

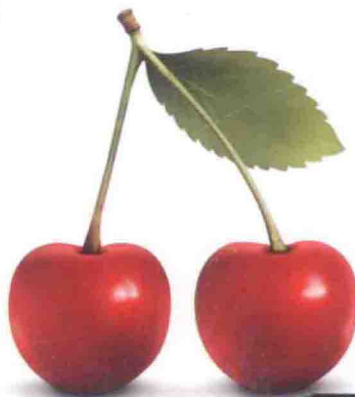
Node.js 开发实战详解

黄丹华 等编著

**腾讯Web前端工程师全面揭秘原生Node.js的开发实践
不借助任何第三方框架，通过编写原生代码，讲解Node.js应用开发**

- ◎ 深入详解Node.js原生文档，根据原生API实践和大量应用实例，详细分析Node.js的开发过程，了解原生Node.js的API应用
- ◎ 全面涵盖Node.js基础知识、模块与NPM、Web应用、UDP服务、异步编程思想、异常处理过程、操作数据库的方法、框架开发与应用、开发工具等

重实践，讲解时穿插了430多个代码小示例，提供了30多个编程实践练习题及答案，还介绍了5个大型系统的开发，并赠送8小时教学视频（需下载）



Web开发典藏大系

web design

server admin

consulting

marketing

mobile apps

domains

hosting

outsourcing



COMPANY INFORMATION
lorem ipsum dolor sit amet

SERVICES & SOLUTIONS
lorem ipsum dolor sit amet

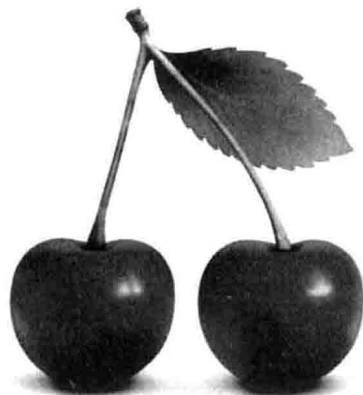
DAILY NEWSLETTER
lorem ipsum dolor sit amet

WORLDWIDE PARTNERS
lorem ipsum dolor sit amet

CUSTOMER SUPPORT
lorem ipsum dolor sit amet

Node.js 开发实战详解

黄丹华 等编著



内 容 简 介

本书由浅入深,全面、系统地介绍了 Node.js 开发技术。书中提供了大量有针对性的实例,供读者实践学习,同时提供了大量的实践练习题及详尽的解答,帮助读者进一步巩固和提高。本书重在代码实践,阅读时应多注重实践编程。本书提供 8 小时配套教学视频及实例源代码,便于读者高效、直观地学习。

本书共分为 11 章。涵盖的主要内容有: Node.js 的概念、应用场景、环境搭建和配置、异步编程; Node.js 的模块概念及应用、Node.js 的设计模式; HTTP 简单服务的搭建、Node.js 静态资源管理、文件处理、Cookie 和 Session 实践、Crypto 模块加密、Node.js 与 Nginx 配合; UDP 服务器的搭建、Node.js 与 PHP 之间合作; Node.js 的实现机制、Node.js 的原生扩展与应用; Node.js 的编码习惯; Node.js 操作 MySQL 和 MongoDB; 基于 Node.js 的 Myweb 框架的基本设计架构及实现; 利用 Myweb 框架实现一个简单的 Web 聊天室; 在线聊天室案例和在线中国象棋案例的实现; Node.js 的日志模块、curl 模块、crontab 模块、forever 模块、xml 模块和邮件发送模块等应用工具。

本书非常适合从事编程开发的学生、教师、广大科研人员和工程技术人员研读。建议阅读本书的读者对 JavaScript 的语法和 PHP 的相关知识有一定的了解。当然,如果你是初学者,本书也是一本难得的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Node.js 开发实战详解 / 黄丹华等编著. —北京:清华大学出版社, 2014

(Web 开发典藏大系)

