

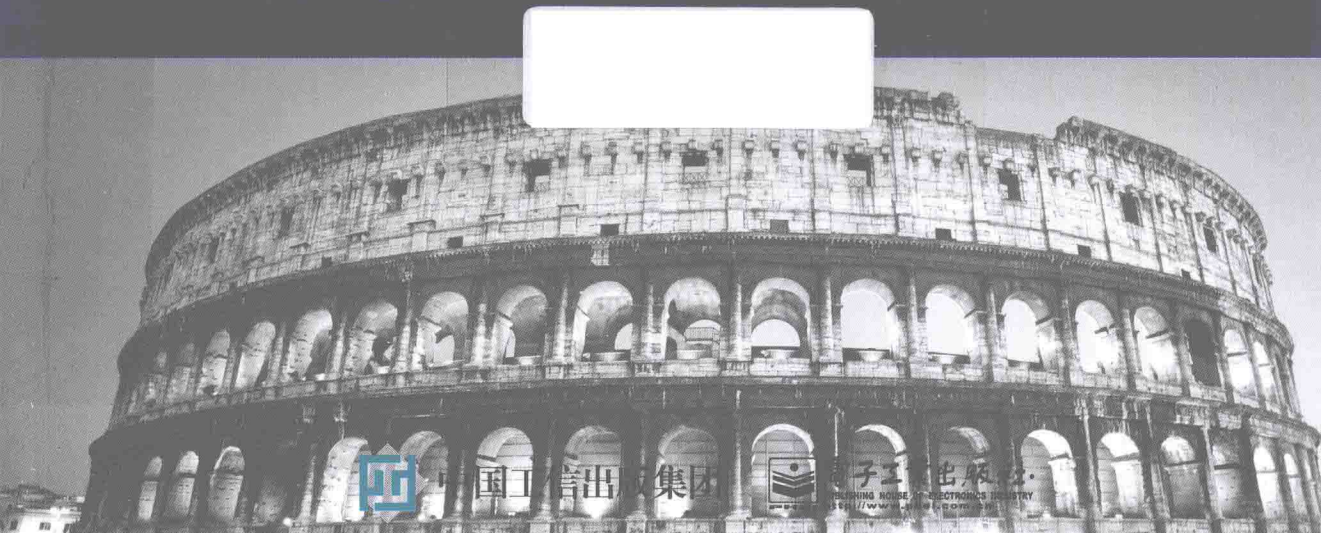
[美] Brad Dayley 著
卢涛 李颖 译

PEARSON

Broadview[®]
www.broadview.com.cn

Node.js+MongoDB +AngularJS Web开发

Node.js, MongoDB and AngularJS Web Development



Node.js, MongoDB and AngularJS
Web Development

Node.js+MongoDB ***+AngularJS Web开发***

[美] Brad Dayley 著
卢涛 李颖 译

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Node.js 是一种领先的服务器端编程环境, MongoDB 是最流行的 NoSQL 数据库, 而 AngularJS 正迅速成为基于 MVC 的前端开发的领先框架。它们结合在一起使得能够完全用 JavaScript 创建从服务器到客户端浏览器的高性能站点和应用程序。

本书为想要将这 3 种技术整合到全面的有效解决方案的 Web 程序员提供了完整指南。它简洁而清晰地介绍了这 3 种技术, 然后迅速转到构建几种常见的 Web 应用程序上面。

读者将学会使用 Node.js 和 MongoDB 来建立更具可扩展性的高性能网站, 并利用 AngularJS 创新的 MVC 方法构建更有效的网页和应用程序, 以及把这三者结合在一起使用, 从而提供卓越的下一代 Web 解决方案。

本书适合对 HTML 的基础知识已经有所了解, 并可以用现代编程语言完成一些编程的读者。读者如果对 JavaScript 有一定了解, 则将更易于理解本书的内容。

Authorized translation from the English language edition, entitled Node.js, MongoDB and AngularJS Web Development, 9780321995780 by Brad Dayley, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2014 Pearson Education, Inc.

