

Building Node Applications with MongoDB and Backbone



Node 应用程序构建

——使用*MongoDB*和*Backbone*

O'REILLY®

[美] *Mike Wilson* 著
林冀 范俊 张鹏 译

人民邮电出版社
POSTS & TELECOM PRESS

014037079

TP312JA
1575

O'REILLY®

Node 应用程序构建 ——使用 MongoDB 和 Backbone



[美] Mike Wilson 著
林冀 范俊 张鹏 译

TP312JA
1575



北航

C1725123

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Node应用程序构建 : 使用MongoDB和Backbone /
(美) 威尔逊 (Wilson, M.) 著 ; 林冀, 范俊, 张鹏译
— 北京 : 人民邮电出版社, 2014. 4
ISBN 978-7-115-34629-2

I. ①N… II. ①威… ②林… ③范… ④张… III. ①
JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第034759号

版权声明

Copyright© 2013 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2014. Authorized translation of the English edition, 2013 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.

本书中文简体版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

-
- ◆ 著 [美] Mike Wilson
译 林 冀 范 俊 张 鹏
责任编辑 陈冀康
责任印制 程彦红 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>
三河市海波印务有限公司印刷
- ◆ 开本: 787×1000 1/16
印张: 11.5
字数: 212 千字 2014 年 4 月第 1 版
印数: 1—3 000 册 2014 年 4 月河北第 1 次印刷

著作权合同登记号 图字: 01-2013-1024 号

定价: 39.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

作者简介

Mike Wilson 曾有幸在世界上一些最大和最有影响力的品牌工作，其中包括迪斯尼、微软和麦当劳。他具有多年的 web 开发经验，从小型企业网站到能够容纳百万玩家的大型 MMO 服务器集群都能够设计和构建。在他空闲的时间里，Mike 维护着他的个人博客 (<http://www.alwaysgetbetter.com/>) 并参与到一些论坛中去，对新兴的软件和框架进行试验。Mike 与他的妻子和三个孩子住在温哥华。

封面简介

本书的封面上的动物是小灵猫 (*Viverricula indic*)，在南亚和东南亚以及印度尼西亚群岛都有发现。这种动物是以自卫过程中会产出厚黄麝香气味的物质命名的。

小灵猫身材修长，能够敏捷爬树，没有竖起的鬃毛，生活在洞里岩石上和毛茸茸的位置。在夜间，独来独往，通常它会栖息在树上；也就是说，它会爬树，并待在那里。然而，当狩猎时，它会在地面水平位置。灵猫基本上是杂食的包括蜥蜴，鼠类，鸟类，昆虫，甚至是鸡蛋。在人工饲养下，它很容易驯服，给它喂小动物，它会像猫一样灵巧。

在小灵猫身上有一排排黑色的斑点和尾巴上的条纹搭配上灰色或黄褐色。它的腿、耳朵和嘴巴也是黑色的。

灵猫（也称果子狸）产生的麝香是非常珍贵的香味和香料的稳定剂。雄性和雌性狸猫身体上的腺体都会分泌强烈的气味。人们为获得狸猫的肉而捕猎它，将它的皮肤净化后入药。



北航

C1725123

内 容 提 要

Node.js 是一套用来编写高性能网络服务器的 JavaScript 工具包。Backbone.js 是面向客户端的 JavaScript 框架。MongoDB 是一种 NoSQL 的数据库。三者结合使用，可以构建出高效的 Web 应用。

本书分为两部分，共 10 章。第一部分包括第 1 到 4 章，概述了 Node.js、MongoDB 和 Backbone.js 的核心技术。第二部分包括第 5 章到第 10 章，介绍如何使用这些工具去构建一个具有社交网络风格的网站。如果是刚开始学习，建议先从第一部分获取一些背景知识，然后在第二部分深入学习。如果你已经熟悉 JavaScript，可以直接跳过第一部分，会发现第二部分中的例子也是可以理解的。