ISBN 978-7-302-34947-1

I. ①N… II. ①黄… III. ①JAVASCRIPT 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 8213 号



责任编辑:夏兆彦

封面设计:欧振旭

责任校对:胡伟民

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印刷者:北京鑫丰华彩印有限公司

装订者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.5 字 数:615 千字

版 次:2014 年 4 月第 1 版 印 次:2014 年 4 月第 1 次印刷

印 数:1~4000

定 价:59.80 元

前 言

Node.js 是一个 JavaScript 运行环境 (runtime)。实际上它是对 GoogleV8 引擎 (应用于 Google Chrome 浏览器) 进行了封装。由于其拥有异步非阻塞、环境搭建简单、实践应用快等特性, 使得其在新一代编程开发中更为流行。同时, 由于 Node.js 基于 JavaScript 语法, 因此在学习 Node.js 时也可以了解和学习 JavaScript 的语法, 拓宽和加深读者对 Web 前端开发技术的理解。

当前, Node.js 主要应用于 HTTP Web 服务器的搭建和快速实现的独立服务器应用。在实践项目中, Node.js 更适合做一些小型系统服务或者一些大项目的部分功能的实现。由于其版本不稳定, 很多公司中主要将其应用于一些小项目中。如果以后其版本能够更加稳定可控, 相信会有更多的公司将其应用于各种项目和服务中。

目前, 国内 Node.js 的相关技术图书还非常稀缺。为了给想要学习 Node.js 开发技术的人员一个必要的指导, 笔者编写了这本书。本书既注重基础知识讲解, 又非常注重编程实践练习, 讲解时给出了有针对性的实例, 各章还给出了一些编程实践练习题。相信读者通过阅读本书, 不仅可以全面掌握 Node.js 开发技术, 还可以不需要借助任何框架而独立运用 Node.js 实现 HTTP Web 服务器的相关功能, 从而摒弃对框架的依赖, 进一步加深读者自我研发及独立思考的能力。

本书的特点

1. 编码不依赖任何框架

为了便于读者更好地了解原生 Node.js 的开发实践, 本书没有借助任何其他框架来讲解 Web 实践应用, 书中的所有模块都是通过编写原生代码来实现。

2. 结构合理, 内容全面、系统

本书全面、系统地介绍了 Node.js 从入门到编程实践的各种技术, 涵盖 Node.js 网络编程、Node.js 与数据库等方方面面的知识。

3. 叙述详实, 例程丰富

本书提供了大量例程, 便于读者实践演练。书中的每个例子都经过精挑细选, 有很强的针对性。这些实例都给出了完整的代码和详细的代码注释。这些代码非常简洁和高效, 便于读者学习和调试。当然, 读者也可以直接重用这些代码来解决自己的问题。

4. 结合实际, 编程技巧贯穿其中

本书写作时特意给出了大量的实用编程技巧, 对这些编程技巧的灵活使用, 将会使读

者的开发事半功倍。

5. 语言通俗，图文并茂

本书以通俗易懂的语言讲解每一个技术点和实例，讲解时还穿插了大量效果图，并给出了程序的运行结果插图，便于读者更加直观地学习和理解。

6. 大量习题，详尽解答

本书提供了大量的编程实践练习题和详尽的解答，便于读者进一步巩固和加深所学的各个技术点，从而达到更好的学习效果。

7. 配多媒体教学视频

为了便于读者更加高效、直观地理解书中的技术点，作者为本书专门录制了 8 小时配套的多媒体教学视频。这些视频和本书源代码一起收录于配书光盘中。虽然视频录制的设备条件有限（主要靠耳麦），但依然相信这些视频可以给读者的学习提供有益的帮助。

本书内容体系

本书共分 11 章，各章的具体内容介绍如下。

第 1 章主要介绍了 Node.js 的概念、配置、第一个 Node.js 程序 Hello World 的实现及异步编程思想等。

第 2 章主要介绍了 Node.js 中的模块的概念，以及 Node.js 中 `exports` 和 `module.exports` 之间的联系和区别。Node.js 中的 NPM 模块包含 `request`、`socket.io`、`express`、`jade` 和 `forever` 模块。Node.js 中的设计模式包含单例、适配器和装饰模式。

第 3 章主要介绍了 Node.js 的 Web 开发技术。包含 HTTP 简单服务搭建、Node.js 静态资源服务器实现、文件处理、Cookie 和 Session 实践、Crypto 模块加密及 Node.js 与 Nginx 配合实践等。