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 retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2015

本书简体中文版专有出版权由 Pearson Education 培生教育出版亚洲有限公司授予电子工业出版社。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标签, 无标签者不得销售。

版权贸易合同登记号 图字: 01-2014-8175

图书在版编目(CIP)数据

Node.js+MongoDB+AngularJS Web 开发/(美)德雷(Dayley,B.)著;卢涛,李颖译. —北京:电子工业出版社, 2015.6

书名原文: Node.js, MongoDB and AngularJS Web Development

ISBN 978-7-121-26117-6

I. ①N… II. ①德… ②卢… ③李… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 109641 号

策划编辑:张春雨

责任编辑:李云静

印 刷:北京丰源印刷厂

装 订:三河市皇庄路通装订厂

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:37 字数:780 千字

版 次:2015 年 6 月第 1 版

印 次:2015 年 6 月第 1 次印刷

定 价:108.00 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。

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

服务热线: (010) 88258888。

译者序

随着互联网的发展，曾几何时，“全栈工程师”（Full Stack Engineer）的概念开始兴起，这种职位要求应征者对开发堆栈的每个方面都有所掌握，包括服务器、网络及宿主环境、数据建模、业务逻辑、API/Action/MVC、用户界面等。过去，这意味着全栈工程师需要面对归属不同层次的多种软件环境和语言，并处理各层之间的交互，这些技术中的任何一种都要耗费大量的精力和时间来学习，因而会面临很大的挑战。现在，这种情况已经改变，JavaScript 语言既可用于客户端开发，又能用于编写服务器端应用程序，还能方便地与传统及新型数据库交互，其中有些代码还能复用，从而为开发功能丰富的互联网应用程序创造了良好的条件。

Node.js 是一种领先的服务器端编程环境，它和 Express 的结合能够实现高度可扩展的动态 Web 服务器，并可用 JavaScript 编写 Web 服务。

MongoDB 是目前最流行的 NoSQL 数据库，可用于 Web 应用程序数据的存储，并能从 Node.js JavaScript 代码访问。

AngularJS 正迅速成为基于 MVC 的前端开发的领先框架，它的自定义指令扩展了 HTML 语言。

这三者整合在一起使得能够完全用 JavaScript 创建从服务器到客户端浏览器的高性能站点和应用程序。

本书提供了将这 3 种技术整合成全面的有效解决方案的完整指南。它简要而清晰地讲述了这 3 种技术，然后迅速转到构建几种常见的 Web 应用程序上面。最后还用多个实际的例子实现了可以与 Node.js Web 服务器交互的客户端服务，为用户提供了丰富的交互功能的动态浏览器视图，以及为 Web 网页添加用户身份验证和嵌套评论等组件。

作者 Brad Dayley 是一名高级软件工程师，精通 jQuery、JavaScript、MongoDB，著有多本技术书籍，并在企业应用程序及 Web 界面方面具有丰富的开发经验，本书正是他对实际工作成果的总结。

通过学习本书，读者将学会如何使用 Node.js 和 MongoDB 来建立更具可扩展性的高

IV 译者序

性能网站，如何利用 AngularJS 创新的 MVC 方法构建更有效的网页和应用程序，以及如何把这三者结合在一起使用，从而提供卓越的下一代 Web 解决方案。

李绿霞、卢林、陈克非、李洪秋、张慧珍、李又及、卢晓瑶、陈克翠、汤有四、李阳、刘雯、贾书民、苏旭晖也参与了部分翻译工作，感谢他们在本书翻译工作中的辛勤付出。

感谢我们的儿子卢〇一小朋友，他知道我们在翻译书稿就常常自己安静地读书和玩耍，还放弃了很多出去玩的机会，让我们能够专注于本书的翻译，本书的出版也有他的一份贡献。

最后，希望这本书能对读者有所助益。但由于译者经验和水平有限，译文中难免有不妥之处，恳请读者批评指正！

卢涛 李颖

2015 年 5 月 9 日

致 谢

谨以此篇向所有为本书做出贡献的人致谢。首先，我要感谢我的贤妻，感谢她给予我灵感、爱和支持。没有你，我不会写就此书。我还要感谢我的儿子们，感谢在写作时间里你们给予我的支持，以及你们带给我的欢乐时光。

感谢 Mark Taber 确保此书顺利出版；感谢 Russell Kloepper 和 Siddhartha Singh，他们的技术审查使我能够保持诚实；Kitty Wilson 将“源于我的头脑中的技术”随笔转换成优美的文字；Tammy Graham 和 Laura Robbins 则负责本书所有图形的设计；本书这个超炫的封面是由 Chuti Prasertsith 设计的；而 Elaine Wiley 负责这个项目，并确保这本书是最优质的。

