

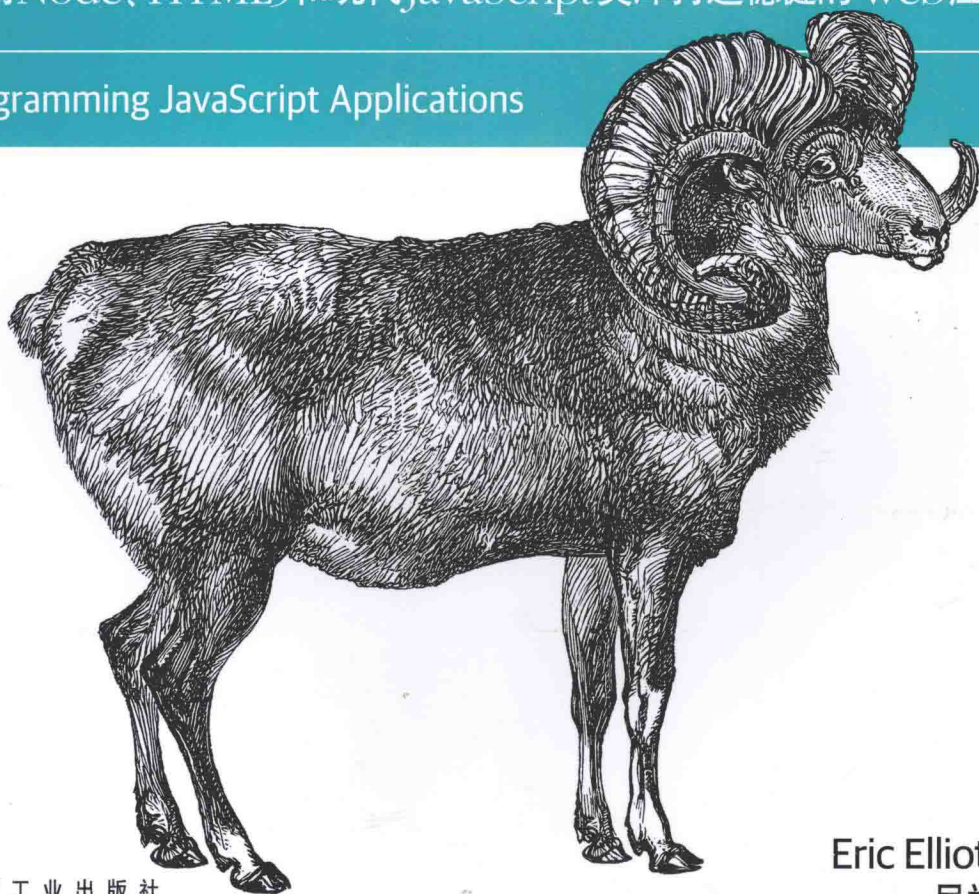
O'REILLY®



# JavaScript 应用程序设计

使用Node、HTML5和现代JavaScript类库打造稳健的Web应用

Programming JavaScript Applications



Eric Elliott 著  
吴斌 译



机械工业出版社  
China Machine Press

# JavaScript应用程序设计

## 使用Node、HTML5和现代JavaScript类库打造稳健的Web应用

*Eric Elliott* 著

吴斌 译

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社



## 图书在版编目 (CIP) 数据

JavaScript应用程序设计: 使用Node、HTML5和现代JavaScript类库打造稳健的Web应用/ (美) 利奥特 (Elliott, E.) 著; 吴斌译. —北京: 机械工业出版社, 2015.6

(O'Reilly精品图书系列)

书名原文: Programming JavaScript Applications

ISBN 978-7-111-50694-2

I. J… II. ①利… ②吴… III. Java语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2015) 第146254号

北京市版权局著作权合同登记

图字: 01-2014-7784号

©2014 James M. Westerfield.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2015. Authorized translation of the English edition, 2014 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2014。

简体中文版由机械工业出版社出版 2015。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。

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

书 名/ JavaScript应用程序设计:

使用Node、HTML5和现代JavaScript类库打造稳健的Web应用

书 号/ ISBN 978-7-111-50694-2

责任编辑/ 陈佳媛

封面设计/ Randy Comer, 张健

出版发行/ 机械工业出版社

地 址/ 北京市西城区百万庄大街22号 (邮政编码 100037)