第 4 章主要介绍了 Node.js 中 UDP 服务器的搭建实践及 Node.js 与 PHP 之间的合作方式。

第 5 章主要介绍了 Node.js 中 `require` 机制的实现、Node.js 的 C++ 扩展（同步和异步接口）编译实践方法。

第 6 章主要介绍了一些关于 Node.js 的编码习惯。

第 7 章主要介绍了利用 Node.js 操作 MySQL 和 MongoDB 的实例，其中包含实现 Node.js 的两个基类分别对应于 MySQL 和 MongoDB。另外，还介绍了 MySQL 和 MongoDB 环境的搭建，以及两个数据中 Node.js 的 NPM 模块。

第 8 章主要从框架开发的角度介绍了一个基于 Node.js 的 Myweb 框架的基本设计架构及其实现的功能，以及该框架的实现。其中用到了 `express` 模块和 `jade` 解析模板，可帮助读者进一步了解 Node.js 的 Web 应用开发和 `express` 框架的应用。

第 9 章主要从框架应用的角度介绍了如何使用框架做一个简单的项目开发，即利用第 8 章的 MyWeb 1.0 框架实现一个简单的 Web 聊天室 MyChat 应用。

第 10 章主要介绍了两个实例：在线聊天室和联网在线中国象棋。这两个应用都是用

本书中自我实践的代码框架 MyWeb 2.0 来实现的。

第 11 章主要介绍了 Node.js 的一些应用工具，包含日志模块、curl 模块、crontab 模块、forever 模块、xml 模块和邮件发送模块。

本书读者对象

- Node.js 初学者；
- PHP 或者 JavaScript 程序员；
- 想全面、系统地学习 Node.js 的人员；
- Node.js 技术爱好者；
- 利用 Node.js 进行开发的技术人员；
- 大中专院校的学生和老师；
- 相关培训学校的学员。

本书作者

本书由黄丹华主笔编写。其他参与编写和资料整理的人员有陈杰、陈贞、樊俊、高彩丽、高莹婷、管磊、郭丽、韩亚、李红、李龙海、梁伟、刘忆智、曲宝军、孙忠贤、唐正兵、王全政、王勇浩、武文琛、徐学英、闫伍平、于轶、占海明、张帆。

致谢

本书在写作过程中参阅了大量的相关资料。在此对原文的作者、相关网站及社区表示特别的感谢！没有这些资料，笔者完成本书将会需要花费更多的时间，本书的推出时间也会延迟。下面给出本书参考的主要资料及来源。

CSDN 社区中的《程序员如何说服老板采用 Node.js》：由于 Node.js 已经越来越多地被程序员和公司关注，基于此 CSDN 有针对性地写了这篇文章，系统地告诉程序员在适当的机会下从哪些方面入手才能让团队及老板来支持 Node.js 的项目实现。本书中多处参考了该文章。

HACK SPARROW 的 *Node.js Module - exports vs module.exports* 和 *Create NPM Package - Node.js Module*：本书中介绍的 `exports` 与 `module.exports` 之间的区别和联系参考了英文资料 *Node.js Module - exports vs module.exports*；本书中介绍的 Node.js NPM 模块发布参考了文章 *Create NPM Package - Node.js Module*。

CNode 社区 ctrlacv 的《静态文件服务器代码整理》：本书实现的一个静态服务器参考了国内知名 Node.js 社区 CNode 中的 ctrlacv 文章《静态文件服务器代码整理》。

田永强编著的《深入浅出 Node.js(三)：深入 Node.js 的模块机制》：本书在深入 Node.js 中介绍的 `require` 机制实现则是参阅了田永强的文章《深入浅出 Node.js(三)：深入 Node.js 的模块机制》编写而成。

移动开发博客 lishen 的《编写 Node.js 原生扩展》：本书中介绍的关于实现 Node.js 原生扩展模块方法，主要参考了国内网站移动开发博客 lishen 的文章《编写 Node.js 原生扩展》。

