

Node.js+MongoDB+AngularJS Web开发



[Node.js+MongoDB+AngularJS Web开发_下载链接1](#)

著者:[美] Brad Dayley

出版者:电子工业出版社

出版时间:2015-6

装帧:平装

isbn:9787121261176

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。

《Node.js+MongoDB+AngularJS Web开发》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。

《Node.js+MongoDB+AngularJS Web开发》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。

作者介绍:

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。他曾设计并实现了一大批网站，其范围涵盖从企业应用服务器到富互联网应用程序，以及用于移动Web服务的智能界面。他的著作包括jQuery and JavaScript Phrasebook、Sams Teach Yourself NoSQL with MongoDB in 24 Hours和 Sams Teach Yourself jQuery and JavaScript in 24 Hours。

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	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
第8章 在Node.js中实现套接字服务	131
8.1 了解网络套接字	131
8.2 了解TCP服务器和Socket对象	132
8.2.1 net.Socket对象	132
8.2.2 net.Server对象	135
8.3 实现TCP套接字服务器和客户端	137
8.3.1 实现TCP套接字客户端	138
8.3.2 实现TCP套接字服务器	140
8.4 实现TLS服务器和客户端	143
8.4.1 创建TLS套接字客户端	144
8.4.2 创建TLS套接字服务器	145
8.5 小结	147
8.6 下一章	147
第9章 在Node.js中使用多处理器扩展应用程序	149
9.1 了解process模块	149
9.1.1 了解进程I/O管道	149
9.1.2 了解进程的信号	150
9.1.3 使用process模块控制进程执行	150
9.1.4 从process模块获取信息	151
9.2 实现子进程	153
9.2.1 了解ChildProcess对象	153
9.2.2 使用exec()在另一个进程上执行一个系统命令	154
9.2.3 使用execFile()在另一个进程上执行一个可执行文件	156
9.2.4 使用spawn()在另一个Node.js实例中产生一个进程	157
9.2.5 实现子派生	159
9.3 实现进程集群	161
9.3.1 使用cluster模块	162
9.3.2 了解Worker对象	163

9.3.3 实现一个HTTP集群	164
9.4 小结	166
9.5 下一章	166
第10章 使用其他Node.js模块	167
10.1 使用os模块	167
10.2 使用util模块	169
10.2.1 格式化字符串	169
10.2.2 检查对象类型	170
10.2.3 同步写入输出流	170
10.2.4 将JavaScript对象转换为字符串	170
10.2.5 从其他对象继承功能	171
10.2.6 使用dns模块	172
10.3 小结	174
10.4 下一章	174
第3部分 学习MongoDB	
第11章 了解NoSQL和MongoDB	177
11.1 为什么要采用NoSQL	177
11.2 了解MongoDB	178
11.2.1 理解集合	178
11.2.2 了解文档	178
11.3 MongoDB的数据类型	179
11.4 规划你的数据模型	180
11.4.1 使用文档引用来规范化数据	181
11.4.2 使用嵌入式文档反规范化数据	182
11.4.3 使用封顶集合	183
11.4.4 了解原子写操作	184
11.4.5 考虑文件增长	184
11.4.6 识别索引、分片和复制的机会	184
11.4.7 大集合与大量集合的对比	185
11.4.8 决定数据生命周期	185
11.4.9 考虑数据的可用性和性能	185
11.5 小结	186
11.6 下一章	186
第12章 MongoDB入门	187
12.1 构建MongoDB的环境	187
12.1.1 MongoDB的安装	187
12.1.2 启动MongoDB	188
12.1.3 停止MongoDB	189
12.1.4 从shell客户端访问MongoDB	189
12.2 管理用户账户	192
12.2.1 列出用户	192
12.2.2 创建用户账户	193
12.2.3 删除用户	194
12.3 配置访问控制	194
12.3.1 创建用户管理员账户	195
12.3.2 打开身份验证	196
12.3.3 创建数据库管理员账户	196
12.4 管理数据库	197
12.4.1 显示数据库清单	197
12.4.2 切换当前数据库	197
12.4.3 创建数据库	197
12.4.4 删除数据库	198
12.4.5 复制数据库	198
12.5 管理集合	199

