

《精通自动化测试框架设计》

图书基本信息

书名：《精通自动化测试框架设计》

13位ISBN编号：978711541565X

出版时间：2016-4-1

作者：陈冬严,邵杰明,王东刚,蒋涛

页数：395

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《精通自动化测试框架设计》

内容概要

全书共分3部分，第1部分包括第1~8章，介绍了有关测试框架与UI自动化测试的主题。在第1章中介绍了若干种不同类型的自动化测试实践，并提出了T型自动化测试框架以及纺锤形自动化测试模型。在第2章介绍了自动化测试中如何进行数据的交互与管理，并结合实际案例列举了Java属性文件、XML、Excel、数据库以及CSV等不同的处理方式。第3章和第4章介绍如何分层构建Solvent、SeMet等底层框架。第5~8章是有关TestLink UI自动化测试框架的构建。第2部分是UI自动化框架横向拓展的主题。第3部分讲解API测试。实现了UI/API的分层自动化测试，并探讨了在敏捷模式下，通过统一框架，同一用例在不同技术实现方式下无缝切换运行的可行性和由此带来的价值。第16章介绍了Webdriver的基础概念及简单元素封装、自定义Page Object等内容。读者可以按照此书介绍的方法，根据自身工作中被测应用的实际特点，加上适当的工具，来尝试建立适合自身项目实际需求的自动化测试框架。本书是帮助读者从初学者踏上通往更高层级测试工程师的阶梯。本书适合作为测试初学者、测试从业人员、测试经理、测试管理人员、测试开发人员的参考用书，也可以作为大专院校相关专业的师生用书和培训学校的教材。

《精通自动化测试框架设计》

作者简介

陈冬严，浙江大学硕士，具有10年软件测试和团队管理的工作经验，先后服务于ITSM、PLM软件研发企业，现就职于某金融行业核心机构IT规划部门。业余时间喜欢园艺。

邵杰明，热爱测试工作，10多年的测试行业经验，曾先后供职于多家世界一流软件公司担任测试开发和测试管理工作，积累了丰富的行业工作经验，拥有PMP认证，目前担任测试架构师的工作，致力于自动化测试设计、持续交付等方面的工作。

王东刚，常用网名fastpoint，资深测试专家，《软件测试与JUnit实践》作者，在项目管理、产品开发和测试方面均有建树。也曾自主创业，主推专业的软件测试服务。个人网站：<http://www.fastpoint.cn/>。

蒋涛，高级测试工程师，现负责雅虎两个主要移动端产品的测试工作。从Selenium 1.0时代便开始使用其进行工作，期间经历过多个国际级的大型项目，属于国内较早一批接触Selenium的人之一，拥有丰富的自动化项目经验。现在主要工作重心在移动互联网上，部署、实施、开发过多个移动平台的自动化项目。同时也是一个摄影爱好者，喜欢独自背包旅行。

