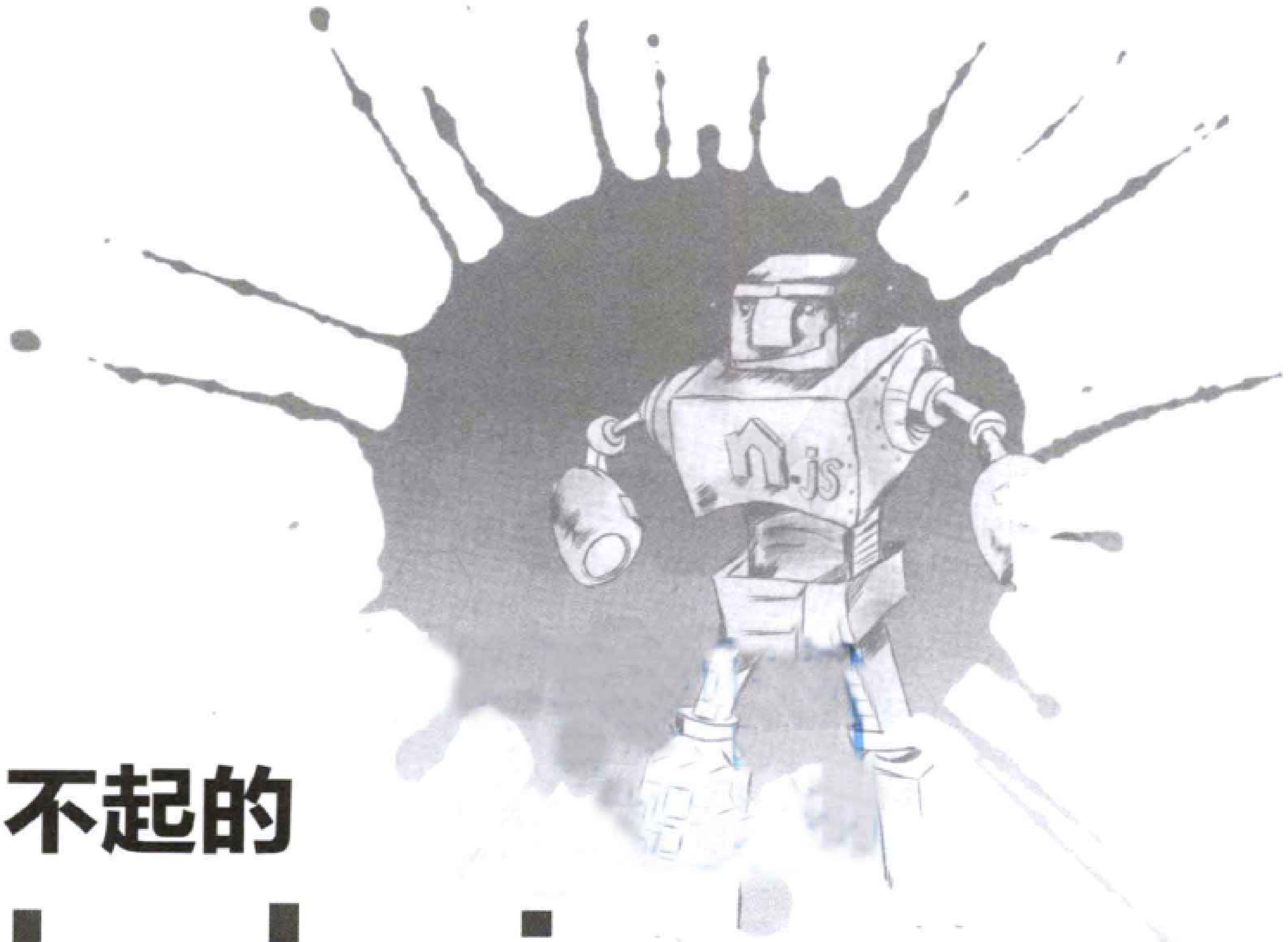
SMASHING Node.js: JavaScript Everywhere



了不起的 Node.js

将JavaScript进行到底

Guillermo Rauch 著　Goddy Zhao 译

WILEY

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书是一本经典的Learning by Doing的书籍。它由Node社区著名的 Socket.IO作者——Guillermo Rauch，通过大量的实践案例撰写，并由 Node社区非常活跃的开发者——Goddy Zhao翻译而成。