印 刷/ 北京瑞德印刷有限公司

开 本/ 178毫米×233毫米 16开本 14.5印张

版 次/ 2015年9月第1版 2015年9月第1次印刷

定 价/ 59.00元 (册)

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

购书热线: (010)68326294; 88379649; 68995259

投稿热线: (010)88379604

读者信箱: hzit@hzbook.com

# 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。无论是通过书籍出版，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

# 译者序

说来也巧，结识JavaScript这门语言还是在我上初中那会儿，那时候微软与网景两大浏览器巨头还在为市场份额互相掐架，许多人仍旧搞不清JScript与JavaScript之间的关系，市面上少有的几本与JavaScript有关的教程除了基本语法介绍外，就是挨个将DOM属性罗列一遍，看得让人着急。就当时Web平台的现状而言，很难想象这门语言今后会与“应用”两个字紧密联系在一起。时光一晃就是10年，2011年，当我以前端开发工程师的身份加入阿里时，我才领会到这门语言背后的真正魔力。

JavaScript正在成为编程界的一门通用语言，它已经从母体，即负责信息展现与内容分发的Web平台中慢慢走出来，开始涉足并接管更多的领域，从服务端系统搭建到创建效率工具，从移动OS平台到各类沉浸感十足的3D游戏，JavaScript正在不断地向外界证明它不仅仅只是一门“胶水语言”，到目前为止，它的语言规范仍在向前演进。

本书从JavaScript独有的语言特性开始，作者以灵活、简洁、优雅为中心思想，一路向我们介绍了构建一款完整JavaScript应用所需的方方面面。要确保JavaScript应用既稳健又具伸缩性，开发者除了掌握JavaScript本身的语言特性，还需要对软件工程中的API设计、关注点分离、异步编程、单元测试等重要的概念做深入的了解，除此以外在项目开发中采用良好且一致的编码风格也是降低应用后期维护成本的有效手段，所有这些在本书中均有囊括。无论你是PC/移动端的WebApp开发者，还是仅对JavaScript这门语言所独有的特性感到好奇的入门者，这本书都不会让你失望。

本书作者Eric Elliott是一位JavaScript专家，他的过往工作经历与头衔中无处不闪烁着JavaScript字样，同时他还是一位音乐人、一位摄影师和一名电影爱好者。也许出于这个原因，在本书中你能明显感觉到Eric Elliott所注入的艺术气息，无论是解决问题的思路

还是代码的编排，Eric Elliott都在试图向读者呈现更为优雅的解决方案。当然，由于专业知识与精力有限，译者在翻译过程中难免出现疏漏与失误，恳请广大读者批评指正。

感谢机械工业出版社华章公司的陈佳媛编辑对我的信任与帮助，感谢我的师兄刘洋洋(额台)对很多关键的技术性问题提出的宝贵意见，感谢阿里智能云前端团队的同事於振华(方元哥)、李春城(琦轩)、曹仁(曹纯)在翻译过程中给予我诸多建议和指导，还要感谢我的两位好朋友蒙克、赤目给予我的鼓励。感谢家里“领导”对我翻译工作的理解与支持，最后特别感谢我的母亲，过年在家为了赶稿没少让您操心。

吴斌(柳翎)

2015年4月于杭州

# 目录

|                                 |    |
|---------------------------------|----|
| 前言 .....                        | 1  |
| 第1章 JavaScript的革命时代 .....       | 7  |
| JavaScript的特性 .....             | 8  |
| 性能 .....                        | 8  |
| 对象 .....                        | 8  |
| 语法 .....                        | 9  |
| 函数是一等公民 .....                   | 9  |
| 事件 .....                        | 9  |
| 复用性 .....                       | 10 |
| 结论 .....                        | 10 |
| 解构现代JavaScript应用 .....          | 10 |
| 基础架构 .....                      | 10 |
| JSON —— 数据存储与通信 .....           | 11 |
| NoSQL数据库 .....                  | 12 |
| RESTful JSON Web Services ..... | 12 |
| 第2章 函数 .....                    | 15 |
| 减少隐式副作用 .....                   | 16 |
| 函数声明 .....                      | 18 |
| lambdas .....                   | 22 |
| 立即执行函数表达式 .....                 | 23 |

