程序设计丛书



# JSON实战

JSON at Work

[美] 汤姆·马尔斯 著  
邵钊 译



Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社  
北京

## 图书在版编目 (C I P) 数据

JSON实战 / (美) 汤姆·马尔斯 (Tom Marrs) 著 ;  
邵钊译. — 北京 : 人民邮电出版社, 2018. 7  
(图灵程序设计丛书)  
ISBN 978-7-115-48555-7

I. ①J… II. ①汤… ②邵… III. ①JAVA语言—程序  
设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第111633号

## 内 容 提 要

本书来自于作者实际使用 JSON 的经验所得, 主要内容包括 JSON 基础知识, 对 JSON 数据建模, 在 Node.js、Ruby on Rails 和 Java 中使用 JSON, 结构化 JSON 文档并设计测试 API, 搜索 JSON 文档的内容, 将 JSON 文档转换成其他数据格式, 将 JSON 作为企业级架构中的一部分来使用, 等等。

本书适合对 Web 和移动端应用、RESTful API 以及消息系统进行设计或实现的架构师和开发人员阅读。

- 
- ◆ 著 [美] 汤姆·马尔斯
  - 译 邵 钊
  - 责任编辑 朱 巍
  - 责任印制 周昇亮
  - ◆ 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: 18.75
  - 字数: 443千字 2018年7月第1版
  - 印数: 1-3 500册 2018年7月北京第1次印刷
  - 著作权合同登记号 图字: 01-2018-3441号
- 

定价: 89.00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: 京东工商广登字 20170147 号

站在巨人的肩上  
**Standing on Shoulders of Giants**



[iTuring.cn](http://iTuring.cn)

---

# 版权声明

© 2017 by Vertical Slice, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2018. Authorized translation of the English edition, 2017 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2017。

简体中文版由人民邮电出版社出版，2018。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

---

# O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 *Make* 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过图书出版、在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

---

# 献词

致在 Web/ 移动端应用程序、REST API 和消息系统中提供或使用 JSON 数据的所有人，希望本书能让你的工作变得更加简单。

致 JSON 社区中提供工具和类库的无名英雄，感谢你们为 JSON 所付出的艰辛劳动。

# 前言

JavaScript 对象表示法 (JavaScript Object Notation, JSON) 已经成为 RESTful 接口设计中的事实标准, 架构师和开发人员可以使用一整套现成的技术生态系统 (鲜为人知的标准、工具和相关技术) 来搭建设计精巧的应用程序。JSON 不仅仅是 Ajax 调用中 XML 的一个简单替代品, 它也正日益成为互联网数据交换领域的骨干元素。严谨的标准和技术最佳实践加上对 JSON 的热爱, 有助于我们搭建一个真正优雅、有用而又高效的应用程序。

唯一的缺憾是, 没有一本书将这一切串连起来进行介绍。本书旨在帮助开发人员使用 JSON, 以搭建企业级的应用程序与服务。我们的目标是促进 JSON 工具的使用, 同时力图让消息 / 文档设计这一理念在日新月异的 API 社区中成为“一等公民”。

我和 JSON 的接触始于 2007 年, 当时我正在负责一个大型的 Web 门户项目, 而该项目要求实现拥有几千个选项的下拉列表。那时我刚好在阅读 Rebecca Riordan 所著的《Head First Ajax (中文版)》, 因此设计了比较优雅的架构方案。Ajax 能够解决延迟和页面加载问题, 但是该如何处理数据呢? 前几年我一直在使用 XML 技术而且很成功, 但对于将数据从 Web 应用程序后端传输到前端展现层这样的任务, 继续使用 XML 技术显得有些大炮打蚊子。《Head First Ajax (中文版)》中提到了名为 JSON 的一种新数据格式, 而这一策略看上去似乎是可行的。我的整个团队开始研究能将 Java 对象转换为 JSON 的 API, 并最终选择了 JUnit 测试程序最简短的方案, 我们的目标是在代码有效的前提下, 尽可能简化所需的工作。我们对完成后的应用程序执行了严格的压力测试, 而从 Java 转换为 JSON 的操作在测试中从未成为性能瓶颈。最终, 这一应用程序在生产环境中呈现出了很好的可扩展性, 用户也能瞬间看到下拉列表。

