

# JavaScript测试驱动开发



[JavaScript测试驱动开发\\_下载链接1](#)

著者:[美] Venkat Subramaniam

出版者:人民邮电出版社

出版时间:2018-3

装帧:平装

isbn:9787115477156

JavaScript已经成为使用最广泛的语言之一，它强大且高度灵活，但同时也颇具风险，所以应该用更出色的开发实践来支持。自动化测试和持续集成就是很好的方法，可以降低JavaScript带来的风险。本书介绍JavaScript自动化测试及其相关实践，主体内容包括两部分：第一部分涵盖自动化测试的基础，介绍如何为同步函数和异步函数编写测试

，以及当代码包含复杂的依赖关系时如何实现自动化测试；第二部分通过一个测试驱动开发的示例，让读者能够运用在第一部分所学的内容，为客户端和服务端编写自动化测试。本书在帮助读者学习和研究测试工具和技术的同时，还会介绍一些软件设计原则，有助于实现轻量级设计，并得到可维护的代码。

作者介绍:

作者简介:

Venkat Subramaniam

博士，Agile

Developer公司创始人，敏捷开发先驱，通过演讲、著书、培训等方式影响了全球数十万软件开发人员。另著有《高效程序员的45个习惯：敏捷开发修炼之道》《Scala程序设计：Java虚拟机多核编程实战》《Groovy程序设计》等书。

译者简介:

毛姝雯

对编程有着浓烈的兴趣，iOS开发、Web前端开发、UI设计等领域都有涉足。曾就职于IBM，专注于移动端Web开发。现在已经是一个可爱宝宝的母亲。

目录: 第1章 自动化测试让你重获自由 1

1.1 变更的挑战 1

1.1.1 变更的成本 1

1.1.2 变更的影响 2

1.2 测试与验证 2

1.3 采用自动化验证 3

1.4 为什么难以验证 5

1.5 如何实现自动化测试 6

1.6 小结 6

第一部分 创建自动化测试

第2章 测试驱动设计 10

2.1 让我们开始吧 10

2.1.1 检查npm和Node.js的安装 11

2.1.2 创建示例项目 11

2.1.3 创建测试套件和金丝雀测试 12

2.1.4 验证函数的行为 14

2.1.5 验证另一个数据 16

2.2 正向测试、反向测试和异常测试 18

2.3 设计服务器端代码 20

2.3.1 从测试列表开始 20

2.3.2 回文项目 21

2.3.3 编写正向测试 22

2.3.4 编写反向测试 26

2.3.5 编写异常测试 27

2.4 评估服务器端代码覆盖率 29

2.5 为测试客户端代码做准备 31

2.5.1 切换到客户端项目 31

2.5.2 配置Karma 33

2.5.3 从金丝雀测试开始 34

- 2.6 设计客户端代码 34
- 2.7 评估客户端代码覆盖率 37
- 2.8 小结 38
- 第3章 异步测试 39
  - 3.1 服务器端回调 39
    - 3.1.1 一次天真的尝试 40
    - 3.1.2 编写异步测试 41
    - 3.1.3 编写一个反向测试 43
  - 3.2 客户端的回调函数 44
  - 3.3 测试promise 46
    - 3.3.1 对promise的简单介绍 46
    - 3.3.2 promise异步测试的类型 47
    - 3.3.3 返回promise对象的函数 48
    - 3.3.4 使用done()进行测试 49
    - 3.3.5 返回promise的测试 . 49
    - 3.3.6 使用chai-as-promised 50
    - 3.3.7 结合eventually和done() 50
    - 3.3.8 为promise编写反向测试 51
  - 3.4 小结 52
- 第4章 巧妙处理依赖 53
  - 4.1 问题以及spike解决方案 53
    - 4.1.1 转移到spike项目 53
    - 4.1.2 从spike中获得见解 54
  - 4.2 模块化设计 55
  - 4.3 尽量分离依赖 56
    - 4.3.1 结束spike，准备自动化测试 56
    - 4.3.2 测试creatURL 57
  - 4.4 使用测试替身 59
  - 4.5 依赖注入 61
  - 4.6 交互测试 62
  - 4.7 使用Sinon清理测试代码 64
    - 4.7.1 安装Sinon 65
    - 4.7.2 初探Sinon 65
    - 4.7.3 使用Sinon的mock测试交互 67
    - 4.7.4 使用Sinon的stub测试状态 68
    - 4.7.5 使用Sinon的spy拦截调用 70
  - 4.8 回顾与继续 72
  - 4.9 小结 75
- 第二部分 真实的自动化测试
- 第5章 Node.js测试驱动开发 78
  - 5.1 从策略设计开始——适度即可 78
  - 5.2 深入战略设计——测试优先 79
    - 5.2.1 创建初始测试列表 79
    - 5.2.2 编写第一个测试 80
    - 5.2.3 编写一个正向测试 82
  - 5.3 继续设计 85
    - 5.3.1 readTickersFile的反向测试 85
    - 5.3.2 设计parseTickers函数 87
    - 5.3.3 设计processTickers函数 88
  - 5.4 创建spike以获得启发 89
    - 5.4.1 为getPrice创建spike 89
    - 5.4.2 设计getPrice函数 90
  - 5.5 模块化以易于测试 93
    - 5.5.1 设计processResponse和processError函数 93