|                          |           |
|--------------------------|-----------|
| 方法上下文 .....              | 25        |
| 函数作用域 .....              | 27        |
| 变量提升 .....               | 28        |
| 闭包 .....                 | 30        |
| 方法API的设计 .....           | 33        |
| 参数命名 .....               | 33        |
| 多态函数 .....               | 34        |
| 范型编程与集合多态 .....          | 37        |
| 链式调用与流式API .....         | 40        |
| 函数式编程 .....              | 41        |
| 无状态函数(纯函数) .....         | 43        |
| 偏函数应用与函数加里化 .....        | 44        |
| 异步操作 .....               | 45        |
| 回调函数 .....               | 46        |
| Promises与Deferreds ..... | 46        |
| 小结 .....                 | 49        |
| <b>第3章 对象 .....</b>      | <b>50</b> |
| 过时的类继承 .....             | 51        |
| 流式JavaScript .....       | 53        |
| 原型 .....                 | 56        |
| 原型代理 .....               | 56        |
| 原型克隆 .....               | 58        |
| 享元模式 .....               | 60        |
| 创建对象 .....               | 61        |
| 工厂函数 .....               | 63        |
| 使用Stamps进行原型继承 .....     | 66        |
| 小结 .....                 | 71        |
| <b>第4章 模块 .....</b>      | <b>73</b> |
| 模块化编程原则 .....            | 74        |
| 接口 .....                 | 75        |

|                                              |            |
|----------------------------------------------|------------|
| 模块模式 .....                                   | 79         |
| AMD规范 .....                                  | 81         |
| 加载器插件 .....                                  | 82         |
| CommonJS模块规范 .....                           | 84         |
| npm包管理器 .....                                | 85         |
| ES6模块规范 .....                                | 87         |
| 基于CommonJS、npm、Grunt与Browserify构建客户端应用 ..... | 89         |
| 定义应用 .....                                   | 89         |
| 特性实现 .....                                   | 91         |
| 构建与部署 .....                                  | 94         |
| 小结 .....                                     | 98         |
| <b>第5章 关注点分离 .....</b>                       | <b>99</b>  |
| 客户端的关注点 .....                                | 100        |
| 模块管理 .....                                   | 101        |
| 模型－视图－控制器 .....                              | 113        |
| 表现层与Dom操作 .....                              | 116        |
| 服务端关注点 .....                                 | 123        |
| Node与Express入门 .....                         | 124        |
| 在Node中组织代码 .....                             | 125        |
| 小结 .....                                     | 132        |
| <b>第6章 访问控制 .....</b>                        | <b>134</b> |
| 身份验证 .....                                   | 134        |
| 密码认证 .....                                   | 134        |
| Credential类库 .....                           | 138        |
| 多因素身份验证 .....                                | 140        |
| 联合认证与授权认证 .....                              | 142        |
| 应用功能授权 .....                                 | 144        |
| 应用授权 .....                                   | 144        |
| OAuth 2.0协议 .....                            | 145        |
| 小结 .....                                     | 146        |

|                          |            |
|--------------------------|------------|
| <b>第7章 日志</b>            | <b>147</b> |
| 调试                       | 147        |
| 服务端数据监控                  | 149        |
| 安全/入侵检测                  | 149        |
| 审计                       | 150        |
| 业务指标分析                   | 150        |
| 传播力度                     | 151        |
| 日志清单                     | 151        |
| 应用请求日志                   | 152        |
| 异常日志                     | 156        |
| 日志输出示例                   | 158        |
| 日志预警                     | 161        |
| 跟踪业务指标（效果跟踪）             | 162        |
| 性能分析与优化                  | 162        |
| 客户端事件记录                  | 163        |
| 日志数据挖掘                   | 164        |
| 小结                       | 165        |
| <b>第8章 构建RESTful API</b> | <b>166</b> |
| 可用性                      | 167        |
| 专一性                      | 167        |
| 一致性                      | 169        |
| 自描述的超媒体                  | 177        |
| 功能可见性                    | 177        |
| 超媒体即应用状态引擎               | 178        |
| HTML：一种特别的API媒介类型        | 180        |
| Jade                     | 180        |
| Jiron                    | 182        |
| 响应式API                   | 184        |
| 性能优化                     | 186        |
| 小结                       | 187        |