笔者在本书中给出了大量的脚注，注明这些资料的来源。其目的—是表示对原作者的尊重和感谢；二是便于读者查阅和学习。

本书的编写对笔者而言是一个“浩大的工程”。虽然笔者投入了大量的精力和时间，但只怕百密难免一疏。若有任何疑问或疏漏，请发邮件至 bookservice2008@163.com。最后祝读者读书快乐！

编著者

目 录

第 1 章 Node.js 基础知识	1
1.1 概述	1
1.1.1 Node.js 是什么	1
1.1.2 Node.js 带来了什么	1
1.2 Node.js 配置开发	3
1.2.1 Windows 配置	3
1.2.2 Linux 配置	5
1.2.3 Hello World	6
1.2.4 常见问题	7
1.3 异步编程	8
1.3.1 同步调用和异步调用	8
1.3.2 回调和异步调用	11
1.3.3 获取异步函数的执行结果	12
1.4 本章实践	12
1.5 本章小结	14
第 2 章 模块和 NPM	16
2.1 什么是模块	16
2.1.1 模块的概念	16
2.1.2 Node.js 如何处理模块	16
2.1.3 Node.js 实现 Web 解析 DNS	18
2.1.4 Node.js 重构 DNS 解析网站	24
2.1.5 exports 和 module.exports	28
2.2 NPM 简介	30
2.2.1 NPM 和配置	30
2.2.2 Express 框架	31
2.2.3 jade 模板	33
2.2.4 forever 模块	36
2.2.5 socket.io 模块	38
2.2.6 request 模块	40
2.2.7 Formidable 模块	43
2.2.8 NPM 模块开发指南	45
2.3 Node.js 设计模式	47

2.3.1	模块与类	47
2.3.2	Node.js 中的继承	49
2.3.3	单例模式	55
2.3.4	适配器模式	57
2.3.5	装饰模式	59
2.3.6	工厂模式	61
2.4	本章实践	63
2.5	本章小结	75
第 3 章	Node.js 的 Web 应用	77
3.1	HTTP 服务器	77
3.1.1	简单的 HTTP 服务器	77
3.1.2	路由处理	81
3.1.3	GET 和 POST	84
3.1.4	GET 方法实例	84
3.1.5	POST 方法实例	87
3.1.6	HTTP 和 HTTPS 模块介绍	90
3.2	Node.js 静态资源管理	91
3.2.1	为什么需要静态资源管理	92
3.2.2	Node.js 实现简单静态资源管理	93
3.2.3	静态资源库设计	96
3.2.4	静态文件的缓存控制	99
3.3	文件处理	104
3.3.1	File System 模块介绍	104
3.3.2	图片和文件上传	108
3.3.3	jade 模板实现图片上传展示功能	112
3.3.4	上传图片存在的问题	116
3.3.5	文件读写	117
3.4	Cookie 和 Session	122
3.4.1	Cookie 和 Session	122
3.4.2	Session 模块实现	123
3.4.3	Session 模块的应用	126
3.5	Crypto 模块加密	127
3.5.1	Crypto 介绍	127
3.5.2	Web 数据密码的安全	131
3.5.3	简单加密模块设计	132
3.6	Node.js+Nginx	136
3.6.1	Nginx 概述	137
3.6.2	Nginx 的配置安装	137
3.6.3	如何构建	142

3.7	文字直播实例	145
3.7.1	系统分析	145
3.7.2	重要模块介绍	147
3.8	扩展阅读	155
3.9	本章实践	159
3.10	本章小结	173
第4章	Node.js 高级编程	175
4.1	构建 UDP 服务器	175
4.1.1	UDP 模块概述	175
4.1.2	UDP Server 构建	176
4.2	UDP 服务器应用	179
4.2.1	应用分析介绍	180
4.2.2	UDP Server 端（图片处理服务器）实现	181
4.2.3	UDP Client 端（Web Server）	184
4.2.4	Jade 页面实现	186
4.2.5	应用体验	187
4.3	Node.js 与 PHP 合作	189
4.3.1	UDP 方式	189
4.3.2	脚本执行	191
4.3.3	HTTP 方式	191
4.3.4	三种方式的比较	192
4.4	本章实践	193
4.5	本章小结	196
第5章	深入 Node.js	199
5.1	Node.js 的相关实现机制	199
5.2	Node.js 原生扩展	202
5.2.1	Node.js 扩展开发基础 V8	202
5.2.2	Node.js 插件开发介绍	204
5.3	Node.js 异步扩展开发与应用	205
5.4	本章实践	212
5.5	本章小结	214
第6章	Node.js 编码习惯	216
6.1	Node.js 规范	216
6.1.1	变量和函数命名规范	216
6.1.2	模块编写规范	219
6.1.3	注释	220
6.2	Node.js 异步编程规范	221
6.2.1	Node.js 的异步实现	221