作者简介

Brad Dayley 是一名高级软件工程师，在开发企业应用程序及 Web 界面方面，他拥有超过 20 年的工作经验。他熟练应用 JavaScript 和 jQuery 多年，并是 *jQuery and JavaScript Phrasebook* 和 *Sams Teach Yourself jQuery and JavaScript in 24 Hours* 的作者。他曾设计并实现了一大批应用程序和服务——从应用服务器到复杂的 Web 2.0 界面。同时，他还著有 *Python Developer's Phrasebook* 和 *Sams Teach Yourself Django in 24 Hours*。

前言

欢迎阅读本书。本书将引领你进入使用 JavaScript 的世界——在你的 Web 开发项目中，从服务器和服务到浏览器客户端。本书涵盖 Node.js、MongoDB 和 AngularJS 的实现和集成，而它们是 Web 开发世界中新兴的一些最令人兴奋和创新的技术。

本篇前言包括：

- 本书受众；
- 为什么要阅读本书；
- 从本书你将了解到的知识；
- Node.js、MongoDB 和 AngularJS 分别是什么，以及为什么它们都是出色的技术；
- 本书的组织结构；
- 在哪里可以找到代码示例。

让我们开始吧。

本书受众

这本书假定读者已经对 HTML 的基础知识有所了解，并可以用现代编程语言完成一些编程。读者如果对 JavaScript 有一定了解，将更容易理解本书的内容；但这不是必需的，因为本书确实也涵盖了 JavaScript 的基础知识。

为什么要阅读本书

这本书将教你如何创建功能强大的互动网站和 Web 应用程序——从 Web 服务器和服务端上的服务到基于浏览器的交互式 Web 应用程序。这里所涉及的技术都是开源的，在服务器端组件和浏览器端的组件上你都可以使用 JavaScript。

这本书的大多数读者想要掌握 Node.js 和 MongoDB，以便可以达到构建高度可扩展和高性能网站的目的。大多数读者也想利用 AngularJS 创新的 MVC 方法来实现精心设计和结构化的网页和 Web 应用程序。总之，Node.js、MongoDB 和 AngularJS 提供了一个易于实现并完全集成的 Web 开发套件，它可以让你实现神奇的 Web 2.0 应用程序。

从本书你将了解到的知识

阅读本书将帮助你构建现实中的动态网站和 Web 应用程序。网站不再由 HTML 页面和集成的图像及格式化的文本等简单的静态内容构成。相反，网站变得更加动态，单个网页往往充当一个完整的网站或应用程序。

使用 AngularJS 技术，可让你在网页中构建逻辑，这可以与 Node.js 服务器相互通信并从 MongoDB 数据库获取必要的数据库。Node.js、MongoDB 和 AngularJS 的组合可以让你实现交互式动态网页。通过阅读本书，你将学会如下的事情：

- 如何使用 Node.js 和 Express 来实现一个高度可扩展的动态 Web 服务器；
- 如何在 JavaScript 中创建服务器端的 Web 服务；
- 如何在 Web 应用程序中实现 MongoDB 的数据存储；
- 如何用 Node.js JavaScript 代码实现对 MongoDB 的访问和交互；
- 如何定义静态和动态 Web 路由并实现服务器端脚本来支持它们；
- 如何定义扩展 HTML 语言的你自己的自定义的 AngularJS 指令；
- 如何实现可以与 Node.js Web 服务器交互的客户端服务；
- 如何建立提供丰富的用户交互的动态浏览器视图；
- 如何将用户账户的身份验证添加到网站/Web 应用程序；
- 如何将嵌套的评论组件添加到网页；
- 如何建立一个端到端的购物车。

何为 Node.js

Node.js 是基于谷歌的 V8 JavaScript 引擎的开发框架。你可以用 JavaScript 编写 Node.js 代码，然后 V8 将它编译为要执行的机器代码。你可以用 Node.js 编写出大部分，或者甚

至全部的服务器端代码，包括 Web 服务器、服务器端的脚本和任何支持 Web 应用程序的功能。Web 服务器和支持 Web 应用程序的脚本在同一个服务器端应用程序中运行这一事实，允许在 Web 服务器和脚本之间有更紧密的集成。