- 12.5.1 显示数据库中的集合列表 199
- 12.5.2 创建集合 199
- 12.5.3 删除集合 200
- 12.5.4 在集合中查找文档 201
- 12.5.5 将文档添加到集合中 201
- 12.5.6 从集合中删除文档 202
- 12.5.7 更新集合中的文档 202
- 12.6 小结 204
- 12.7 下一章 204
- 第13章 MongoDB和Node.js入门 205
- 13.1 把MongoDB的驱动程序添加到Node.js 205
- 13.2 从Node.js连接到MongoDB 206
 - 13.2.1 了解写入关注 206
 - 13.2.2 了解Server对象 206
 - 13.2.3 通过Client对象连接到MongoDB 207
 - 13.2.4 通过一个连接字符串连接到MongoDB 210
- 13.3 了解用在MongoDB Node.js驱动程序中的对象 211
 - 13.3.1 了解Db对象 212
 - 13.3.2 了解Admin对象 213
 - 13.3.3 了解Collection对象 214
 - 13.3.4 了解Cursor对象 215
- 13.4 访问和操作数据库 216
 - 13.4.1 列出数据库 217
 - 13.4.2 创建数据库 217
 - 13.4.3 删除数据库 217
 - 13.4.4 创建、列出和删除数据库实例 218
 - 13.4.5 获取MongoDB服务器的状态 219
- 13.5 访问和操作集合 220
 - 13.5.1 列出集合 220
 - 13.5.2 创建集合 221
 - 13.5.3 删除集合 221
 - 13.5.4 创建、列出和删除集合的示例 221
 - 13.5.5 获取集合信息 222
- 13.6 小结 223
- 13.7 下一章 223
- 第14章 从Node.js操作MongoDB文档 225
- 14.1 了解数据库更改选项 225
- 14.2 了解数据库更新运算符 226
- 14.3 将文档添加到集合 227
- 14.4 从集合获取文档 229
- 14.5 更新集合中的文档 230
- 14.6 原子地修改文档的集合 232
- 14.7 保存集合中的文档 234
- 14.8 使用upsert往集合中插入文档 235
- 14.9 从集合中删除文档 236
- 14.10 从集合中删除单个文档 238
- 14.11 小结 240
- 14.12 下一章 240
- 第15章 从Node.js访问MongoDB的文档 241
- 15.1 介绍数据集 241
- 15.2 了解query对象 242
- 15.3 了解查询options对象 243
- 15.4 查找特定文档集合 244
- 15.5 清点文档数量 247

- 15.6 对结果集进行限制 249
 - 15.6.1 按大小限制结果 249
 - 15.6.2 限制对象返回的字段 250
 - 15.6.3 对结果进行分页 252
- 15.7 对结果集进行排序 253
- 15.8 查找不同的字段值 255
- 15.9 对结果进行分组 256
- 15.10 通过聚合结果来应用MapReduce 259
 - 15.10.1 了解aggregate()方法 259
 - 15.10.2 使用聚合框架运算符 260
 - 15.10.3 实现聚合表达式运算符 261
 - 15.10.4 聚合的例子 262
- 15.11 小结 263
- 15.12 下一章 264
- 第16章 利用Mongoose来使用结构化模式与验证 265
 - 16.1 了解Mongoose 265
 - 16.2 利用Mongoose连接到MongoDB数据库 266
 - 16.3 定义模式 267
 - 16.3.1 了解路径 268
 - 16.3.2 创建一个模式定义 268
 - 16.3.3 把索引添加到一个模式 269
 - 16.3.4 实现字段的唯一性 269
 - 16.3.5 强制字段的必需性 270
 - 16.3.6 添加Schema对象的方法 270
 - 16.3.7 在words数据库上实现一个模式 270
 - 16.4 编译模型 271
 - 16.5 了解Query对象 272
 - 16.5.1 设置查询数据库操作 273
 - 16.5.2 设置查询数据库操作选项 274
 - 16.5.3 设置查询运算符 275
 - 16.6 了解Document对象 276
 - 16.7 利用Mongoose查找文档 278
 - 16.8 利用Mongoose添加文档 280
 - 16.9 利用Mongoose更新文档 281
 - 16.9.1 保存文档更改 282
 - 16.9.2 更新单个文档 283
 - 16.9.3 更新多个文档 284
 - 16.10 利用Mongoose删除文档 286
 - 16.10.1 删除单个文档 286
 - 16.10.2 删除多个文档 287
 - 16.11 利用Mongoose聚合文档 288
 - 16.12 使用验证框架 291
 - 16.13 实现中间件函数 293
 - 16.14 小结 295
 - 16.15 下一章 295
- 第17章 高级MongoDB概念 297
 - 17.1 添加索引 297
 - 17.2 使用封顶集合 299
 - 17.3 应用复制 300
 - 17.3.1 复制策略 302
 - 17.3.2 部署一个副本集 302
 - 17.4 实施分片 303
 - 17.4.1 分片服务器类型 304
 - 17.4.2 选择一个分片键 305