本书可以作为学习和掌握 Node.js、Backbone.js 和 MongoDB 的实践教程，也适合对这几种技术感兴趣的读者阅读参考。

前言

Google 在 2008 年发布了第一个版本的 V8 JavaScript 引擎时，兴奋就像安静波澜，在开发者社区荡漾开来。第一次（承诺），我们能够在客户端和服务端都使用 JavaScript 进行编程：使用一种语言来控制一切。Web 应用程序已经开始变得更像桌面并且复杂程度也在膨胀，所以减少语言数量的依赖这个想法有利于使技术开放和透明，推出更加令人兴奋和不断突破的应用程序。

Ryan Dahl 就是一个看到了新的机会的开发者，他没有浪费时间，马上将他原来写的非阻塞套接字库转换到新的 V8 引擎上，导致了 Node.js 的诞生。他发布的技术已经让原来荡漾着的兴奋变成一次重大范式变革，对于实时响应的应用程序的兴趣到达了高峰。Node.js 不仅仅是一组套接字函数的集合，它提供了一个框架处理异步 I/O，以全新的事件驱动编程模式作为基础。

网上的情况在过去的几年中已经快速发生改变，没有任何迹象表明会放缓。“社交”网络的爆发对我们来说意义重大：越来越多的人在线，已经不只是技术用户。互联网是所有人的，在这个新的空间里，赢家将是那些明白如何让在线体验温暖而人性化，真正让人与人之间联系起来的公司。

使用 JavaScript 来连接系统提供了一个优势，因为你可以快速从处理与个人用户交互的前端 Web 堆栈转换到处理后端的数据存储，以及之间的所有网络管道。你将能够把系统真正模块化；每个部分都是可插拔的，可以部署到资源最合适它的地方。不同于以前，你创建的应用程序将可以与用户群共同成长、共同呼吸。

读者和假设

本书的读者应该了解网站和网络应用是如何组织在一起的。为了保持对核心技术的专注，本书略过“为什么”构建 Web 应用程序，而关注“如何”构建。

要完全理解书中的示例，需要一些 JavaScript 的知识。这些示例会进行透彻的解析，但是以前的知识能够帮助读者理解编写过程中做出的编程决策和背后的故事。

许多开发者使用 NoSQL 数据库是从关系型数据库系统转过来的。这本书并没有假

设读者熟练掌握数据库设计，我详细讨论了在整个数据库架构阶段做出的设计决策的缘由。MongoDB 对于 SQL 的概念是友好的，这是选择它作为这个项目数据存储的主要动机。

在本书的最后一部分，我将走出第一和第二部分中纯 JavaScript 环境的构建，讨论如何选择支持的工具和技术。不要求读者对其他语言（比如 Scala、Java、PHP 和 Bash 脚本编程）有深刻的理解，因为深入探索这些概念超出了本书的范畴，所以我鼓励使用这些示例作为跳板，进行进一步研究。

组织结构

这本书基本分为两部分，第一部分概述 Node.js、Backbone.js 和 MongoDB（本书讲述的核心技术），第二部分是如何使用这些工具去构建一个具有社交网络风格的网站。如果是刚开始学习，建议先从第一部分获取一些背景知识，然后在第二部分深入学习。如果你已经熟悉 JavaScript，可以直接跳过第一部分，会发现第二部分中的例子也是可以理解的。

下面是本书的组织结构。

第一部分 Node.js、Backbone.js 和 MongoDB 简介

第 1 章 介绍和总览

本章介绍了 JavaScript 和贯穿全书探讨的一些核心概念。

第 2 章 Node.js

本章介绍了 Node.js，并引导你开始第一个独立的应用程序。当你熟悉关键模块后，你会用它来建立一个完整的实时应用。

第 3 章 Backbone.js

接下来，您将探索 Backbone.js 如何使得 JavaScript 在浏览器上的编程更像是构建传统应用程序而不是网站。我们也会遇到维护基于 JavaScript 项目中的一些问题，引入模板将可视的 HTML 布局与功能性的 JavaScript 应用代码分离开。

第 4 章 MongoDB

喜欢 MongoDB 是因为它可以快速和简单的搭建，易于接口，并与 Node.js 应用语言相通。在这一章我们将着眼于如何做基本的查询和数据处理，以及当 MongoDB 使用增长时复杂随之增加，这些情况也是需要考虑的。