Node.js 之所以是一个出色的框架，基于下面几个原因。

- **JavaScript 端到端：**Node.js 最大的一个优点是，它可以让你用 JavaScript 同时编写服务器端和客户端脚本。在决定是把逻辑放入客户端脚本还是服务器端脚本方面一直有困难。利用 Node.js，你可以在客户端上编写 JavaScript，并轻松地在服务器上适应它，反之亦然。另外一个好处是，客户端的开发者和服务器的开发者使用同一种语言。
- **事件驱动的可扩展性：**Node.js 应用独特的逻辑来处理 Web 请求。使用 Node.js，不是让多个线程等待处理 Web 请求，而是采用一种基本的事件模型在同一个线程上处理它们。这使得 Node.js Web 服务器可以用传统的 Web 服务器不能做到的方式进行扩展。
- **可扩展性：**Node.js 有很多的追随者和非常活跃的开发社区。人们正在不断提供新的模块来扩展 Node.js 的功能。此外，在 Node.js 中安装和包含新的模块是非常简单的，你可以在几分钟内扩展 Node.js 的项目来包含新的功能。
- **快速执行：**建立 Node.js，并在其中开发是超级容易的。在短短几分钟内就可以安装 Node.js，并拥有一个能工作的 Web 服务器。

何为 MongoDB

MongoDB 是一个灵活的和可扩展性非常好的 NoSQL 数据库。Mongo 这个名字来自单词“堆积如山”（humongous），用来强调 MongoDB 提供的可扩展性和性能。MongoDB 为需要存储诸如用户评论、博客或其他条目数据的高流量的网站提供了出色的网站后端存储，因为它可快速扩展并易于实现。

下面是 MongoDB 真正适合于 Node.js 套件的一些原因。

- **针对文档：**因为 MongoDB 是针对文档的，所以数据在数据库中存储的格式，非常接近你在服务器端和客户端脚本中处理它们的格式。这消除了把数据从行转换为对象和转换回来的需要。
- **高性能：**MongoDB 是目前性能最高的数据库之一。尤其是在现在，有越来越多的人与网站进行交互，具有能够支持大流量的后端是很重要的。

- **高可用性:** MongoDB 的复制模型使得它很容易维护可扩展性,同时保持高性能。
- **高可扩展性:** MongoDB 的结构使得它可以很容易地通过在多个服务器上对数据分片实现横向扩展。
- **无 SQL 注入:** MongoDB 是不容易受到 SQL 注入攻击的(也就是向 Web 表单或从浏览器的其他输入中输入 SQL 语句,从而危及数据库的安全性)。这是因为对象被存储为对象,不使用 SQL 字符串。

何为 AngularJS

AngularJS 是由谷歌开发的客户端框架。它是用 JavaScript 编写的,带有一个精简过的 jQuery 库。AngularJS 背后的理论是提供一个框架,使得可以很容易地使用 MVC 框架实现设计良好的结构化网页和应用程序。

AngularJS 提供了在浏览器中处理用户输入、操纵客户端上的数据及控制元素如何在浏览器界面上显示的功能。下面是 AngularJS 具有的一些优势。

- **数据绑定:** AngularJS 利用其强大的范围机制,有一个将数据绑定到 HTML 元素的非常干净的方法。
- **可扩展性:** AngularJS 架构允许你轻松地扩展语言的各个方面,以提供你自己的自定义实现。
- **整洁:** AngularJS 迫使你编写整洁的、合乎逻辑的代码。
- **可重用代码:** 可扩展性和简洁代码的结合,使得很容易用 AngularJS 编写可重用的代码。事实上,在创建自定义服务的时候,此语言往往迫使你这样做。
- **支持:** 谷歌正把大量资金投入到这个项目,这使得它比那些已失败的类似举措更具优势。
- **兼容性:** AngularJS 基于 JavaScript 并与 jQuery 有着密切的联系。这使得更易于开始将 AngularJS 整合到你的环境和重用在 AngularJS 框架结构内的现有代码片段。

本书的组织结构

这本书分为 6 个主要部分:

- 第 1 部分“引言”,概述了 Node.js、MongoDB 和 AngularJS 之间的相互作用,并

对这 3 种产品如何形成一个完整的 Web 开发套件进行了概述。第 2 章是 JavaScript 的初步介绍，它提供了实现 Node.js 和 AngularJS 代码时，你需要用到的 JavaScript 语言的基本知识。