- 17.4.3 选择一种分区方法 306
- 17.4.4 部署一个分片的MongoDB集群 307
- 17.5 实现一个GridFS Store 310
 - 17.5.1 在Node.js实现Grid对象 311
 - 17.5.2 在Node.js实现GridStore对象 313
- 17.6 修复MongoDB数据库 315
- 17.7 备份MongoDB 316
- 17.8 小结 317
- 17.9 下一章 317
- 第4部分 使用Express使生活更轻松
- 第18章 在Node.js中实现Express 321
 - 18.1 Express入门 321
 - 18.1.1 配置Express设置 321
 - 18.1.2 启动Express服务器 322
 - 18.2 配置路由 323
 - 18.2.1 实现路由 323
 - 18.2.2 在路由中应用参数 324
 - 18.3 使用Request对象 328
 - 18.4 使用Response对象 330
 - 18.4.1 设置标头 330
 - 18.4.2 设置状态 331
 - 18.4.3 发送响应 331
 - 18.4.4 发送JSON响应 333
 - 18.4.5 发送文件 335
 - 18.4.6 发送下载响应 336
 - 18.4.7 重定向响应 336
 - 18.5 实现一个模板引擎 337
 - 18.5.1 定义一个模板引擎 338
 - 18.5.2 加入本地对象 338
 - 18.5.3 创建模板 339
 - 18.5.4 在响应中呈现模板 340
 - 18.6 小结 342
 - 18.7 下一章 342
- 第19章 实现Express中间件 343
 - 19.1 了解中间件 343
 - 19.1.1 在全局范围内把中间件分配给某个路径 344
 - 19.1.2 把中间件分配到单个路由 345
 - 19.1.3 添加多个中间件函数 345
 - 19.2 使用query中间件 345
 - 19.3 提供静态文件服务 346
 - 19.4 处理POST正文数据 348
 - 19.5 发送和接收cookie 349
 - 19.6 实现会话 350
 - 19.7 应用基本的HTTP身份验证 352
 - 19.8 实现会话身份验证 354
 - 19.9 创建自定义中间件 356
 - 19.10 小结 357
 - 19.11 下一章 357
- 第5部分 学习AngularJS
- 第20章 AngularJS入门 361
 - 20.1 为什么选择AngularJS 361
 - 20.2 了解AngularJS 362
 - 20.2.1 模块 362
 - 20.2.2 作用域和数据模型 363

20.2.3 具有模板和指令的视图	363
20.2.4 表达式	363
20.2.5 控制器	363
20.2.6 数据绑定	364
20.2.7 服务	364
20.2.8 依赖注入	364
20.2.9 职责分离	364
20.3 AngularJS生命周期概述	365
20.3.1 引导阶段	365
20.3.2 编译阶段	365
20.3.3 运行时数据绑定阶段	366
20.4 将AngularJS与现有的JavaScript和jQuery整合	366
20.5 将AngularJS添加到Node.js的环境	367
20.6 在HTML文档中引导AngularJS	367
20.7 使用全局API	368
20.8 创建基本的AngularJS应用程序	369
20.8.1 加载AngularJS库和你的主模块	370
20.8.2 定义AngularJS应用程序根元素	370
20.8.3 将控制器添加到模板	370
20.8.4 实现作用域模型	371
20.9 小结	372
20.10 下一章	372
第21章 了解AngularJS模块和依赖注入	373
21.1 模块和依赖注入概述	373
21.1.1 了解模块	373
21.1.2 依赖注入	374
21.2 定义AngularJS模块	374
21.2.1 创建AngularJS Module对象	374
21.2.2 使用配置块	375
21.2.3 使用运行块	375
21.2.4 把提供者添加到AngularJS模块	376
21.3 实现依赖注入	377
21.4 小结	379
21.5 下一章	380
第22章 把作用域实现为数据模型	381
22.1 了解作用域	381
22.1.1 根作用域和应用程序之间的关系	381
22.1.2 作用域和控制器之间的关系	382
22.1.3 作用域和模板之间的关系	383
22.1.4 作用域和后端服务器数据之间的关系	385
22.1.5 作用域的生命周期	385
22.2 实现作用域层次结构	387
22.3 发出和广播活动	389
22.4 小结	392
22.5 下一章	393
第23章 使用AngularJS模板来创建视图	395
23.1 了解模板	395
23.2 使用表达式	396
23.3 使用过滤器	399
23.3.1 使用内置过滤器	400
23.3.2 使用过滤器来实现排序和过滤	402
23.4 创建自定义过滤器	405
23.5 小结	406
23.6 下一章	407