6.2.2	异步函数的调用	224
6.2.3	Node.js 异步回调深度	226
6.2.4	解决异步编程带来的麻烦	227
6.3	异常逻辑的处理	231
6.3.1	require 模块对象不存在异常	231
6.3.2	对象中不存在方法或者属性时的异常	233
6.3.3	异步执行的 for 循环异常	234
6.3.4	利用异常处理办法优化路由	236
6.3.5	异常情况汇总	240
6.4	本章实践	241
6.5	本章小结	241
第 7 章	Node.js 与数据库	243
7.1	两种数据库介绍	243
7.1.1	MySQL 介绍	243
7.1.2	MongoDB 模块介绍	247
7.2	Node.js 与 MySQL	250
7.2.1	MySQL 安装配置应用	250
7.2.2	MySQL 数据库接口设计	251
7.2.3	数据库连接	252
7.2.4	数据库插入数据	254
7.2.5	查询一条数据记录	256
7.2.6	修改数据库记录	258
7.2.7	删除数据库记录	259
7.2.8	数据条件查询	260
7.3	Node.js 与 MongoDB	262
7.3.1	MongoDB 的安装以及工具介绍	263
7.3.2	MongOD 的启动运行方法	264
7.3.3	MongoDB 的启动运行	266
7.3.4	MongoDB 数据库接口设计	268
7.3.5	数据插入	272
7.3.6	数据修改	274
7.3.7	查询一条数据	276
7.3.8	删除数据	278
7.3.9	查询数据	279
7.4	MySQL 与 MongoDB 性能	281
7.4.1	测试工具及测试逻辑	282
7.4.2	MySQL 性能测试代码	282
7.4.3	MongoDB 性能测试代码	283
7.4.4	性能测试数据分析	283

7.5	本章实践	285
7.6	本章小结	289
第 8 章	MyWeb 框架介绍	290
8.1	MyWeb 框架介绍	290
8.1.1	MyWeb 框架涉及的应用	290
8.1.2	MyWeb 框架应用模块	291
8.2	MyWeb 源码架构	292
8.2.1	框架 MVC 设计图	292
8.2.2	框架文件结构	293
8.2.3	扩展阅读之更快地了解新项目	294
8.3	框架源码分析	295
8.3.1	框架入口文件模块	295
8.3.2	路由处理模块	297
8.3.3	Model 层基类	299
8.3.4	Controller 层基类	301
8.4	本章实践	302
8.5	本章小结	302
第 9 章	框架应用 MyChat	304
9.1	编码前的准备	304
9.1.1	应用分析	305
9.1.2	应用模块	305
9.1.3	功能模块设计	307
9.2	系统的编码开发	309
9.2.1	Model 层	309
9.2.2	Controller 层	311
9.2.3	View 层	316
9.3	项目总结	318
9.3.1	forever 启动运行项目	318
9.3.2	系统应用体验	320
9.3.3	系统开发总结	323
9.4	扩展阅读之 MyWeb 2.0 的介绍	323
9.5	本章实践	325
9.6	本章小结	325
第 10 章	Node.js 实例应用	326
10.1	实时聊天对话	326
10.1.1	系统设计	326
10.1.2	系统的模块设计	327
10.1.3	系统编码实现	328