第二部分 · 建立社交网络

第 5 章 创建项目

如何构建项目并将项目中的文件组织在一起是面临的一个大问题。在这一章中配置 Node.js 和 Backbone.js 形成这个网站，为其余的代码奠定基础。

第 6 章 认证

在使用应用程序之前，你需要创建账号并登录。在这一章中解释了如何从数据库中获取用户，并且一旦拥有他们的数据，如何确保数据的安全。

第 7 章 用户界面

在基本结构和登录功能完成之后，这一章引导你建立网页，包含所有展现给用户的内容。这里我们将详细介绍模板和 Backbone.js 的使用，集成访问控制和 Node.js。

第 8 章 交朋友

联系人列表就是这个网站背后的社交部分。在这一章中，你将学习到如何在列表中添加和删除联系人，将数据去规范化，存放到 MongoDB 中。对于一个来自关系型数据库环境的来说，这一章是很值得阅读的。

第 9 章 聊天

这一章在第 8 章创建的联系人列表的基础上，利用 Socket.io 添加了实时聊天的功能。你和朋友之间收发消息不需要重新加载页面。

第 10 章 实时互动

最后，我们将回顾书中构建的用户界面，利用 Socket.io 进行扩展，就像聊天列表一样。这将为网站增加活力，让用户及时知道联系人的上线和下线，将所有共享的消息空间变为可以交互的聊天室。

目录

第一部分 Node.js、Backbone.js 和 MongoDB 简介

第 1 章	介绍与总览	2
1.1	打造一个社交网络	3
1.2	模型—视图—控制器 (MVC)	3
1.3	纯 JavaScript	4
第 2 章	Node.js	5
2.1	安装 Node.js	5
2.2	Express	6
	模板	8
2.3	事件	10
2.4	Socket.io	12
2.5	模块与 CommonJS	14
第 3 章	Backbone.js	16
3.1	模型	16
3.2	视图	17
3.3	集合	20
3.4	路由和历史	21
第 4 章	MongoDB	23
4.1	数据访问	23
4.1.1	写入	24
4.1.2	查询	27
4.1.3	索引	28
4.1.4	映射规约	29
4.2	携手 Node.js	31
4.3	并发访问	31

第二部分 建立社交网络

第 5 章	创建项目	36
5.1	目录结构	36
	文件列表	37
5.2	包定义	38
5.3	网站服务器	39
5.3.1	索引模板	41
5.3.2	应用程序 JavaScript	42
5.3.3	应用程序类	43
5.3.4	索引视图对象	43
第 6 章	认证	45
6.1	账户	45
6.2	路由	48
6.2.1	身份验证	49
6.2.2	身份验证处理程序	50
6.3	注册	51
6.3.1	注册模版	51
6.3.2	注册处理程序	53
6.4	登录	54
6.4.1	登录模板	54
6.4.2	登录处理程序	56
6.5	忘记口令	57
6.5.1	忘记口令模板	57
6.5.2	忘记口令处理程序	58
6.6	重置口令	60
6.6.1	重置口令模板	60
6.6.2	重置口令处理函数	61
6.7	整合	62
	Node.js	62
第 7 章	用户界面	65
7.1	账户明细	65
7.1.1	账户明细模板	66
7.1.2	账户明细处理程序	67

7.2	联系人列表	68
7.3	活动流	69
7.3.1	活动流模板	69
7.3.2	活动流处理程序	72
7.4	数据模型	73
7.5	整合	76
7.5.1	Backbone	76
7.5.2	Node.js	78
第 8 章	交朋友	82
8.1	联系人列表	82
8.1.1	联系人列表模板	82
8.1.2	联系人列表处理程序	86
8.2	添加联系人	87
8.2.1	添加联系人模板	87
8.2.2	添加联系人处理程序	89
8.3	删除联系人	91
8.3.1	删除联系人模板	91
8.3.2	删除联系人处理程序	92
8.4	评论	93
8.4.1	评论模板	94
8.4.2	评论处理程序	96
8.5	整合	97
8.5.1	Backbone	97
8.5.2	Node.js	99
第 9 章	聊天	109
9.1	重构	109
9.2	连接到聊天服务器	110
9.2.1	Backbone	111
9.2.2	Node.js	113
9.3	发送和接收聊天消息	114
9.3.1	Backbone	116
9.3.2	Node.js	121
9.4	整合	121
9.4.1	Backbone	121
9.4.2	Node.js	125