第24章 在AngularJS视图中实现指令 409

24.1 了解指令 409

24.2 使用内置指令 409

24.2.1 支持AngularJS功能的指令 410

24.2.2 扩展表单元素的指令 411

24.2.3 把模型绑定到页面元素的指令 414

24.2.4 把页面事件绑定到控制器的指令 418

24.3 创建你自己的指令来扩展HTML 420

24.3.1 定义指令视图模板 421

24.3.2 限制指令行为 422

24.3.3 更换模板元素 423

24.3.4 转置外部作用域 423

24.3.5 配置指令的作用域 423

24.3.6 操纵DOM的链接功能 425

24.3.7 添加一个控制器到指令 426

24.3.8 创建自定义指令 426

24.4 小结 428

24.5 下一章 429

第25章 在Web应用程序中实现AngularJS服务 431

25.1 了解AngularJS服务 431

25.2 使用内置的服务 431

25.2.1 使用\$http服务发送HTTP的GET和PUT请求 432

25.2.2 使用\$cacheFactory服务 436

25.2.3 使用\$window服务实现浏览器警报 437

25.2.4 使用\$cookieStore服务与浏览器cookie交互 437

25.2.5 使用\$interval和\$timeout服务实现定时器 439

25.2.6 使用\$animate服务 440

25.3 创建自定义服务 445

25.3.1 创建value服务 445

25.3.2 创建constant服务 445

25.3.3 创建factory服务 445

25.3.4 创建service服务 446

25.3.5 把服务集成在一个模块中 446

25.4 小结 447

25.5 下一章 447

第6部分 建立实用的Web应用程序组件

第26章 将用户账户添加到你的网站 451

26.1 用到的库 451

26.2 项目的目录结构 451

26.3 定义用户模型 452

26.4 创建服务器 453

26.5 实现路由 454

26.6 实现用户控制器路由 456

26.6.1 实现用户注册路由 456

26.6.2 实现用户登录路由 457

26.6.3 实现获得用户配置文件的路由 457

26.6.4 实现更新用户路由 458

26.6.5 实现删除用户路由 458

26.6.6 完整的用户控制器代码 459

26.7 实现用户和身份验证视图 461

26.7.1 实现signup视图 461

26.7.2 实现index视图 462

26.7.3 实现login视图 463

26.7.4 实现user视图 464

- 26.7.5 添加CSS代码到样式视图 466
- 26.8 实现AngularJS模块和控制器 466
- 26.9 使用社交媒体账户作为身份验证来源 467
 - 26.9.1 添加Passport 467
 - 26.9.2 添加身份验证策略 467
 - 26.9.3 实现应用程序中间件 468
 - 26.9.4 添加已验证身份的用户序列化和反序列化 469
 - 26.9.5 实现路由以使用身份验证 469
 - 26.9.6 使用谷歌验证身份 470
- 26.10 小结 473
- 26.11 下一章 473
- 第27章 添加评论主题页 475
 - 27.1 用到的库 475
 - 27.2 项目的目录结构 475
 - 27.3 定义评论、回复、照片和页面模型 477
 - 27.3.1 定义Page模型 477
 - 27.3.2 定义Photo模型 478
 - 27.3.3 定义CommentThread模型 478
 - 27.4 创建评论服务器 479
 - 27.5 实现路由，以支持查看和添加评论 480
 - 27.6 实现基于模型的控制器路由 481
 - 27.6.1 实现Page模型控制器 481
 - 27.6.2 实现Photo模型控制器 482
 - 27.6.3 实现CommentThread模型控制器 482
 - 27.7 实现照片和评论视图 486
 - 27.7.1 实现照片视图 486
 - 27.7.2 实现评论主题视图 489
 - 27.7.3 实现评论视图 491
 - 27.7.4 添加CSS来设置视图的样式 492
 - 27.8 实现AngularJS模块和控制器来支持评论视图 493
 - 27.8.1 创建一个评论服务 493
 - 27.8.2 实现照片控制器 494
 - 27.8.3 实现页面AngularJS控制器 495
 - 27.8.4 完整的AngularJS应用程序 496
 - 27.9 初始化应用程序 499
 - 27.10 小结 500
 - 27.11 下一章 500
- 第28章 创建你自己的购物车 501
 - 28.1 项目简介 501
 - 28.2 用到的库 502
 - 28.3 项目的目录结构 502
 - 28.4 定义顾客、产品和订单模式 504
 - 28.4.1 定义地址模式 504
 - 28.4.2 定义账单模式 505
 - 28.4.3 定义产品模式 505
 - 28.4.4 定义数量模式 506
 - 28.4.5 定义订单模式 506
 - 28.4.6 定义顾客模式 506
 - 28.5 创建购物车服务器 508
 - 28.6 实现路由，以支持产品、购物车和订单请求 508
 - 28.7 实现基于模型的控制器路由 509
 - 28.7.1 实现产品模型控制器 509
 - 28.7.2 实现订单模型控制器 510
 - 28.7.3 实现顾客模型控制器 511