- 第 2 部分“学习 Node.js”，涵盖了 Node.js 的语言平台，从安装到实现 Node.js 模块。这部分向你提供你需要的基本框架，以实现自己的自定义 Node.js 模块以及 Web 服务器和服务器端脚本。
- 第 3 部分“学习 MongoDB”，涵盖了 MongoDB 数据库，从安装到与 Node.js 应用程序的集成。本部分将讨论如何规划你的数据模型，以满足你的应用程序需求，以及如何从 Node.js 应用程序对 MongoDB 进行访问和交互。
- 第 4 部分“使用 Express 使生活更轻松”，讨论了 Node.js 的 Express 模块以及如何利用它作为应用程序的 Web 服务器。你将学习如何为数据设置动态和静态路由，以及如何实现安全性、缓存和 Web 服务器的其他基本功能。
- 第 5 部分“学习 AngularJS”，涵盖了 AngularJS 框架的架构，以及如何将它集成到 Node.js 套件。本部分介绍了创建自定义的 HTML 指令和在浏览器中利用的客户端服务。
- 第 6 部分“建立实用的 Web 应用程序组件”，转换了主题，给出了一些使用 Node.js、MongoDB 和 AngularJS 的端到端应用案例；这些案例为 Web 应用程序提供用户账户的身份验证、评论部分，以及购物车。这部分还介绍了一些在 Web 应用程序中实现 Web 2.0 互动的方法。

获取代码示例

在本书中你会看到很多以代码清单形式提供的代码示例。每个清单的标题都包括源代码的文件名。要访问示例中使用的源代码文件和图片，请访问如下网址：

<https://github.com/bwdbooks/nodejs-mongodb-angularjs-web-development>

结束语

我希望你和我一样喜欢学习 Node.js、MongoDB 和 AngularJS。它们都是出色而创新的技术，使用起来真的充满乐趣。很快，你就可以加入这一庞大的 Web 开发人员之列，和他们一起使用 Node.js-to-AngularJS Web 套件来建立交互式网站和 Web 应用程序了。我也希望你喜欢这本书！

目 录

第 1 部分 引 言

第 1 章 介绍 Node.js-to-AngularJS 套件	3
1.1 了解基本的 Web 开发框架	3
1.1.1 用户	4
1.1.2 浏览器	4
1.1.3 Web 服务器	6
1.1.4 后端服务	6
1.2 了解 Node.js-to-AngularJS 套件组件	7
1.2.1 Node.js	7
1.2.2 MongoDB	8
1.2.3 Express	9
1.2.4 AngularJS	9
1.3 小结	10
1.4 下一章	10
第 2 章 JavaScript 基础	11
2.1 定义变量	11
2.2 了解 JavaScript 数据类型	12
2.3 使用运算符	13
2.3.1 算术运算符	13
2.3.2 赋值运算符	14
2.3.3 运用比较和条件运算符	14
2.4 实现循环	16
2.4.1 while 循环	17
2.4.2 do/while 循环	17
2.4.3 for 循环	17

2.4.4	for/in 循环	18
2.4.5	中断循环	19
2.5	创建函数	19
2.5.1	定义函数	20
2.5.2	传递变量给函数	20
2.5.3	从函数返回值	20
2.5.4	使用匿名函数	21
2.6	理解变量作用域	22
2.7	使用 JavaScript 对象	22
2.7.1	使用对象语法	23
2.7.2	创建自定义对象	23
2.7.3	使用原型对象模式	24
2.8	处理字符串	25
2.8.1	合并字符串	26
2.8.2	在字符串中搜索子串	26
2.8.3	在一个字符串中替换单词	27
2.8.4	将字符串分割成数组	27
2.9	使用数组	27
2.9.1	合并数组	28
2.9.2	遍历数组	29
2.9.3	将数组转换为字符串	29
2.9.4	检查数组是否包含某个条目	29
2.9.5	在数组中添加条目和删除条目	30
2.10	添加错误处理	30
2.10.1	try/catch 块	30
2.10.2	抛出你自己的错误	31
2.10.3	使用 finally	31
2.11	小结	32
2.12	下一章	32