第 10 章 实时互动	134
10.1 增加自定义事件	134
10.1.1 触发事件	135
10.1.2 添加监听器	135
10.2 联系人登录通知	137
10.2.1 Backbone.js	137
10.2.2 Node.js	139
10.3 状态更新	141
10.3.1 Backbone.js	141
10.3.2 Node.js	143
10.4 整合	144
10.4.1 Backbone.js	144
10.4.2 Node.js	155
10.4.3 静态文件	166
第 7 章 用户界面	151
7.1 账户列表	151
7.1.1 账户列表创建	151
7.1.2 账户列表增删改	152

Node.js、Backbone.js 和 MongoDB 简介

第1章

介绍与总览

互联网已经成为发展最快的技术领域之一，它还在加速。对于希望通过编写软件来获得收入的人来说，这既是好消息，也是坏消息。今天，优秀的开发人员拥有难得的机会，可以做他们喜欢的事，拓展视野，不断进步，并从工作中得到更大的满足，只要他们愿意付出必要的努力，去理解海量的、快速增长的知识。

精彩的职业生涯是有代价的。作为一名软件开发者，你必须不断寻找下一个优秀的工具，来帮助你更多、更好、更快地实现梦想。你在 10 年后面对的东西，与今天面对的东西会完全不同。本质上说，你需要多次再培训自己，才能保持好的状态。

马尔科姆·格拉德威尔（Malcolm Gladwell）在他 2008 年所著的《Outliers》一书中指出，专业水准的融汇贯通需要 10 000 小时的努力。即使是天才也需要投入时间来收获成功，平庸者和精湛者之间的差异归结于个人实践经验的多少。阅读本书这样的书籍，能让你成为精湛者。现在你正在花费额外的时间，接触这门前沿的手艺。未来即将到来，你会首先获得它的好处。

Node.js 已经将一大批软件开发者引入了面向事件编程。无论你的技术背景如何，只要有开放的心态，放下对 JavaScript 的偏见，就会领悟到在这个疯狂应用多线程的世界里，单线程编程是多么强大。更重要的是，你会更欣赏事件处理，在其他编程语言中处理多线程问题时，也会有所帮助。

JavaScript 是一种独特的编程语言，有时会被误解，现在它在开发人员的工具箱中终于得到了应有的地位。由于开发 JavaScript 应用程序的工具集不断完善和成熟，你会看到全球范围内这种编程语言的重要性在持续增长。

1.1 打造一个社交网络

本书将手把手指导你搭建一个社交网络，类似于 LinkedIn、MySpace 或 Facebook。使用 Node.js, Backbone.js 和 MongoDB 作为工具，你将学会如何创建一个快速响应的应用，并且可以扩展到几百万用户。

由于是例子，本书中介绍的许多组件会走捷径，使用 Node 或 MongoDB 提供的内建方法，来展示某项功能，但在“真正”大型的部署中是不合适的。如果出现这样的捷径，我会给出特别提示，并探讨如何转向更具扩展性或更容易修改的结构。本书的挑战是在清晰性和构建真正实用的应用之间取得平衡。

什么是社交网络？“社交网络”这个词组似乎传达了很多意义（在行为科学领域，确实如此），让我们逐词解读，并应用于因特网。“网络”是互联的一组系统，它可以是任何东西，从遍布全国的高速公路网，到学校实验室里的一排计算机，或是名片盒里的专业联系人。“社交”这个词指的是生物体的交互（如动物或人），以及它们作为群体的存在。所以社交网络在这里指的是一群互联、互动的人。

在社交网络里，人的因素高于一切。在构建任何软件时，如果不首先（并且持续）考虑最终用户（不论是客户、教授还是你自己），就针对某一特定功能或目标进行开发，这就是不负责任。要抵制为了技术的缘故而继续编程之路的冲动，直至你能看清楚工作的最终目的。