本书内容主要由对五大部分的介绍组成：Node核心设计理念、Node核心模块API、Web开发、数据库以及测试。从前到后、由表及里地对使用 Node进行Web开发的每一个环节都进行了深入的讲解，并且最大的特点就是通过大量的实际案例、代码展示来剖析技术点，讲解最佳实践。

版权贸易合同登记号　图字：01-2013-8005

图书在版编目（CIP）数据

了不起的Node.js：将JavaScript进行到底 / (美)劳奇 (Rauch,G.) 著；赵静译. —北京：电子工业出版社，2014.1
书名原文：SMASHING Node.js:JavaScript Everywhere
ISBN 978-7-121-21769-2

Ⅰ. ①了… Ⅱ. ①劳… ②赵… Ⅲ. ①JAVA语言—程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2013）第257626号

策划编辑：张春雨
责任编辑：贾　莉
印　　刷：中国电影出版社印刷厂
装　　订：中国电影出版社印刷厂
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：19.5　字数：436千字
印　　次：2014年1月第1次印刷
定　　价：79.00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

从2009年Ryan Dahl着手开发Node.js开始，到现在Node已经快4岁了。尽管它至今还未发布1.0版本，甚至连alpha都还没有，但是Node这几年的发展大家都是有目共睹的。越来越多的开发者和公司开始尝试使用 Node.js：微软在其Azure云上支持了部署Node.js应用，它同时还是Node Windows版本的主要贡献者；雅虎主站大量使用了Node；LinkedIn也使用Node为其移动应用提供服务器端服务。除此之外，Node官方Wiki页面（https://github.com/joyent/node/wiki/Projects,-Applications,-and-Companies-Using-Node）列出了更多在使用Node的公司和项目。可以说，Node以其异步 IO、服务器端JavaScript的特点为Web开发掀开了新的篇章。

然而，尽管Node这几年发展神速，但是相关的书籍、资料却很少，尤其在国内就更是寥寥无几。这就让我萌发了为国内Node爱好者翻译一本 Node书籍的想法，于是我就想到了*SMASHING Node.js: JavaScript Everywhere*。之所以选择这本书，是因为：首先，这本书的作者Guillermo Rauch在Node社区非常有名，作为Socket.IO的作者、Express的开发者之一，他为社区贡献了很多质量很高的Node模块；其次，我此前碰巧有幸受原书出版社WILEY之邀，担任了原书的技术审校，所以我对原书内容非常熟悉；最后，也是主要的原因，是因为这本书有大量的实践案例，我个人始终认为学习技术的最佳方式就是实践，而且本书中的案例在阐述技术点的同时，还非常具有实践价值。在上述三点原因的驱使下，最终让我决定向电子工业出版社引荐此书，最后也很高兴出版社能够认同这本书并决定引进此书。

本书根据Web开发的流程，从Node核心概念——事件轮询、V8中的 JavaScript的介绍，Node核心库——TCP、HTTP的讲解，到应用层开发——Connect、Express、Socket.IO的实践，再到数据库——MongoDB、Redis、MySQL的剖析，最后到测试——Mocha、BDD的阐述，每个环节都一一做了深入的讲解。另外，本书始终贯穿了 Learning by Doing的理念，每一章都有大量的实践案例、代码展示，以编写实际代码的方式让读者掌握技术、同时教会读者如何将其运用到实际项目中。总的来说，本书确实是一本学习Node的好书。

最后，在本书翻译过程中要特别感谢来自淘宝网工程师——易敛（花名）以及聚美优品工程师——邵信衡给予的帮助。另外，还要感谢本书的编辑张春雨和贾莉的辛苦工作，以及我太太的大力支持。