之后的一段时间里, 我思考过在 Web 应用程序、RESTful API 和消息系统中均使用 JSON。2009 年, 因为 XML Schema 可以在数据交换过程中提供语义校验, 所以我仍旧在项目中继续使用 XML。当时我的技术决策是这样的: 在 Web 用户界面上使用 JSON (出于速度考虑), 在 Web Service 和消息系统中则使用 XML (出于数据集成考虑)。不过, 当 2010 年听说 JSON Schema 后, 我就意识到自己已经不再需要 XML 了。JSON Schema 标准目前还在完善中, 但已经足够成熟, 足以用于企业级应用程序中的数据集成任务。

时至今日, 我已经习惯, 或者更准确地说, 迷上了 JSON。我开始在网络上搜索 JSON 的



其他功能，并发现了大量的 API、在线工具、内容搜索功能等。简而言之，能够使用 XML 实现的功能都可以（也应当）用 JSON 实现。

之后我开始搜索有关 JSON 的图书，却失望地发现只能在讲述 JavaScript 或者 RESTful Web Service 的书中，零星地找到一两章有关 JSON 的内容。JSON 社区欣欣向荣，拥有大量的支持工具、文章和博客，但除了 Douglas Crockford 的 JSON 官方网站，尚没有一处地方对这些知识和资源进行汇聚。

## 本书的目标读者

本书的目标读者是设计或实现 Web 和移动端应用程序、RESTful API 以及消息系统的架构师和开发人员。本书中的代码示例是用以下编程语言编写的：JavaScript、Node.js、Ruby on Rails 和 Java。如果你使用的是其他编程语言，如 Groovy、Go、Scala、Perl、Python、Clojure 或者 C#，同样需要阅读本书中的示例代码。不过你大可放心，绝大多数主流的现代编程语言都可以提供优秀的 JSON 支持。对于架构师，本书提供了指南、最佳实践以及架构和设计图表。然而，除了提供技术愿景，真正的架构师往往会用实际代码来佐证自己的观点。虽然我很喜欢编写代码来使用 JSON，但如果没有用例，缺少业务和技术背景，那么一切都将毫无意义。对于开发人员，本书汇聚了代码示例、工具、单元测试，等等。

为了保持简洁和专注，第 5~10 章仅提供在 Node.js 中编写的代码示例。但是，将这些示例转换为你使用的编程语言的代码并不难。

## “实战”的含义

2000 年年中，当我与 Scott Davis 合作编写 *JBoss at Work* 时，我们的愿景是编写一本开发人员能在日常工作中使用的书。同样，本书的目的也是为开发人员提供实用示例，这些示例是我根据实际的 JSON 使用经验所编写的。为此，我在每章后面添加了单元测试（如果这一章的内容可以编写单元测试的话）。原因很简单：如果一段代码没有对应的测试，则该段代码不存在。

准备好卷起袖子看代码吧。无论你是架构师还是开发人员，本书都会对你的工作有所帮助。

## 本书内容

通过阅读本书并练习书中的示例，你将学到以下实战操作：

- JSON 基础知识，以及如何对 JSON 数据进行建模；
- 在 Node.js、Ruby on Rails 以及 Java 中使用 JSON；
- 使用 JSON Schema 结构化 JSON 文档来设计并测试 API；
- 使用 JSON 搜索工具来搜索 JSON 文档的内容；
- 使用 JSON 转换工具将 JSON 文档转换成其他数据格式；
- 将 JSON 作为企业级架构中的一部分来使用；
- 比较 HAL、json:api 等 JSON 超媒体格式；

- 使用 MongoDB 来存储和访问 JSON 文档；
- 使用 Apache Kafka 在服务间交换 JSON 消息；
- 使用免费的 JSON 工具来简化测试；
- 通过简单的工具和类库，使用自己偏好的编程语言来调用 API。

## 本书使用的工具

以下是本书中用到的 JSON 工具：

- JSON 编辑器 / 建模工具；
- 单元测试工具（如 Mocha/Chai、Minitest、JUnit）；
- JSON 校验工具；
- JSON Schema 生成器；
- JSON 搜索工具；
- JSON 转换（模板）工具。