10.2	联网中国象棋游戏	332
10.2.1	系统设计	333
10.2.2	系统的模块设计	334
10.2.3	系统编码实现	334
10.2.4	系统体验	337
10.3	本章小结	339
第 11 章	Node.js 实用工具	340
11.1	日志模块工具	340
11.1.1	日志模块介绍	340
11.1.2	日志模块实现	341
11.1.3	日志模块应用	345
11.2	配置文件读取模块	347
11.2.1	配置文件解析模块介绍	347
11.2.2	配置文件解析模块实现	348
11.3	curl 模块	352
11.3.1	curl 模块介绍	352
11.3.2	curl 模块实现	353
11.3.3	curl 模块应用	356
11.4	crontab 模块	357
11.4.1	crontab 模块介绍	358
11.4.2	crontab 模块设计实现	358
11.4.3	crontab 模块应用	361
11.5	forever 运行脚本	362
11.5.1	forever 运行脚本介绍	362
11.5.2	forever 运行脚本实现	363
11.5.3	forever 运行脚本应用	366
11.6	xml 模块的应用	367
11.6.1	xml 解析模块介绍	368
11.6.2	xml 模块设计实现	369
11.6.3	xml 模块应用	371
11.7	邮件发送模块应用	374
11.7.1	邮件模块介绍	374
11.7.2	邮件模块设计实现	374
11.7.3	邮件模块应用	376
11.8	本章小结	377

第 1 章 Node.js 基础知识

本章主要介绍 Node.js 的概念，以及 Node.js 的出现给我们程序员带来了什么方便、解决了什么问题、在项目开发中是否应该选择使用 Node.js。同时介绍 Node.js 的配置、开发，以及 Node.js 的入门程序 Hello world。最后介绍 Node.js 的编程思想、如何得到一个异步结果、区分同步和异步的编程思想的不同之处。

1.1 概 述

本节主要介绍 Node.js 基本概念，包括什么是 Node.js、它与其他编程语言之间的区别、它的特点和优势在哪里，以及项目开发过程中如何抉择是否使用 Node.js 开发应用。

1.1.1 Node.js 是什么

Node.js 是一个 JavaScript 运行环境 (runtime)。实际上它是对 Google V8 引擎 (应用于 Google Chrome 浏览器) 进行了封装。Node.js 对一些特殊用例进行了优化，提供了替代的 API，使得 V8 在非浏览器环境下运行得更好。其目的是可以在服务器端执行和运行 JavaScript 代码。长久以来 JavaScript 都是一个基于浏览器的客户端脚本语言，通过将其运行环境抽离出来，就可以在服务器端运行 JavaScript 代码，而并非仅仅依赖浏览器解析，从而就可以将其作为服务器端语言，由于其拥有异步非阻塞特性，因此其在长连接、多请求的环境下优势非常明显。

Node.js 的编程语言是基于 JavaScript 的语法，因此想深入学习 Node.js 的入门者，可以先熟悉 JavaScript 编程语言，了解其基本的语法 (本书不会介绍 JavaScript 的语法)，同时要对服务器端开发有所了解。当然，如果你是初学者也可以通过本书学习 Node.js 的编程，但需要你对知识了解更深入一些。Node.js 提供了一些特殊的 API (Node.js 官网提供了详细的 API 说明) 库，因此在编写 Node.js 的时候可以理解为，使用 JavaScript 语言，利用 Node.js 的 API 库进行服务器端开发。

1.1.2 Node.js 带来了什么

初学者经常会提及一个问题——Node.js 到底有什么用？这个问题也经常令我苦恼，是不是有了 Node.js 我们就可以抛弃 PHP 或者其他服务器语言了呢？答案是否定的。

首先要理解 Node.js 的优点，以及 Node.js 与其他语言之间的区别和联系。传统的服务器语言，如 PHP 和 Java 等，每个 Web HTTP 请求连接都会产生一个线程，假设每个线程

需要 2MB 的配置内存，因此相对一个 8GB 的服务器主机，也只能承受来自 4000 个并发用户的请求，当服务器承受不了这么多用户的情况下就需要添加服务器，从而导致增加项目运营成本（当然现在有 Nginx 支撑，可以提供更高的并发量请求）。

其次，理解 Node.js 解决的问题是什么？Node.js 解决多请求的方法，在于其处理连接服务器的方式。在 Node.js 中每个 HTTP 连接都会发射一个在 Node.js 引擎的进程中运行的事件，而不是为每个连接生成一个新的 OS 进程（并为其分配一些配套内容）。

