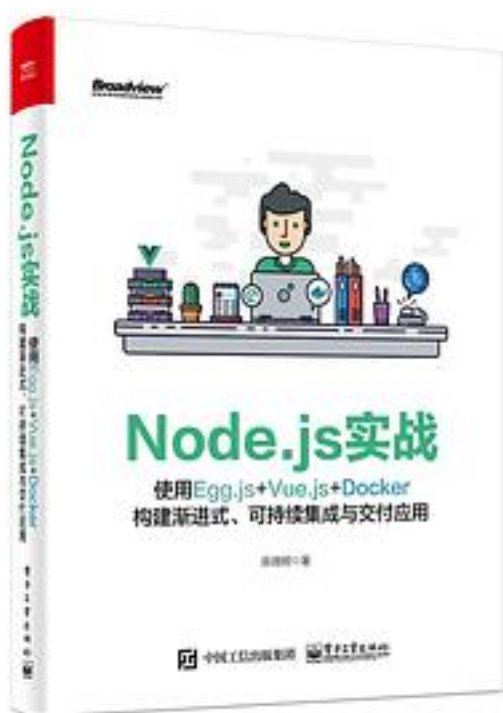


Node.js实战：使用Egg.js+Vue.js+Docker构建渐进式、可持续集成与交付应用



[Node.js实战：使用Egg.js+Vue.js+Docker构建渐进式、可持续集成与交付应用 下载链接1](#)

著者:yugo

出版者:电子工业出版社

出版时间:2018-9

装帧:平装

isbn:9787121349126

《Node.js实战：使用Egg.js+Vue.js+Docker构建渐进式、可持续集成与交付应用》讲解Node.js在Web开发方面的实际应用，以一个类Dribbble图片画廊应用为实例，内容包括底层的Koa.js/Egg.js框架核心与实现原理，上层服务的构建、OAuth服务、JWT登录认证服务、前后端分离架构，以及使用TypeScript和Vue.js实现前后端同构的前端界面，解决SEO问题，部署与持续集成，使用时下流行的Docker实现DevOps。最后还介绍了压力测试与上线之后的数据收集的注意事项，可解决日常企业需求。

《Node.js实战：使用Egg.js+Vue.js+Docker构建渐进式、可持续集成与交付应用》适合从事Web开发并对Node.js感兴趣的读者阅读。

作者介绍:

目录: 第1章 Node.js的优势	1
1.1 为什么是JavaScript语言	1
1.2 为什么经常说Node.js不适合大型应用	3
第2章 Egg.js框架核心原理与实现	6
2.1 异步基础	6
2.2 Koa.js基础知识	15
2.2.1 Koa.js 中间件核心代码	16
2.2.2 Koa.js插件	18
2.3 Egg.js 基础知识	21
2.3.1 实现 egg-core	22
2.3.2 实现 egg-init	26
2.3.3 实现egg-cluster	30
2.4 Egg.js插件	33
2.4.1 egg-socket.io	33
2.4.2 原理解读	39
2.5 制作一个Egg.js插件	43
第3章 构建后端API服务	52
3.1 安装相关组件	52
3.2 发布一个插件	59
3.2.1 创建Flash插件	59
3.2.2 使用egg-msg-flash	72
3.2.3 使用egg-y-validator	73
3.3 规范化	73
3.3.1 添加新的scripts支持ESLint自修复	74
3.3.2 添加插件支持	74
3.3.3 prettier格式化工具	76
3.3.4 同步代码编辑器配置	76
3.4 第一个JSON请求	77
3.4.1 给全局添加一些方法	77
3.4.2 全局化一些东西	84
3.4.3 自动路由	86
3.4.4 创建 PostMan 测试	88
3.5 注册服务	91
3.5.1 Invitation模型	91
3.5.2 注释	93
3.5.3 User模型	96
3.5.4 修改控制器	97
3.5.5 添加验证逻辑	98
3.5.6 帮助方法	99
3.5.7 User服务	101
3.5.8 PostMan测试	103
3.6 登录服务	104
3.7 邮件与调试	115
3.7.1 理解发送邮件的原理	115
3.7.2 安装邮件插件	115
3.7.3 环境与调试	116
3.7.4 全局调试	118

- 3.7.5 VSCode全局调试 121
- 3.7.6 发送验证邮件 122
- 3.7.7 添加逻辑 125
- 3.7.8 验证 135
- 3.8 构建 RESTful API 137
 - 3.8.1 什么是RESTful API 137
 - 3.8.2 创建RESTController基础类便于继承 138
 - 3.8.3 测试Images RESTful API 141
 - 3.8.4 构建后台的REST路由 143
 - 3.8.5 构建控制器 145
 - 3.8.6 测试后台路由 148
 - 3.8.7 关于验证 149
- 3.9 安全地开放API 151
- 3.10 实现OAuth接口 158
 - 3.10.1 实现授权码官方文档所要求的接口 158
 - 3.10.2 实现刷新验证码接口 167
 - 3.10.3 实现 authenticate 所需接口 169
- 3.11 完善OAuth与测试 170
 - 3.11.1 发放Token 170
 - 3.11.2 新建客户端项目 172
 - 3.11.3 测试OAuth 173
- 3.12 支付宝支付 176
 - 3.12.1 什么是非对称加密 176
 - 3.12.2 注册支付宝 176
 - 3.12.3 生成非对称秘钥 176
 - 3.12.4 实现 177
 - 3.12.5 添加路由 180
 - 3.12.6 内网穿透 180
 - 3.12.7 测试 182
- 3.13 社会化登录 183
- 第4章 构建后台管理页面 189
 - 4.1 后端开发 189
 - 4.1.1 安装VSCode插件 189
 - 4.1.2 获取脚手架 189
 - 4.1.3 安装依赖 190
 - 4.1.4 修改代码 190
 - 4.1.5 跨域请求 191
 - 4.1.6 修改后端代码支持跨域 192
 - 4.1.7 在前端添加存储 198
 - 4.2 模型列表 200
 - 4.3 添加数据 209
 - 4.4 修改逻辑 220
- 第5章 前端界面设计与实现 228
 - 5.1 搭建前端开发环境 228
 - 5.1.1 开始 228
 - 5.1.2 创建Header头部 229
 - 5.1.3 将变量提取出来 234
 - 5.1.4 添加路径重写 235
 - 5.2 AppFooter组件 237
 - 5.2.1 做一些配置 237
 - 5.2.2 创建src/components/layouts/AppFooter.vue 238
 - 5.2.3 网络识别信息 247
 - 5.2.4 修改一下全局样式 247
 - 5.2.5 查看页面 248