- 5.5.2 设计processHttpError 95
- 5.5.3 设计parsePrice和processError 96
- 5.6 分离关注点 97
  - 5.6.1 设计printReport 98
  - 5.6.2 设计sortData 98
- 5.7 集成和运行 100
- 5.8 回顾代码覆盖率和设计 102
  - 5.8.1 评估代码覆盖率 102
  - 5.8.2 代码设计 103
- 5.9 提供HTTP访问 104
- 5.10 小结 105
- 第6章 Express测试驱动开发 106
  - 6.1 为可测试性设计 106
    - 6.1.1 创建策略设计 107
    - 6.1.2 通过测试创建战略设计 108
  - 6.2 创建Express应用并运行金丝雀测试 108
  - 6.3 设计数据库连接 109
  - 6.4 设计模型 113
    - 6.4.1 建立数据库连接和测试固件 113
    - 6.4.2 设计all函数 114
    - 6.4.3 设计get函数 115
    - 6.4.4 设计add函数 116
    - 6.4.5 处理delete函数 121
    - 6.4.6 设计共享的校验代码 123
  - 6.5 设计路由函数 126
    - 6.5.1 重温路由 127
    - 6.5.2 从为Router创建stub开始 128
    - 6.5.3 测试路径/的GET方法 129
    - 6.5.4 测试路径/:id的GET方法 131
    - 6.5.5 处理路径/的POST方法 133
    - 6.5.6 以路径/:id的DELETE方法结束整个测试 135
  - 6.6 评估代码覆盖率 136
  - 6.7 运行应用 138
    - 6.7.1 使用Curl 139
    - 6.7.2 使用Chrome扩展程序 140
    - 6.7.3 观察响应 141
  - 6.8 小结 141
- 第7章 与DOM和jQuery协作 142
  - 7.1 创建策略设计 142
  - 7.2 通过测试创建战略设计 143
    - 7.2.1 创建测试列表 143
    - 7.2.2 创建项目 144
  - 7.3 增量开发 145
    - 7.3.1 设计getTasks 146
    - 7.3.2 更新DOM 147
    - 7.3.3 调用服务 150
    - 7.3.4 注册window对象的onload事件 154
  - 7.4 运行 UI 155
  - 7.5 完成设计 156
    - 7.5.1 设计addTask 157
    - 7.5.2 设计deleteTask 163
  - 7.6 使用jQuery进行测试 165
    - 7.6.1 准备工作 166
    - 7.6.2 使用jQuery选择器 166