综上所述就可以清晰地看出 Node.js 相对 PHP 来说其优点在于能处理高并发请求，并且由于 Node.js 是事件驱动，因此可以更好地节约服务器内存资源。

在项目的开发中，我们应该如何抉择是否使用 Node.js 作为项目开发实现呢？首先，必须要了解项目的类型是不是适合 Node.js 去开发项目。例如需要开发一个博客、论坛或者微博，那么是否能使用 Node.js 去开发？回答是肯定的，但是不合适，相对来说 PHP 在这方面已经很成熟。其次，要理解一个事实，在本书编写过程中 Node.js 版本还不是很稳定，在 Node.js 版本升级的过程中，可能引发一些项目代码中的问题，例如 Node.js 官方提供的 API 有所改动时，以及当项目 Node.js 版本升级时，则必须修改代码。

Node.js 可以单独实现一个 server，这也是 Node.js 一个非常大的优点，对于那些简单的 server，通过 Node.js 实现比使用 C++ 实现会简单得多。最后，牢记 Node.js 解决了长连接、多请求引发的成本问题，因此在一些项目，例如实时在线 Game（如一起来画画、黑暗杀人游戏、实时休闲游戏等）、实时在线聊天室、实时消息推送功能、SNS 实时交流、实时监控系统（如股票、系统运行状态等）等开发过程中，应该把握住机会，应用 Node.js 来开发。那么在如此好的机会面前，应该如何去说服团队和老板呢？

在 CSDN 社区中有一篇博文《程序员如何说服老板采用 Node.js》¹，详细说明了一个项目是否适合用 Node.js 开发，当适合时应该如何用事实说服老板、团队和客户。总结其过程包括如下几个阶段，如图 1-1 所示。

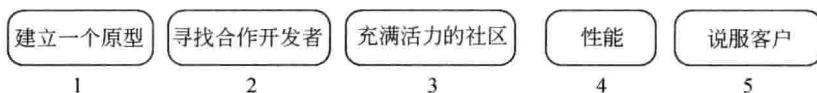


图 1-1 说服过程

- ❑ 建立原型：对于老板来说他希望看到的是结果，不管你使用什么开发，他的最终目的是节省资源、快速开发、满足客户。因此首先需要给老板看到一个系统原型，能够让老板明白你的系统开发是有条理的，不是简简单单的一个想法。
- ❑ 寻找合作开发者：项目开发是一个团队的事情，我们依靠的是团队力量，因此想成功开发一个 Node.js 应用，必须要说服团队的成员和你共同开发。有团队的依靠才能更进一步地让老板放心。
- ❑ 充满活力的社区：此刻，Node.js 社区用户数正以一个疯狂的速度增长，该社区吸引了众多顶尖开发者。也就是说，Node.js 生态系统每天都在完善，并且通过不同渠道获得了各个企业的免费支持。你需要告诉老板的是，现在这门新技术正在蓬勃的发展，并且也会是未来 Web 应用的一门比较重要的语言。


1 CSDN 社区博文网站 <http://www.csdn.net/article/2012-05-03/2805296>。

- 性能：在合适的项目中，更要突出 Node.js 开发项目的优点，以及可行性。可以在一个项目的会议上，通过你的讲解更进一步地说服老板，让老板认为你的想法是正确并且可靠的。
- 说服客户：最后你需要做的就是说服客户，需要让他们知道的是，Node.js 的开发更能够满足他们的需求，能够更快地开发出他们需要的产品。

把握以上几点以后，还需要我们加强个人的 Node.js 知识，多了解一些服务器端的知识扩展自己的知识层面。可以说 Node.js 给我们带来的是一个选择，在项目开发中遇到问题时，我们可以考虑使用 Node.js 来解决问题。对于程序员来说，给我们带来的是一个机遇和挑战，能够让我们成为小组 Node.js 项目开发中不可或缺的成员。