## 本书不适合哪些读者

如果对 JSON 的兴趣仅限于用 JavaScript 来发起 Ajax 调用，那么本书并不适合你。虽然本书也涉及了 Ajax 调用，但这只是所有内容中的冰山一角。有关 JavaScript 的很多图书中都包含有关 Ajax 调用的章节。

本书不会包含 REST、Ruby on Rails、Java 和 JavaScript 等内容的深入介绍。本书会用到上述技术，但将关注点放在了如何通过这些技术来使用 JSON 上。

## 本书的架构

本书由以下几部分内容组成：

- 第一部分，JSON 概述与平台；
- 第二部分，JSON 生态系统；
- 第三部分，JSON 的企业级应用；
- 附录。

### 第一部分，JSON概述与平台

- **第 1 章 JSON 概述**  
这一章从概述 JSON 数据格式开始，描述使用 JSON 过程中的最佳实践，并介绍本书中所使用的工具。
- **第 2 章 在 JavaScript 中使用 JSON**  
这一章展示了如何在 JavaScript、Node.js、Mocha/Chai 单元测试中使用 JSON。
- **第 3 章 在 Ruby on Rails 中使用 JSON**  
这一章描述了如何在 Ruby 对象和 JSON 之间进行转换，以及如何与 Rails 进行集成。

- 第 4 章 在 Java 中使用 JSON  
这一章讲述了如何在 Java 和 Spring Boot 中使用 JSON。

## 第二部分，JSON生态系统

- 第 5 章 JSON Schema  
这一章将帮助你用 JSON Schema 对 JSON 文档进行结构化操作。同时，你还会学习如何生成 JSON Schema 并用其来设计 API。
- 第 6 章 在 JSON 中进行搜索  
这一章展示了如何通过 jq 和 JSONPath 搜索 JSON 文档。
- 第 7 章 JSON 转换  
这一章提供了工具，从而将设计糟糕的 JSON 文档转换为更优雅、更有用的 JSON 文档。这一章还介绍了如何在 JSON 与 XML、HTML 等其他格式间进行相互转换。

## 第三部分，JSON的企业级应用

- 第 8 章 JSON 与超媒体  
这一章介绍了如何在 JSON 中使用多种知名的超媒体格式，如 HAL 和 jsonapi。
- 第 9 章 JSON 与 MongoDB  
这一章展示了如何使用 MongoDB 来存储与处理 JSON 文档。
- 第 10 章 用 Kafka 实现 JSON 消息系统  
这一章描述了如何使用 Apache Kafka 在服务间交换 JSON 消息。

## 附录

- 附录 A 介绍了如何安装运行本书示例所需的应用程序。
- 附录 B 提供了更多有关 JSON 社区（如标准、教程）的信息与链接，有助于你深入学习 JSON。

## 代码示例

本书中的所有代码示例及网址链接均可在图灵社区本书页面免费下载：<http://www.it-ebooks.com.cn/book/2093>。

本书是要帮你完成工作的。一般来说，如果本书提供了示例代码，你可以把它用在你的程序或文档中。除非你使用了很大一部分代码，否则无须联系我们获得许可。比如，用本书的几个代码片段写一个程序就无须获得许可，销售或分发 O'Reilly 图书的示例光盘则需要获得许可；引用本书中的示例代码回答问题无须获得许可，将书中大量的代码放到你的产品文档中则需要获得许可。

我们很希望但并不强制要求你在引用本书内容时加上引用说明。引用说明一般包括书名、作者、出版社和 ISBN。比如：“*JSON at Work* by Tom Marrs (O'Reilly). Copyright 2017 Vertical Slice, Inc., 978-1-449-35832-7.”