第 2 部分 学习 Node.js

第 3 章	开始使用 Node.js	35
3.1	了解 Node.js	35
3.1.1	谁在使用 Node.js	35

3.1.2	Node.js 的用途	36
3.2	Node.js 安装	36
3.2.1	纵观 Node.js 安装位置	36
3.2.2	验证 Node.js 可执行文件	37
3.2.3	选择 Node.js IDE	37
3.3	使用 Node.js 包	38
3.3.1	什么是 Node 封装模块	38
3.3.2	了解 Node 包注册表	38
3.3.3	使用 Node 包管理器	38
3.3.4	搜索 Node 封装模块	39
3.3.5	安装 Node 封装模块	40
3.3.6	使用 package.json	41
3.4	创建 Node.js 应用程序	43
3.4.1	创建 Node.js 模块封装	43
3.4.2	将一个 Node.js 封装模块发布到 NPM 注册表	45
3.4.3	在 Node.js 应用程序中使用 Node.js 封装模块	46
3.5	将数据写入控制台	47
3.6	小结	49
3.7	下一章	49
第 4 章	在 Node.js 中使用事件、监听器、定时器和回调	51
4.1	了解 Node.js 事件模型	51
4.1.1	比较事件回调和线程模型	51
4.1.2	在 Node.js 中阻塞 I/O	52
4.1.3	会话示例	54
4.2	将工作添加到事件队列	54
4.2.1	实现定时器	55
4.2.2	使用 nextTick 来调度工作	58
4.2.3	实现事件发射器和监听器	59
4.3	实现回调	62
4.3.1	向回调函数传递额外的参数	63
4.3.2	在回调中实现闭包	64
4.3.3	链式回调	65
4.4	小结	66
4.5	下一章	66

第 5 章 在 Node.js 中处理数据 I/O	67
5.1 处理 JSON	67
5.1.1 把 JSON 转换成 JavaScript 对象	67
5.1.2 把 JavaScript 对象转换为 JSON	68
5.2 使用 Buffer 模块缓冲数据	68
5.2.1 了解缓冲数据	69
5.2.2 创建缓冲区	69
5.2.3 写入缓冲区	70
5.2.4 从缓冲区读取	71
5.2.5 确定缓冲区长度	72
5.2.6 复制缓冲区	73
5.2.7 对缓冲区切片	74
5.2.8 拼接缓冲区	75
5.3 使用 Stream 模块来传送数据	76
5.3.1 Readable 流	76
5.3.2 Writable 流	78
5.3.3 Duplex 流	81
5.3.4 Transform 流	82
5.3.5 把 Readable 流用管道输送到 Writable 流	84
5.4 用 zlib 压缩与解压缩数据	85
5.4.1 压缩和解压缩缓冲区	85
5.4.2 压缩/解压缩流	87
5.5 小结	88
5.6 下一章	88
第 6 章 从 Node.js 访问文件系统	89
6.1 同步和异步文件系统调用	89
6.2 打开和关闭文件	90
6.3 写入文件	91
6.3.1 简单文件写入	91
6.3.2 同步文件写入	92
6.3.3 异步写入文件	93
6.3.4 流式文件写入	95
6.4 读取文件	96
6.4.1 简单文件读取	96

6.4.2	同步文件读取	97
6.4.3	异步文件读取	98
6.4.4	流式文件读取	100
6.5	其他文件系统任务	101
6.5.1	验证路径的存在性	101
6.5.2	获取文件信息	102
6.5.3	列出文件	103
6.5.4	删除文件	104
6.5.5	截断文件	105
6.5.6	建立和删除目录	105
6.5.7	重命名文件和目录	106
6.5.8	监视文件更改入	107
6.6	小结	107
6.7	下一章	107
第 7 章	在 Node.js 中实现 HTTP 服务	109
7.1	处理 URL	109
7.1.1	了解 URL 对象	110
7.1.2	解析 URL 组件	111
7.2	处理查询字符串和表单参数	111
7.3	了解请求、响应和服务器对象	112
7.3.1	http.ClientRequest 对象	112
7.3.2	http.ServerResponse 对象	114
7.3.3	http.IncomingMessage 对象	115
7.3.4	HTTP Server 对象	116
7.4	在 Node.js 中实现 HTTP 客户端和服务	118
7.4.1	提供静态文件服务	118
7.4.2	实现动态的 GET 服务器	120
7.4.3	实现 POST 服务器	122
7.4.4	与外部源交互	124
7.5	实现 HTTPS 服务器和客户端	127
7.5.1	创建 HTTPS 客户端	127
7.5.2	创建 HTTPS 服务器	129
7.6	小结	129
7.7	下一章	129