- 5.2.6 提升编译速度 248
- 5.3 首页 249
- 5.4 替换成为真实数据 269
 - 5.4.1 完成后端Image API 269
 - 5.4.2 修改首页的代码 271
 - 5.4.3 添加API逻辑 276
 - 5.4.4 效果 278
- 5.5 图片详情页 278
 - 5.5.1 创建路由 279
 - 5.5.2 安装依赖 279
 - 5.5.3 创建视图 279
 - 5.5.4 添加插件 285
 - 5.5.5 创建评论组件 286
 - 5.5.6 测试 290
 - 5.5.7 关于服务端访问DOM 290
- 5.6 注册页面 294
 - 5.6.1 注册路由 294
 - 5.6.2 新建signup.vue页面 295
 - 5.6.3 增强错误提示 299
- 5.7 登录页面 299
- 5.8 完善详情与评论 310
- 5.9 个人中心 321
- 5.10 创建图片 336
 - 5.10.1 创建又拍云存储 336
 - 5.10.2 添加后端API 338
 - 5.10.3 前端界面 340
 - 5.10.4 测试 348
- 5.11 团队 349
 - 5.11.1 功能是如何工作的 350
 - 5.11.2 数据库 350
 - 5.11.3 后端 356
 - 5.11.4 前端 363
 - 5.11.5 测试 372
- 第6章 部署与运维 374
 - 6.1 认识Docker 374
 - 6.1.1 解决了什么问题 374
 - 6.1.2 使用Docker的流程 375
 - 6.1.3 安装Docker 378
 - 6.1.4 使用加速器 378
 - 6.1.5 下载一个基础镜像 379
 - 6.1.6 hello world 379
 - 6.2 手动构建镜像 380
 - 6.3 编写Dockerfile文件 384
 - 6.4 Docker Compose 387
 - 6.4.1 安装docker-compose 387
 - 6.4.2 命令行接口 388
 - 6.4.3 Egg.js简单实例 389
 - 6.4.4 增加服务 391
 - 6.5 集群 396
 - 6.5.1 Docker集群 396
 - 6.5.2 集群初始化 396
 - 6.5.3 实例 397
 - 6.6 持续部署 400
 - 6.6.1 部署主机免密码登录 400

- 6.6.2 客户端钩子 401
- 6.6.3 使用服务端钩子进行部署 403
- 6.6.4 使用 shipit 404
- 6.6.5 使用 Ansible 部署 406
- 6.7 持续集成 409
- 6.8 Kubernetes集群 423
 - 6.8.1 简单使用 423
 - 6.8.2 如何创建应用 425
 - 6.8.3 命令行管理 430
 - 6.8.4 通过UI创建应用 433
 - 6.8.5 添加持续集成 439
 - 6.8.6 固定IP地址 441
 - 6.8.7 部署前端 442
- 第7章 性能分析与优化 448
 - 7.1 服务器性能分析与测试 448
 - 7.2 用户追踪 458
 - 7.2.1 百度分析 458
 - 7.2.2 Google分析 460
 - 7.2.3 其他付费服务 461
 - 7.3 前端性能分析与优化 461
 - 7.3.1 lighthouse 461
 - 7.3.2 sonarwhal 462
 - 7.3.3 图片压缩 464
 - 7.3.4 错误上报 465
 - 7.3.5 接收用户反馈 466
 - • • • • [\(收起\)](#)

[Node.js实战：使用Egg.js+Vue.js+Docker构建渐进式、可持续集成与交付应用_下载链接1](#)

标签

nodejs

eggjs

Web前端

docker

Web前端开发

评论

内容比较全面，学到很多egg的小技巧，适合有一点egg基础的读者阅读，作者很多对代码和项目的见解和我很一致。koa，docker，k8s，性能与测试不太详细

书是有内容的，但是我感觉作者有炫耀的成分，用pug写html，用stylus写css，用class ts写vue，看懂的成本太大

主要目的是对全栈有一个比较粗浅但基本点都有的认识。已达到目的。

写的章法真的8行

想要看看别人用node搭建的整站，并且想知道每个阶段的意义可以读一下，给自己项目做个参考，但是想细读从中学习很多知识比较困难，因为通篇的代码没有注释，想要细致研究耗不起时间

作者想表达清楚但又过于精简，或者说不知道读者的痛点，很多篇幅就是报流水账，所以感觉本书不适合基础薄弱的读者

真的是垃圾，完全就是开发笔记流水账，毫无章法思维，完全是作者根据写代码的思路在写书

作者不适合写书，完全看不出组织的逻辑在哪，通篇流水账

[Node.js实战：使用Egg.js+Vue.js+Docker构建渐进式、可持续集成与交付应用_下载链接1](#)

[Node.js实战：使用Egg.js+Vue.js+Docker构建渐进式、可持续集成与交付应用_下载链接1](#)