希望本书能够为广大Node开发者带来帮助，谢谢！

Goddy Zhao（赵静），SuccessFactors（SAP子公司）软件工程师。毕业于复旦大学，先后在IBM、淘宝工作过，专注于企业级富客户端Web应用的开发，擅长前后端相结合的技术解决方案。曾与人合译过多本前端图书，并曾在沪JS及D2前端技术论坛担任过主持人和演讲嘉宾。

前言

绝大部分Web应用都包含客户端和服务器端两部分。服务器端的实现往往比较复杂、麻烦。创建一个简单的服务器都要求对多线程、伸缩性以及服务器部署有专业的技术知识。除此之外，由于客户端软件是用HTML和JavaScript来实现的，而服务器端核心代码通常都是用静态编程语言实现的，所以，开发Web应用经常会有错乱的感觉。由于这种前后端开发语言的差异，不得不让开发者使用多种编程语言，同时还要对特定的程序逻辑事先做好设计选型。

几年前，要用JavaScript来实现服务端软件几乎是想都不敢想的一件事情。糟糕的性能、不成熟的内存管理以及缺乏操作系统层面的集成，不解决这些问题，JavaScript很难成为一门服务器端的语言。作为 Google Chrome浏览器的一部分，新的V8引擎能够解决前两个问题。V8是一个开源的项目，通过简单的API就可以将其集成进去。

Ryan Dahl洞察到了这样一个机会，可以通过将V8内嵌到操作系统的集成层，来让JavaScript享受到底层操作系统的异步接口，从而实现将其带到服务器端的目的。这就是Node.js的设计思路。这么做的好处是显而易见的。程序员们可以在客户端和服务器端使用同样的编程语言了。 JavaScript动态语言的特性使得开发和试验服务器端代码变得很自由，使得程序员们摆脱了传统那种又慢又重的编程模式。

Node.js迅速蹿红，衍生了一个强大的开源社区、支持企业，甚至还拥有属于自己的技术大会。我把这种成功归结于它的简洁，高效，同时提高了编程生产力。我很高兴V8成为其一小部分。

本书将带着读者学习如何基于Node.js为Web应用构建服务器端部分，同时还会带着大家学习如何组织服务器端异步代码以及如何与数据库进行交互。

好好享受这本书带来的乐趣吧！

Lars Bak,Virtual Machinist

介绍

2009年年末，Ryan Dahl在柏林的一个JavaScript大会上宣布了一项名为Node.js（http://nodejs.org/）的新技术。有意思的是，出乎所有参会者的意料，这项技术居然不是运行在浏览器端的，要知道浏览器端对于JavaScript来说绝对是拥有霸主地位的，这是毋庸置疑的。

这项技术是关于在服务器端运行JavaScript的。当时，这简单的一句描述，瞬间让听众眼前一亮，同时也宣告了这项新技术的发布大获成功。

如果成真的话，以后开发Web应用就只需要一种语言了。

毫无疑问，这是当时所有人的第一想法。毕竟，要开发一个现代富客户端Web应用，必须要对JavaScript非常熟悉和了解，然而，对于服务器端的技术来说，就有很多不同的选择，而且都需要专业的要求。拿Facebook来说，他们最近透露其总代码库中JS的代码量是服务器端语言PHP的四倍。

不过Ryan感兴趣的是为大家展示一个简洁又强大的示例程序。他展示了一个Node.js中的“hello world”程序——创建一个Web服务器。

```
var http = require('http');
var server = http.createServer(function (req, res) {
  res.writeHead(200);
  res.end('Hello world');
});
server.listen(80);
```

这样一个Web服务器并非只是个“玩具”，相反，它是一个高性能的Web服务器，甚至，在某些场景下，比现有如Apache和Nginx这样的Web服务器性能还要好。Node.js被称为是一个将设计网络应用导向正确道路的特殊工具。

Node.js快速高效的优点得益于一种叫做事件轮询（event loop）的技术，以及其构建于V8之上，V8是Google为Chrome Web浏览器设计的 JavaScript解释器和虚拟机，它运行JavaScript非常快。

