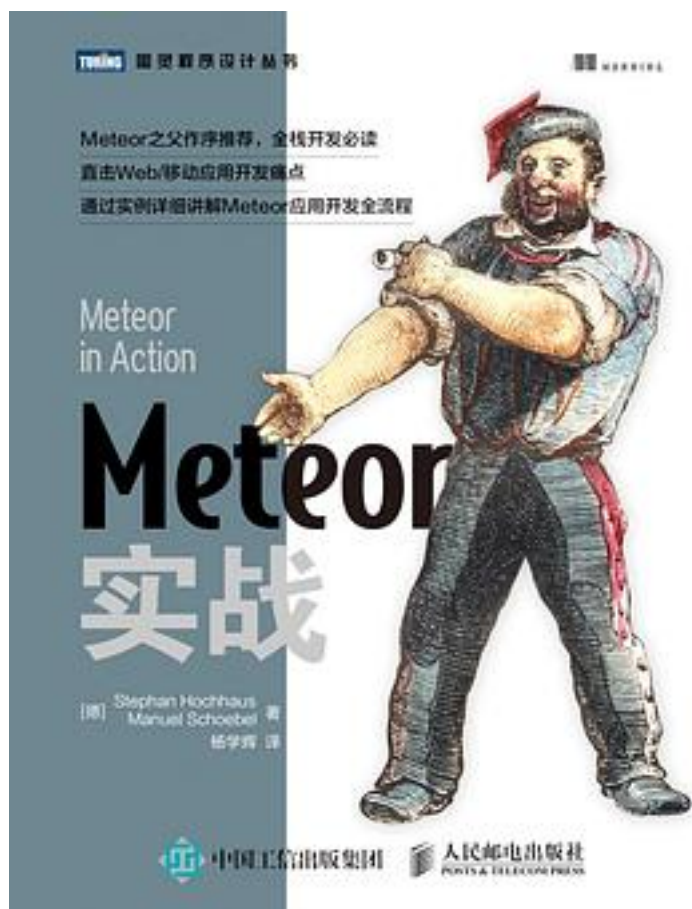


Meteor实战



[Meteor实战_下载链接1](#)

著者:[德] Stephan Hochhaus

出版者:人民邮电出版社

出版时间:2017-4

装帧:平装

isbn:9787115450173

本书秉承“实战”系列图书的一贯风格，以解决开发者实际问题为出发点，通过Meteor平台构建可扩展的高性能应用。全书共分为三部分、12章，详细介绍了如何用Meteor进行全栈开发，涵盖了Meteor栈的所有关键部分，涉及构成Meteor栈的各种组件和概念、响应式应用的基本模块和应用的构建与合理部署等。作者对MongoDB、路由、包

等进行了深入探讨，通过诸多实际案例，让读者全面掌握如何充分发挥Meteor在服务
器端和可扩展性上的优势。

作者介绍:

作者简介:

Stephan Hochhaus

资深Web开发人员，在可扩展的Web解决方案方面具有丰富的经验。精通JavaScript、
PHP、C#和Java。

Manuel Schoebel

资深Web开发人员，其技术博客广受开发者欢迎。

两人均是从Meteor诞生之初就一直在工作上运用Meteor。

译者简介:

杨学辉

自由程序员，译者，独立博客——猎数博客——的博主。数学系毕业的技术爱好者，兴
趣广泛，致力于学习和开发新的技术，以便高效解决现实生活中的实际问题。深入的了
解和交流，可以访问他的博客：<http://www.bagualu.net>。