如果你觉得自己对示例代码的用法超出了上述许可的范围，欢迎你通过 [permissions@oreilly.com](mailto:permissions@oreilly.com) 与我们联系。

## O'Reilly Safari

Safari（前身为 Safari Books Online，<http://oreilly.com/safari>）是一个会员制的培训和参考平台，面向企业、政府、教育从业者和个人。



Safari 用户可访问 O'Reilly Media、Harvard Business Review、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 等 250 多家出版社的上千种图书、培训视频、学习路径、交互式教程和精选播放列表。

如需了解更多信息，请访问 <http://oreilly.com/safari>。

## 联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)  
奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：<http://shop.oreilly.com/product/0636920028482.do>。

对于本书的评论和技术性问题，请发送电子邮件到：

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：<http://www.youtube.com/oreillymedia>

# 致谢

首先，我要感谢创建并标准化 JSON 数据格式的 Douglas Crockford。JSON 是 REST 和微服务领域数据所用的语言，整个社区都应该感谢 Crockford 的愿景与付出。

感谢 O'Reilly 本书的编辑 Megan Foley 和前任编辑 Simon St. Laurent，感谢他们对本书的信任，以及在本书出版过程中给予我的耐心与指导。感谢他们一直支持我，并帮助我完成整个项目。还要感谢这本书的文字编辑和制作编辑 Nick Adams 和 Sharon Wilkey，感谢他们孜孜不倦地改进本书质量。

感谢 O'Reilly 开源大会的 Matthew McCullough 和 Rachel Roumeliotis，No Fluff Just Stuff (NFJS) 大会的 Jay Zimmerman，以及 Great Indian Developer Summit (GIDS) 大会的 Dilip Thomas，感谢他们让我在会议上分享有关 JSON 和 REST 的内容。在会议上进行分享非常有趣，希望未来还能有这样的机会。

感谢对本书提出宝贵反馈的技术审稿人：Joe McIntyre、David Bock、Greg Ostravich 和 Zettie Chinfong。此外，还要感谢以下人员，他们帮助我梳理了讲述 JSON 的思路：Matthew McCullough、Scott Davis、Cristian Vyhmeister、Senthil Kumar、Sean Pettersen、John Gray、Doug Clark、Will Daniels、Dan Carda 和 Peter Piper。

科罗拉多的技术社区是世界一流的，而我也有幸在以下用户组中做过分享，这对改进本书材料不无帮助：

- HTML5 Denver
- Denver Open Source User Group (DOSUG)
- Colorado Springs Open Source User Group (CS OSUG)
- Denver Java User Group (DJUG)
- Boulder Java User Group (BJUG)
- BoulderJS Meetup

感谢鼓励、信任和推动我出版本书的 Toastmasters 社区中的朋友：Darryle Brown、Deborah Frauenfelder、Eli-nora Reynolds、Betty Funderburke、Tom Hobbs、Marcy Brock，以及其他很多很多人。他们启发我要清晰地沟通、帮助同行、放开眼界。

互联网上的 JSON 社区非常棒。本书中的很多内容都是对社区优秀工作成果的阐述。技术社区启发我将 JSON 知识点串连起来，并讲述整个 JSON 技术图景。

感谢我已故的父母 Al MARR 和 Dorene MARR 一直以来对我的爱、信任和支持，我知道你们已经在一个更美好的地方了。你们启发我学会适应、创新并努力工作。你们总是鼓励我全力以赴。感谢你们为我做的一切。

最后，感谢我美丽的妻子 Linda 和女儿 Abby——我爱你们。当我将晚上和周末的时间用于写作和编程时，谢谢你们对我的耐心和理解。

# 电子书

扫描如下二维码，即可购买本书电子版。



# 目录

前言	xv
----	----

## 第一部分 JSON 概述与平台

第 1 章 JSON 概述	3
1.1 JSON 是一项技术标准	3
1.2 示例	4
1.3 为什么使用 JSON	5
1.4 JSON 的核心概念	6
1.4.1 JSON 数据类型	7
1.4.2 JSON 值类型	9
1.4.3 JSON 的版本	11
1.4.4 JSON 中的注释	12
1.4.5 JSON 文件及 MIME 类型	12
1.4.6 JSON 编码规范	12
1.5 本书示例: MyConference	14
1.5.1 本书技术栈	14
1.5.2 本书架构风格: noBackend	14
1.5.3 用 JSON Editor Online 对 JSON 数据进行建模	15
1.5.4 用 JSON Generator 生成示例数据	16
1.5.5 创建并部署模拟 API	16
1.6 本章回顾	19
1.7 内容预告	19