1.2 Node.js 配置开发

本节将会介绍如何搭建 Node.js 的运行环境，包括 Linux 系统和 Windows 系统环境，同时学习一门语言的第一个程序“Hello World”。学完本节，需要掌握 Linux 系统和 Windows 系统下的 Node.js 环境搭建，以及编写简单的 Node.js 程序、运行和调式方法。

 **注意：**在学习本节时，需要注意在本书编写过程中安装配置可能会和读者阅读本书时的版本安装存在差异，这些知识笔者后续会通过论坛告知大家，但是大致的安装和配置过程是类似的。

1.2.1 Windows 配置

Node.js 的 Windows 运行环境安装步骤简单，只需要注意将执行文件 `node.exe` 路径添加到环境变量。安装主要分为 4 个步骤，其中安装步骤 1 中建议下载 `msi`¹ 的后缀执行文件，主要是便于后续的项目开发。安装步骤 4 中介绍如何将 `node.exe` 添加到全局执行环境中，本步骤没有介绍如何在 Window 中添加环境变量²，这些基本知识在这里就不详细介绍了。主要安装过程如下所述。

(1) 官网 (<http://nodejs.org>) 下载 Node.js 的 Windows 系统 (32 位和 64 位) 最新版本，建议下载 `msi` 为后缀的版本，如图 1-2 中方框内所示的版本。`exe` 为执行文件，但其缺少一些 Node.js 的模块和 `npm` 等。

(2) 下载完成后，执行 MSI 的安装文件。

(3) 安装完成，查看 Node.js 启动文件目录 (右键单击 Node.js 启动快捷方式)，如图 1-3 所示，启动文件目录一般默认情况下是在“`C:\Program Files\nodejs\node.exe`”。

(4) 将 `node.exe` 可执行文件路径添加到 Windows 的环境变量中；运行 `cmd`，进入 `dos`

1 MSI 文件是 Windows Installer 的数据包，它实际上是一个数据库，包含安装一种产品所需要的信息和在很多安装情形下安装 (和卸载) 程序所需的指令和数据。

2 环境变量设置方法参考：<http://www.cnblogs.com/307914070/archive/2011/02/11/1951725.html>。

操作命令窗口，输入 `node -v` 查看是否安装成功，如图 1-4 所示，表明已经成功安装 Node.js。



图 1-2 Node.js 官网下载页面

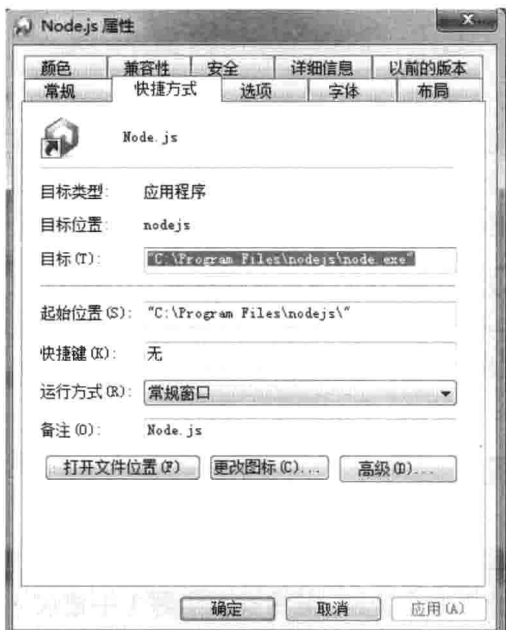


图 1-3 Node.js 右键快捷图标属性

注意：Windows 添加环境变量的方法：右键计算机桌面图片，打开右侧导航栏中的“高级系统设置”，打开以后选择其中的“环境变量”，之后再更改用户的 PATH 值，在其原有变量值上加上 Node.js 可执行文件路径，两者使用分号分割，例如：
`C:\Users\danhuang\AppData\Roaming\npm\;C:\Program Files\nodejs\node.exe`

```
C:\Users\danhuang>node --version
v0.8.21
```

图 1-4 Node.js 查看版本信息

成功安装后可以看到当前的 Node.js 的版本信息，如图 1-4 所示。如果没有成功添加到系统的环境变量中，cmd 运行窗口会提示没有 `node` 指令，这时候应该检查是否已经将 Node.js 的可执行路径成功添加到环境变量中。