Node.js改变了Web开发模式。你无须再将书写部署到独立安装的Web服务器中去运行，如

传统的LAMP模式，它通常包含了PHP环境和 Apache服务器。

正如本书正文中将要介绍的，获得Web服务器完全的控制权催生了另外一类基于Node.js开发的应用：实时Web应用。在一个服务器端和众多客户端进行快速的数据传输，在Node开发中变得越来越常见。这意味着你既可以创建更高效的程序，又能成为社区的一部分，推进理想的Web开发模式。

有了Node，你就有了主动权。同时，本书也会详细介绍这种能力背后所带来的新的挑战和责任。

目标

首先最重要的是，本书是一本关于JavaScript的书。你必须具备一定的JavaScript知识，同时，一开始我也花了一章来介绍JavaScript的相关概念，根据我的经验来看，这是非常有帮助的。

正如你将会学到的，Node.js努力为浏览器端开发者提供一个舒服的开发环境。有些常用的表达式，如`setTimeout`和`console.log`，并非语言标准的一部分，而是浏览器添加的，它们在Node.js中也能使用。

在你理清思绪，准备就绪后，就可以开始Node之旅。作为其核心的一部分，Node自带了很多有用的模块，以及一个名为NPM的简单包管理器。本书从教你如何仅使用Node核心模块构建应用开始，随后教你使用一些最有用的社区开发者基于Node开发的模块来开发应用，这些模块都可以通过NPM安装获得。

在介绍如何用专门设计的模块解决特定问题前，我通常会先介绍如何在不使用模块的情况下解决此问题。理解一个工具最好的方式就是首先搞明白为什么会有这个工具。因此，在学习某个Web框架前，你会先学习为什么用它要比使用Node.js原生的HTTP模块要好。在学习如何使用如Socket.IO的跨浏览器的实时框架构建应用前，你会先学习 HTML5 WebSocket的缺陷。

本书包含大量示例。这些示例，会教你如何一步一步构建小应用或者测试不同的API。本书所有的示例代码都可以通过`node`命令运行，以下是两种不同的使用方式：

- 通过node REPL（Read-Eval-Print Loop）。和Firebug或者Web调试器中的JavaScript控制台类似，node REPL允许你从操作系统的命令行工具输入JavaScript代码，按下回车键，就能执行。
- 通过`node`命令运行node文件。这种方式要求你使用已有的文本编辑器。我个人推荐vim（http://vim.org）编辑器，不过，任何文本编辑器都是可以的。

绝大多数例子，会一步步教你书写示例代码，并且，首次书写会讲解其代码含义。我还会带领你经历不同的考验以及代码重构。当到了重要的里程碑时，我通常会展示一个截图，截图

内容取决于开发的应用，可能是终端的截图，也可能是浏览器端窗口的截图。

有的时候，讲解这些示例的时候，不管考虑得多周全，问题可能还是无法避免。所以，我给你提供了一个资源列表来帮助你解决问题。

资源

要是在阅读本书中，遇到问题，可以通过如下途径获得帮助。

要获得关于Node.js的问题的帮助，可以通过如下途径：

- Node.js邮件列表（http://groups.google.com/group/nodejs）。
- `irc.freenode.net`服务器，`#nodejs`频道。

要获得如`socket.io`或者`express`等的特定项目的帮助，可以通过官方支持频道；如果没有，可以通过像Stack Overflow（http:// stackoverflow.com/questions/tagged/node.js）这样的论坛，都会很有帮助。

绝大多数的Node.js模块都托管在GitHub上。如果你发现了bug，就可以通过GitHub报给他们，并贡献相应的测试用例。

尽力弄清楚你的问题到底属于Node.js还是JavaScript。这对确保你寻求的Node.js帮助确实是与Node相关的问题很有帮助。

如果就本书中的某个问题想要讨论，可以直接通过`rauchg@gmail.com`联系我。

目录

PART Ⅱ Node重要的API

PART Ⅳ 数据库

PART V 测试