- 28.8 实现购物车和结账视图 513
 - 28.8.1 实现购物视图 513
 - 28.8.2 实现产品视图 515
 - 28.8.3 实现产品页面视图 516
 - 28.8.4 实现购物车视图 516
 - 28.8.5 实现发货视图 518
 - 28.8.6 实现账单视图 519
 - 28.8.7 实现复核视图 521
 - 28.8.8 实现订单视图 523
 - 28.8.9 添加CSS来设置视图的样式 524
- 28.9 实现AngularJS模块和控制器，以支持购物车视图 526
 - 28.9.1 初始化购物作用域 526
 - 28.9.2 实现辅助函数 527
 - 28.9.3 将物品添加到购物车 527
 - 28.9.4 从购物车删除物品 528
 - 28.9.5 结账 529
 - 28.9.6 设置发货信息 529
 - 28.9.7 验证账单 530
 - 28.9.8 执行购买 530
 - 28.9.9 完整的控制器 531
- 28.10 初始化应用程序 534
- 28.11 小结 536
- 28.12 下一章 536
- 第29章 构建交互式的Web 2.0应用程序组件 537
 - 29.1 项目简介 537
 - 29.2 用到的库 538
 - 29.3 项目的目录结构 538
 - 29.4 定义项目模型 540
 - 29.5 创建应用程序服务器 540
 - 29.6 实现支持视图的路由 541
 - 29.7 实现选项卡视图 542
 - 29.7.1 创建AngularJS选项卡视图模板 542
 - 29.7.2 定义窗格模板 542
 - 29.7.3 定义卡片模板 542
 - 29.7.4 实现选项卡视图 543
 - 29.7.5 实现AngularJS自定义指令 544
 - 29.7.6 设置卡片视图的样式 545
 - 29.8 实现天气服务视图 546
 - 29.8.1 创建后端天气服务 546
 - 29.8.2 定义天气AngularJS控制器 548
 - 29.8.3 定义天气AngularJS视图 549
 - 29.8.4 设置天气视图的样式 549
 - 29.9 实现可拖动的元素 551
 - 29.9.1 定义可拖动的自定义AngularJS指令 551
 - 29.9.2 在AngularJS视图中实现可拖动的指令 552
 - 29.9.3 设置可拖动视图的样式 552
 - 29.10 实现动态数据访问 553
 - 29.10.1 创建/words路由的Express路由控制器 553
 - 29.10.2 定义表格AngularJS控制器 554
 - 29.10.3 实现表格AngularJS视图 556
 - 29.10.4 设置表格视图的样式 557
 - 29.11 初始化应用程序 558
 - 29.12 小结 560

• • • • • ([收起](#))

[Node.js+MongoDB+AngularJS Web开发_下载链接1](#)

标签

Node.js

angular

mongodb

node.js

node

web

asd

AngularJS

评论

不是什么完整指南，单独的写完三种技术而且写得并不好

涉及面挺广的，但是也都只是浅尝辄止

整个书看完基本可以对完事的前+后端的设计有一定的认识。
书中针对各个模板的内容，都有单独的讲解，知识点讲解还不错。适合新手阅读

[Node.js+MongoDB+AngularJS Web开发_下载链接1](#)

书评

图书馆有，我选了其中两章看，翻译堪比机翻，就在目录前面用铅笔写上“翻译太差，别浪费时间”，还回去了。
下次选书一定要看译者，野路子就算了，这种一句译者介绍都没有的更难保证质量，基本都是抱着所谓“学习”的心态在翻的，简直是在糟蹋东西。

这本书介绍的知识很全面，基本上把做WEB的知识点都涉及到了，而且也同步配了很多代码。
但是本书最大的遗憾是没有任何一个完整的WEB实例：从最基本的环境搭建到API应用的完成，以至到最后的WEB项目完成。如果能够把Node.js MongoDB AngularJS三者融合成一个WEB项目实...

[Node.js+MongoDB+AngularJS Web开发_下载链接1](#)