目录: 第一部分 看，一颗流星！

第1章 构建应用程序的更好方式 2

1.1 Meteor简介 3

1.1.1 Meteor背后的故事 3

1.1.2 Meteor栈 4

1.1.3 同构框架：全栈式JavaScript 5

1.1.4 在浏览器中处理：在分布式平台上运行 7

1.1.5 响应式编程 8

1.2 Meteor的工作原理 10

1.2.1 核心项目 11

1.2.2 Isobuild和CLI工具 13

1.2.3 客户端代码和服务器端代码 14

1.3 优势和劣势 15

1.3.1 使用Meteor的好处 16

1.3.2 使用Meteor时的挑战 17

1.4 创建新的应用程序 19

1.4.1 创建新项目 19

1.4.2 启动应用 20

1.5 剖析默认项目 21

1.5.1 helloWorld.css 21

1.5.2 helloWorld.html 21

1.5.3 helloWorld.js 22

1.6 总结 24

第2章 我的冰箱：一个响应式游戏 25

2.1 应用概述 25

2.2 初始设置 26

- 2.3 创建布局 27
 - 2.3.1 设置样式 27
 - 2.3.2 添加模板 29
- 2.4 向数据库中实时添加内容 31
 - 2.4.1 在数据库中存储物品 31
 - 2.4.2 将数据连接到模板 33
 - 2.4.3 添加一组预定义的物品 35
- 2.5 把物品放进冰箱里 36
 - 2.5.1 为项目添加jQuery UI 36
 - 2.5.2 为物品定义拖放目标 37
 - 2.5.3 允许物品被拖动 39
- 2.6 部署应用到meteor.com并使用它 40
- 2.7 总结 40
- 第二部分 3, 2, 1——撞击!
- 第3章 模板 42
 - 3.1 模板介绍 42
 - 3.2 使用模板 44
 - 3.2.1 Blaze引擎 44
 - 3.2.2 组织模板文件 45
 - 3.3 创建动态HTML模板 45
 - 3.3.1 双重和三重大括号标签 (表达式) 46
 - 3.3.2 包含标签 (局部模板) 48
 - 3.3.3 块标签 50
 - 3.3.4 辅助函数 53
 - 3.4 处理事件 58
 - 3.4.1 模板的事件映射 58
 - 3.4.2 事件传播 59
 - 3.4.3 阻止浏览器的默认行为 60
 - 3.5 模板生命周期 61
 - 3.6 总结 63
- 第4章 数据 64
 - 4.1 Meteor的默认数据源 66
 - 4.1.1 什么使数据源具有响应性 67
 - 4.1.2 如何将响应式数据连接到函数 68
 - 4.2 构建房屋保姆应用 69
 - 4.2.1 设置模板 71
 - 4.2.2 连接到数据库并声明集合 71
 - 4.3 Session对象 72
 - 4.3.1 Session对象简介 73
 - 4.3.2 使用Session存储选定的下拉值 73
 - 4.3.3 使用Tracker.autorun创建响应性上下文 75
 - 4.4 MongoDB集合 76
 - 4.4.1 在MongoDB中查询文档 77
 - 4.4.2 Meteor的集合 78
 - 4.4.3 初始化集合 78
 - 4.4.4 查询集合 79
 - 4.4.5 在模板中显示集合数据 80
 - 4.4.6 在集合中更新数据 83
 - 4.4.7 向集合中插入新数据 85
 - 4.4.8 从集合中删除数据 87
 - 4.5 总结 88

- 第5章 全响应式编辑 89
 - 5.1 响应式编辑的工作流程 89
 - 5.2 响应式前端与DOM操作 90
 - 5.3 在本地集合中进行临时更改 94
 - 5.4 在表单中显示集合数据 97
 - 5.5 使用本地集合进行响应式更新 100
 - 5.5.1 houseForm模板的事件映射 101
 - 5.5.2 plantFieldset模板的事件映射 102
 - 5.6 实现一个简单的通知系统 105
 - 5.6.1 添加通知模板 106
 - 5.6.2 添加状态属性 106
 - 5.6.3 使用会话变量触发通知 107
 - 5.7 总结 109
- 第6章 用户、认证和权限 110
 - 6.1 将用户加入应用 111
 - 6.1.1 添加密码认证 111
 - 6.1.2 注册和密码重置 113
 - 6.1.3 设置邮件 117
 - 6.2 使用OAuth认证用户 120
 - 6.2.1 OAuth介绍 121
 - 6.2.2 整合Facebook认证 122
 - 6.2.3 集成其他的OAuth提供者 126
 - 6.3 管理用户权限、角色和组 127
 - 6.4 总结 130
- 第7章 数据交换 131
 - 7.1 发布和订阅 132
 - 7.1.1 publish()和subscribe() 133
 - 7.1.2 全局订阅 134
 - 7.1.3 模板级订阅 135
 - 7.1.4 参数化订阅 136
 - 7.1.5 向客户端独有的集合发布汇总数据 138
 - 7.1.6 将聚合发布变成响应式数据源 141
 - 7.1.7 通过用户ID限制数据可见性 143
 - 7.2 Meteor的方法 145
 - 7.2.1 删除insecure包 146
 - 7.2.2 使用方法将数据写入集合 147
 - 7.3 总结 150
- 第8章 路由 151
 - 8.1 Web应用中的路由 151
 - 8.2 客户端路由 154
 - 8.2.1 添加Iron.Router 155
 - 8.2.2 创建第一个路由 155
 - 8.2.3 基于路由定义布局 157
 - 8.2.4 根据路由设置数据上下文 161
 - 8.2.5 使用Iron.Router订阅数据 163
 - 8.3 高级的路由方法 166
 - 8.3.1 使用命名路由和链接辅助函数 166
 - 8.3.2 让活动路由有更好的导航链接 167