第 2 章 在 JavaScript 中使用 JSON	20
2.1 安装 Node.js	20
2.2 用 JSON.stringify() 和 JSON.parse() 进行序列化 / 反序列化操作	21
2.2.1 用于 stringify/parse 操作的“JSON”对象	21
2.2.2 JavaScript 中简单数据类型的 JSON 序列化操作	21
2.2.3 使用 toJSON() 进行对象的序列化操作	23
2.2.4 使用 eval() 进行 JSON 的反序列化操作	24
2.2.5 使用 JSON.parse() 进行 JSON 的反序列化操作	25
2.3 JavaScript 对象和 JSON	26
2.3.1 Node REPL	26
2.3.2 有关 JavaScript 对象的更多学习资料	28
2.4 用模拟 API 进行单元测试	28
2.4.1 单元测试风格——TDD 和 BDD	28
2.4.2 使用 Mocha 和 Chai 即可完成单元测试	29
2.4.3 设置单元测试环境	29
2.4.4 Unirest	29
2.4.5 测试数据	30
2.4.6 对演讲者数据进行单元测试	30
2.5 搭建小型 Web 应用程序	31
2.5.1 Yeoman	32
2.5.2 第 1 阶段：使用 Yeoman 生成 Web 应用程序	33
2.5.3 第 2 阶段：使用 jQuery 发起 HTTP 请求	36
2.5.4 第 3 阶段：在模板中使用模拟 API 所提供的演讲者数据	40
2.6 如何继续深入学习 JavaScript	44
2.7 本章回顾	45
2.8 内容预告	45
第 3 章 在 Ruby on Rails 中使用 JSON	46
3.1 安装 Ruby on Rails	46
3.2 Ruby 中与 JSON 有关的 gem 包	46
3.3 用 MultiJson 进行序列化 / 反序列化操作	47
3.3.1 MultiJson 对象	47
3.3.2 Ruby 中简单数据类型的 JSON 序列化 / 反序列化操作	48
3.3.3 用 MultiJson 进行 JSON 反序列化操作	50
3.3.4 关于 JSON 和驼峰式命名	52
3.3.5 用 ActiveSupport 进行 JSON 序列化操作	52
3.3.6 用 ActiveSupport 进行 JSON 反序列化操作	53



3.4	用模拟 API 进行单元测试 .....	54
3.4.1	使用 Minitest 即可完成单元测试 .....	54
3.4.2	设置单元测试环境 .....	55
3.4.3	测试数据 .....	55
3.4.4	用 Minitest 测试 API 所提供的 JSON .....	55
3.4.5	对演讲者数据的单元测试 .....	55
3.4.6	有关 Ruby 和 Minitest 的更多学习资料 .....	59
3.4.7	似乎少了点什么 .....	59
3.5	用 Ruby on Rails 搭建小型 Web API .....	59
3.5.1	选择 JSON 序列化工具 .....	60
3.5.2	speakers-api-1——创建 API 以提供驼峰式命名风格的 JSON .....	61
3.5.3	speakers-api-2——创建 API 以提供自定义风格的 JSON .....	67
3.5.4	有关 Rails 和 Rails API 的更多学习资料 .....	68
3.6	本章回顾 .....	69
3.7	内容预告 .....	69
<b>第 4 章</b>	<b>在 Java 中使用 JSON .....</b>	<b>70</b>
4.1	安装 Java 和 Gradle .....	70
4.2	Gradle 概览 .....	70
4.3	使用 JUnit 即可完成单元测试 .....	72
4.4	Java 中的 JSON 类库 .....	72
4.5	用 Jackson 进行 JSON 序列化 / 反序列化操作 .....	73
4.5.1	对 Java 中的简单数据类型进行序列化 / 反序列化操作 .....	73
4.5.2	对 Java 对象进行序列化 / 反序列化操作 .....	75
4.6	用模拟 API 进行单元测试 .....	79
4.6.1	测试数据 .....	79
4.6.2	用 JUnit 对 API 提供的 JSON 进行测试 .....	80
4.7	用 Spring Boot 搭建小型 Web API .....	83
4.7.1	创建模型 .....	84
4.7.2	创建控制器 .....	85
4.7.3	注册应用程序 .....	87
4.7.4	编写构建脚本 .....	87
4.7.5	部署 API .....	89
4.7.6	用 Postman 测试 API .....	89
4.8	本章回顾 .....	90
4.9	内容预告 .....	90