|                                   |            |
|-----------------------------------|------------|
| <b>第9章 特性开关 .....</b>             | <b>188</b> |
| 组织特性 .....                        | 188        |
| 特性的规模 .....                       | 188        |
| 特性编组 .....                        | 189        |
| 特性的生命周期 .....                     | 189        |
| 开发阶段 .....                        | 190        |
| 预发布阶段 .....                       | 190        |
| 灰度测试 .....                        | 190        |
| 正式发布 .....                        | 190        |
| 默认激活 .....                        | 191        |
| 完整集成 .....                        | 191        |
| 客户端实现 .....                       | 191        |
| 小结 .....                          | 195        |
| <b>第10章 应用工程国际化 .....</b>         | <b>196</b> |
| 小结 .....                          | 198        |
| <b>附录A JavaScript编码风格指南 .....</b> | <b>199</b> |

# 前言

在本书中我们先后提到了不少Web开发类书籍，但对于那些想从头开始学习构建完整JavaScript应用的读者来说，市面上真正值得推荐的JavaScript教程又显得十分稀缺。与此同时，现在几乎每一个互联网创业公司的人员配备中都少不了资深JavaScript应用程序开发者的角色。本书旨在帮助人们了解如何构建易于扩展和维护的完整JavaScript应用。

本书并不打算向读者详细讲解JavaScript语言基础，它基于你现有的JavaScript知识结构，向你展示那些会让你的编码工作长久受益的JavaScript特性与技巧。一般来说，随着应用程序的规模不断扩大，添加新功能与缺陷修复开始变得越来越难以下手，代码随之变得脆弱、臃肿，甚至一个非常小的改动都可能带来大范围的代码重构。如果你在编码过程中遵循了本书中介绍的设计模式，你会发现代码开始变得灵活而富有伸缩性，对代码的细微变更不会再对其他模块产生负面影响。

本书的内容主要聚焦在客户端JavaScript应用设计上，同时也会穿插一些较为基础的Node、RESTful API等服务端技术介绍。应用逻辑前端化已经是大势所趋，以往我们常常将模板渲染、与第三方服务通信等任务交给服务端去处理，诸如此类的工作如今通常在浏览器中完成。

事实上，现今的JavaScript应用已经在浏览器中扮演起了传统桌面级应用的角色。当然，这一切仍然离不开服务端的技术支撑，像静态资源托管、动态模块加载、数据持久化、操作日志记录以及第三方API服务调用等，这些都属于服务端的职能范畴。

本书内容涵盖：

- 面向应用开发者的JavaScript特性与最佳实践。
- 代码组织、模块化与复用。

- 客户端关注点分离（MVC等）。
- 服务端与API的通信机制。
- 用Node.js设计并实现RESTful API。
- 构建、测试、团队协作、应用部署与扩展。
- 应用工程的国际化。

## 本书目标读者

具备一定的JavaScript开发基础，有至少1或2年的JavaScript实践经验，并想了解如何使用JavaScript开发大规模Web或企业级应用的开发者。

抱有强烈的学习热情，尤其是对闭包、函数式编程、原型继承这些JavaScript语言独有的强大特性想做了解与尝试的编程入门者。（即使你是第一次听到这些概念。）

也许你想在下一个具有挑战性的JavaScript项目中使用TDD（测试驱动开发）技术，本书所有的代码范例中均包含测试用例。在你读完本书后，代码自测的意识与方法也会随之建立起来。

## 不太适合本书的读者

本书不面向JavaScript初学者，涉及JavaScript基础技术点的地方会一带而过。如果你想对这些技术点进行深入了解，可以在Douglas Crockford的《JavaScript精粹》（O'Reilly, 2008）以及David Flanagan的《JavaScript 权威指南》<sup>编注1</sup>（O'Reilly, 2011）中找到相关解读。由著名的“四人帮”（Erich Gamma、Richard Helm、Ralph Johnson和John Vlissides）合著的《设计模式：可复用面向对象软件的基础》<sup>编注2</sup>（<http://bit.ly/1pwzcUc>）（Addison-Wesley, 1994）会帮助你更好地理解设计模式。