- 7.6.3 使用\$.ajax验证调用 167
- 7.6.4 测试document的ready函数 168
- 7.6.5 完整的测试和使用jQuery的代码实现 170
- 7.7 评估代码覆盖率 170
- 7.8 小结 171
- 第8章 使用AngularJS 172
- 8.1 测试AngularJS的方式 172
- 8.2 初步设计 175
- 8.3 关注控制器 176
  - 8.3.1 准备工作空间 176
  - 8.3.2 编写第一个测试 177
  - 8.3.3 设计控制器 177
- 8.4 设计服务交互 180
- 8.5 分离关注点，减少mock 182
  - 8.5.1 找到合适的地方 183
  - 8.5.2 结合经验测试和交互测试 183
  - 8.5.3 测试加载顺序 186
- 8.6 继续设计 187
  - 8.6.1 设计addTask 188
  - 8.6.2 设计deleteTask 192
- 8.7 设计服务 194
  - 8.7.1 设计get 函数 194
  - 8.7.2 设计add 函数 197
  - 8.7.3 设计delete 函数 198
- 8.8 评估代码覆盖率 199
- 8.9 运行UI 200
- 8.10 小结 201
- 第9章 Angular 2测试驱动开发 202
- 9.1 通过spike学习Angular 202
  - 9.1.1 管道、服务和组件 203
  - 9.1.2 创建项目 203
  - 9.1.3 创建管道 204
  - 9.1.4 创建服务 205
  - 9.1.5 创建组件 207
  - 9.1.6 集成 211
- 9.2 通过测试设计Angular应用 213
  - 9.2.1 创建项目 214
  - 9.2.2 创建测试列表 215
- 9.3 测试驱动组件的设计 215
  - 9.3.1 验证是否设置组件属性 215
  - 9.3.2 初始化模型 218
  - 9.3.3 设计getTasks 219
  - 9.3.4 对任务进行排序 222
  - 9.3.5 验证依赖注入 224
- 9.4 测试驱动服务的设计 226
- 9.5 测试驱动管道的设计 230
- 9.6 测试驱动启动代码 234
- 9.7 集成 236
- 9.8 完成设计 237
  - 9.8.1 设计任务添加功能 238
  - 9.8.2 设计任务删除功能 245
  - 9.8.3 评估代码覆盖率 248
- 9.9 小结 249
- 第10章 集成测试和端到端测试 250

10.1 认识Protractor	250
10.1.1 使用Protractor的理由	251
10.1.2 安装Protractor	251
10.1.3 使用Protractor进行测试	251
10.1.4 为UI层测试做准备	253
10.1.5 编写第一个测试	254
10.1.6 测试数据发送	255
10.2 启动服务器和配置数据库	256
10.2.1 为TO-DO应用安装Protractor	257
10.2.2 在设置前启动服务器	257
10.2.3 为不同的环境创建数据库	259
10.2.4 在beforeEach中设置数据	260
10.3 测试jQuery UI	261
10.3.1 设置Protractor配置文件	261
10.3.2 发现必要的测试	261
10.3.3 实现集成测试	262
10.4 使用页面对象	265
10.5 测试AngularJS的UI	268
10.6 测试Angular 2的UI	272
10.7 小结	275
第11章 测试驱动你自己的应用	276
11.1 努力的成果	276
11.2 测试的规模和层次	279
11.3 测试驱动：程序员指南	280
11.4 测试驱动：团队领导、架构师指南	282
11.5 测试驱动：项目经理指南	283
11.5.1 促进可持续的敏捷开发实践	283
11.5.2 优雅地处理遗留应用	283
11.5.3 结束新的遗留应用	284
11.6 摇滚吧	285
附录 网络资源	286
参考文献	288
• • • • •	<a href="#">(收起)</a>

[JavaScript测试驱动开发 下载链接1](#)

## 标签

测试

JavaScript

编程

计算机

程序设计

Venkat • Subramaniam

美国

未资源

## 评论

测试入门书籍。全书分为三部分 - 为什么要进行测试 - 测试的基础：工具安装、测试Promise、测试依赖注入，sinon... - 测试实践：基于Express + (DOM/jQuery/Angular) 的TODO项目，带你测试数据库、服务端、客户端、代码覆盖率、集成测试。  
唯一的缺陷，可能就是里面使用的工具相比于现在来说有一点老旧，不过还是很经典的。现在的jest可以直接集成这些工具。  
总之，看完这本书，绝对帮你打开测试驱动开发的大门。  
还等什么，买鸭！！冲鸭！！！！

还不错，两天就读完了。适合有前端开发基础的人阅读。

不错

教你如何TDD

普及型的。

[JavaScript测试驱动开发\\_下载链接1](#)

书评

-----  
[JavaScript测试驱动开发\\_下载链接1](#)