- 8.3.3 等待外部库加载 168
- 8.3.4 将路由组织为控制器 169
- 8.3.5 使用钩子扩展路由过程 171
- 8.3.6 创建Iron.Router插件 172
- 8.4 用于REST API的服务器端路由 173
- 8.5 总结 175
- 第9章 包 176
 - 9.1 所有应用的基础 176
 - 9.2 使用Isopack 177
 - 9.2.1 版本求解器和有语义的版本号 178
 - 9.2.2 查找包 179
 - 9.2.3 添加和删除Isopack 182
 - 9.2.4 更新包 183
 - 9.3 使用npm包 183
 - 9.4 创建Isopack 184
 - 9.4.1 创建包 185
 - 9.4.2 声明包的元数据 186
 - 9.4.3 添加包的功能 188
 - 9.4.4 使用tinytest测试Isopack 191
 - 9.4.5 发布 194
 - 9.5 总结 195
- 第10章 高级服务器方法 196
 - 10.1 再次介绍Node.js 196
 - 10.1.1 同步代码 197
 - 10.1.2 异步代码 198
 - 10.2 使用纤维的异步功能 200
 - 10.2.1 将多任务引入事件循环 200
 - 10.2.2 使用wrapAsync为纤维绑定回调函数 203
 - 10.2.3 为单个客户端解除方法调用阻塞 204
 - 10.2.4 使用bindEnvironment创建纤维 206
 - 10.3 整合外部API 207
 - 10.3.1 使用HTTP包进行RESTful调用 208
 - 10.3.2 使用同步方法查询API 208
 - 10.3.3 使用异步方法调用API 209
 - 10.4 将文件上传到集合 211
 - 10.5 总结 215
- 第三部分 走出陨石坑
- 第11章 构建和调试 218
 - 11.1 Meteor的构建过程 218
 - 11.1.1 构建阶段 220
 - 11.1.2 使用--production选项运行 223
 - 11.1.3 加载顺序 223
 - 11.1.4 通过包添加构建阶段 226
 - 11.1.5 添加自定义构建阶段 227
 - 11.2 访问正在运行的应用 229
 - 11.2.1 使用交互式服务器外壳 229
 - 11.2.2 使用node-inspector进行调试 230

- 11.3 创建浏览器应用 232
 - 11.3.1 使用Meteor.settings进行应用配置 232
 - 11.3.2 构建Meteor项目 233
- 11.4 创建移动应用 235
 - 11.4.1 使用Cordova的混合应用 235
 - 11.4.2 加入移动平台 236
 - 11.4.3 配置移动应用 238
 - 11.4.4 添加移动功能 240
- 11.5 总结 242
- 第12章 开始生产 243
 - 12.1 准备生产 243
 - 12.1.1 使用版本控制 243
 - 12.1.2 功能测试：Velocity框架 245
 - 12.1.3 估算和测试负载 247
 - 12.1.4 服务器管理 249
 - 12.1.5 清单 250
 - 12.2 安装和部署 250
 - 12.2.1 最简单的部署：meteor.com 251
 - 12.2.2 无所不包的主机：云供应商 253
 - 12.2.3 最灵活的方式：手动设置 256
 - 12.3 将各部分连接起来 259
 - 12.3.1 环境变量 260
 - 12.3.2 Meteor和MongoDB的连接 260
 - 12.4 扩展策略 261
 - 12.4.1 使用冗余的主动?被动高可用性 262
 - 12.4.2 单组件部署 263
 - 12.4.3 冗余和负载均衡 263
 - 12.4.4 绝对可用性 265
 - 12.5 总结 266
- 附录A 安装Meteor 267
- 附录B MongoDB剖析 271
- 附录C 设置nginx 278
- • • • • ([收起](#))

[Meteor实战 下载链接1](#)

标签

全栈

web

@上图

评论

[Meteor实战 下载链接1](#)

书评

[Meteor实战 下载链接1](#)