Google与维基百科同样是信息查阅的好帮手，对软件设计模式来说，维基百科是一份非常不错的参考资料。

如果这是你第一次接触JavaScript，我建议你在步入正篇之前先把基础功课做足。Marijn Haverbeke所著的《Eloquent JavaScript》（<http://eloquentjavascript.net/>）（No Starch Press, 2011）是我个人比较喜爱的一本JavaScript入门教程，还有上文中提到

---

编注1：本书中文版已由机械工业出版社引进出版，书号为978-7-111-37661-3。

编注2：本书中文版已由机械工业出版社引进出版，书号为978-7-111-07575-7。

的《JavaScript 精粹》，最好每一个章节都通读一遍。另外，特别留意下附录A，这是JavaScript开发前辈们对自己所犯的错误的系统总结，从中你可以获得一些启发。

## 单元测试

很难估量单元测试在应用中的重要性，本书中几乎所有的代码样例中均附有单元测试用例。编写测试用例让你对功能实现了解得更为透彻，对方案与接口设计的把握越来越谨慎，设计测试用例的过程同样是迫使你思考将代码解耦的过程。保持编写单元测试与解耦代码的习惯会使你在今后的工作中持续受益，相信在读完本书之后你会收获这种编码习惯。

关于单元测试与编码风格的更多内容，请参见本书附录A。

## 本书的约定

本书使用下列排版约定：

### 斜体 (*italic*)

用于表示新术语、URL、电子邮件地址、文件名、文件扩展名。

### 等宽字体 (`constant width`)

用来表示程序列表和代码片段，包括变量或函数名、数据库、数据类型、环境变量、语句和关键字等。

### 等宽加粗字体 (**`constant width bold`**)

用来表示命令或者其他用户直接输入的文本。

### 等宽斜体 (`constant width italic`)

用来表示可被替换的字符或文本，这些字符在合适的场景和特定的条件下会被替换成其他的值。

---

**注意：** 表示一种提示、建议或一般的消息提醒。

---

---

**警告：** 表示一种警告。

---

## Safari® Books Online

Safari在线图书是一个数字图书馆，读者可以在这个图书馆里自选图书，在这里可以搜索到超过7500本与技术和创作相关的书籍和视频，在这里可以迅速找到你想要的内容。

订阅之后,你就可以阅读在线图书馆的任意图书的任意章节和任意视频。你还可以将图书下载到手机和移动设备里。在纸质书籍出版前就可以抢先阅读,甚至可以抢先阅读作者手稿,并实时给作者反馈。同时还可以复制和粘贴实例代码、组织你的收藏内容、下载章节、将关键段落加入书签、创建笔记、打印出来,你既可以节省时间,又可以提升阅读效率。

O'Reilly Media已经将本书上传至Safari在线图书馆里了。如果想在线阅读本书和其他相关内容,请免费登录<http://my.safaribooksonline.com>。

## 联系我们

对于本书的评论或问题请联系出版商:

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

我们为本书制作了一个Web页面,页面中包含了简介、样章以及其他信息。可以从这里访问这个页面:

<http://bit.ly/programming-ja>

如果要留言或者提交关于本书的技术问题的反馈,请发邮件至:

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

本书的更多信息、资源、参考文献和新闻,请登录出版社官网:

<http://www.oreilly.com.cn>  
<http://www.oreilly.com>

Facebook: <http://facebook.com/oreilly>

Twitter: <http://twitter.com/oreillymedia>

## 致谢

感谢@JS\_Cheerleader([https://twitter.com/JS\\_Cheerleader](https://twitter.com/JS_Cheerleader))给予我的鼓励,你提供的JavaScript在线资源非常棒!

感谢Brendan Eich一直以来不知疲倦地推动JavaScript与Web技术向前发展。感谢整个O'Reilly团队,感谢Simon St. Laurent,是你让这本书有机会与读者见面,我知道你一直都在给我加油鼓劲!感谢Brian McDonald对本书的宝贵意见,感谢Meghan Blanchette,你总是不断地给我打气!感谢以下每一个人对本书的技术审校与建议:

- César Andreu
- James Halliday (Substack)
- Hugh Jackson
- Ramsey Lawson
- Shelley Power
- Kyle Simpson
- Kevin Western

特别感谢本书中所有开源软件项目的作者,向开源社区中那些不断提升JavaScript应用程序用户体验的开发者致敬,身为软件开发者,我们正站在巨人的肩膀上。