我们说打造一个社交网络时，当然不可能是打造这里定义的社交网络。你要创建的是一个论坛，一条道路，让社交网络生根成长。系统的每一项功能都为这一目标服务，消除用户的障碍，提供足够的功能来推动、鼓励和促进沟通，而没有过多的修饰。这是一条充满困难的道路，但也是区分伟大和平庸的产品的唯一途径。

1.2 模型—视图—控制器（MVC）

本书多处提到并使用“模型—视图—控制器（MVC）”设计模式，进行服务器端和前端编程。有人认为，MVC 在 Web 上是随着 Ruby on Rails 的增长而流行的，但它最早是在 20 世纪 70 年代为 Smalltalk 平台开发的。

MVC 的主要作用是将系统拆成 3 个弱耦合的部件。

模型：包含了要读取或操作的数据的一种结构。

视图：用户与模型产生互动的界面。

控制器：在视图和底层模型之间代理用户操作。

模型和控制器通常成对出现。在本书中，控制器的工作可作为用户和模型之间来回传递消息的合约。虽然一个控制器可以操控多个模型，但这样做不好，我们建议一个控制器对应一个模型。

视图不一样。就像现实生活中一样，软件往往提供多种方式来展现同一信息。例如，音频的文字版本包含了和原始内容一致的信息，但它的内容传播更容易、更方便。因特网到处都是这样的例子：很多网络服务的数据展现同时采用 JSON 和 XML，用两种不同的格式提供相同的信息。

1.3 纯 JavaScript

Node、Backbone 和 MongoDB 让你使用单一编程语言，专注于应用逻辑，最终降低系统各部分之间的连接次数。你会看到这是一种引人注目的编程方式，因为客户端 UI、后端服务器逻辑和数据库持久之间的界限模糊了，几乎形成一个生动的系统。随着实时网络逐渐进入生活，情况就更清楚了。数据在各个应用中舞动，甚至经过多个用户之手，所有事情就像一齐发生在单个过程中。

有些缺陷是需要预先避免的。虽然连接方式很强大，使用同一种方言，但是在底层，你的工作仍然是跨浏览器、服务器和数据库的编程。有些 JavaScript 开发范式着眼点略有不同，这取决于它们的主要目标是服务于 UI（如在浏览器中使用的 Backbone）、验证（如在服务器端常用的 Node）或持久（如本书中提到的 MongoDB）。你需要时刻保持警惕，注意数据要去哪里，是否阻塞你自己的进程，如何监听进出的事件并做出反应。这可能是一个挑战，但就像其他有许多灵活部分的系统一样，在实验中会有许多有趣的收获。

今天的互联网不同于 20 世纪 90 年代的互联网。在以前，用户和一个网站之间的交互基本上属于信息消费。Web 服务器生成大量静态网页，用户在页面之间穿梭。当然也有动态元素，但互动的信息流主要是限于请求和应答。多年的研究都集中于优化客户端—服务器的信息流，可以肯定地说目前对这方面的理解很透彻。

大约在 IE6 开始出现时，一个微妙而根本的转变开始占据上风。因特网用户变得越来越适应和内行，电脑变得更为强大，宽带连接开始变得常见。人们使用因特网不再是主要为了信息和交易，而是在社交和娱乐上花更多时间。因特网现在是媒体频道，但与之前的电视、广播和报纸不同。

现在互联网络用户不是消费数据，而是产生难以想象的大量数据。Web 服务器和浏览器作为信息消费者的传统观念仍然存在，但了解这一点仅仅看到了信息发布者很少一部分能力。现在的重点是使人控制他们的体验，并借助他们创造的数据去实时改变、改进和提高这些体验。这是一个崭新的世界，网络服务器和程序员不再是体验的来源，而是促进者。

Node.js 是转向“因特网即体验”范式的新型技术之一。

2.1 安装 Node.js

如果你没有安装过 Node.js，那么首先你得安装它。你可以从 Node 项目主页 (<http://nodejs.org>) 下载到适合自己的操作系统的安装包。主页提供了用于 UNIX、Mac 和 Windows 的二进制安装包。如果你真心喜欢冒险，可以从 Node 的 GitHub 版本库中找到链接，安装一个开发快照版。在撰写本书时，Node 的稳定发行版本是 0.8.12。