书籍目录

目 录

第1部分 构建UI自动化框架

第1章 BCO团队的自动化之旅 2

1.1 奥运年的新挑战 2

1.1.1 BCO是什么 2

1.1.2 为什么需要BCO 2

1.1.3 测试任务与测试内容 3

1.1.4 利益干系人 4

1.1.5 Pink Mail、Escalation和SPRTracker 5

1.1.6 沟通，还是沟通 5

1.2 史前的自动化 6

1.2.1 自动化安装系统 6

1.2.2 快速回归测试系统的混合自动化 6

1.2.3 来自小众自动化工具的框架 7

1.3 五天太久，还能压缩吗 8

1.3.1 BCO版本发布用时分布 8

1.3.2 缺陷压力测试 9

1.4 听说又有一个新框架了 10

1.4.1 初见Solvent 11

1.4.2 学习XPath和HTML 12

1.4.3 XPath定位工具 12

1.4.4 自信与挑战 13

1.4.5 找到适合自身的实践对象 14

1.4.6 框架的归属问题 14

1.4.7 Automation 100 15

1.5 冰山 15

1.5.1 假失败 15

1.5.2 低优先级的自动化Backlog 16

1.5.3 破窗与“造轮子” 16

1.6 再启航 16

1.6.1 自动化测试金字塔与冰淇淋 17

1.6.2 橄榄形的自动化测试 18

第2章 测试数据管理 19

2.1 简介 19

2.2 测试数据分类 19

2.3 测试数据交互基本方法 20

2.3.1 使用Java的系统属性 21

2.3.2 使用操作系统的环境变量 22

2.3.3 使用属性文件 23

2.4 使用XML文件 23

2.4.1 Webdriver中的定位方法 24

2.4.2 使用dom4j进行解析 25

2.5 使用Excel 27

2.5.1 经典的DataTable 27

2.5.2 强关键字驱动的自动化用例 28

2.5.3 Apache POI介绍 29

2.5.4 单个工作表的解析 30

- 2.5.5 整个工作簿的解析 34
- 2.5.6 用例解析与执行 35
- 2.5.7 Mock API类 38
- 2.5.8 Mock API适配器类 39
- 2.6 使用数据库 40
 - 2.6.1 JDBC连接数据库 41
 - 2.6.2 使用Apache DbUtils 44
 - 2.6.3 从TestLink数据库中读取数据 46
- 2.7 使用CSV文件 47
 - 2.7.1 CSV文件数据解析思路 48
 - 2.7.2 实现泛型解析 48
 - 2.7.3 对象类案例 49
 - 2.7.4 提供数据源的外部访问 50
 - 2.7.5 CSV文件通过SQL方式查询结果 50
- 第3章 构建Solvent 53
 - 3.1 框架是什么 53
 - 3.1.1 六万立方米的混凝土大底板 53
 - 3.1.2 测试框架及其目标 53
 - 3.2 Solvent与JUnit 54
 - 3.3 迟早，你需要一个基类 54
 - 3.3.1 用例基类SolventTestCase 55
 - 3.3.2 业务功能基类Solvent 57
 - 3.4 基于Log4j的日志 60
 - 3.4.1 Log4j简介 60
 - 3.4.2 一个简单的例子 62
 - 3.4.3 实现SolventLogger类 63
 - 3.4.4 使用SolventLogger 64
 - 3.5 使用TestWatcher截屏 65
 - 3.5.1 功能分析 65
 - 3.5.2 实现示例 65
 - 3.6 经改造的Apache计时器 66
 - 3.6.1 Apache的计时器 66
 - 3.6.2 Solvent增强的计时器 67
 - 3.6.3 部署计时器 68
 - 3.6.4 计时精度与延迟 69
 - 3.7 使用Timeout进行超时判负 70
 - 3.8 自定义检查点CheckPoint 70
 - 3.8.1 检查点类CheckPoint 70
 - 3.8.2 部署与使用 73
 - 3.9 Configurator提供统一框架配置参数 75
 - 3.9.1 Configurator类的实现 75
 - 3.9.2 双重参数文件的读取 78
 - 3.9.3 配置参数的使用 80
 - 3.10 基于XML的运行时参数处理 81
 - 3.10.1 来自XML的用例数据集 81
 - 3.10.2 InputFileFinder获取输入文件 82
 - 3.10.3 InputFileDigester进行解析 83
 - 3.10.4 数据结构之SolventTestDataSetBean 85
 - 3.10.5 数据结构之SolventTestDataSet 87

3.10.6	在SolventTestCase中完成解析	88
3.10.7	SolventTestCase提供统一数据获取方法	89
3.10.8	测试用例参数化案例	90
第4章	构建SeMet	92
4.1	SeMet是什么	92
4.2	引入SeMet类	92
4.3	引入TLTestCase	98
4.4	引入SeMetWebDriverSession类	99
4.5	引入Webdrivers工厂类	100
4.6	小结	101
第5章	TestLink自动化测试	102
5.1	自动化测试策略	102
5.2	为什么是TestLink	103
5.3	TestLink简介	105
5.4	TestLink安装	105
5.4.1	XAMPP的安装	106
5.4.2	TestLink部署	107
5.5	TestLink主要功能分析	108
5.5.1	基于文档的测试	108
5.5.2	TestLink基本业务对象	109
5.5.3	模块与功能点	109
5.6	TestLink页面分析	110
5.6.1	页面结构分析	110
5.6.2	Web组件分析	112
5.7	框架	113
5.7.1	Webdriver中有关(i)frame的处理方法	114
5.7.2	回到原点的frame处理方法	114
5.7.3	工具类FrameHelper	115
5.7.4	一级页面的使用方法	116
5.7.5	二级以下页面的使用方法	116
5.7.6	不同框架下使用组件的方法	117
5.8	左右选择框	118
5.8.1	页面分析及自动化实现	118
5.8.2	功能分析及自动化实现	120
5.9	浮动对话框	121
5.9.1	页面分析及自动化实现	121
5.9.2	功能分析及自动化实现	123
5.9.3	复选框的简单增强	125
5.10	树	126
5.10.1	页面分析及自动化实现	126
5.10.2	功能分析及自动化实现	128
5.11	表格	130
5.11.1	页面分析及自动化实现	131
5.11.2	功能分析及自动化实现	132
5.11.3	TableSeMet子类	134
5.12	下拉选择框	136
5.12.1	组件介绍	136
5.12.2	功能与测试点分析	137
5.12.3	SelectboxSeMet类	138

5.13	小结	140
第6章	页面封装—从登录到测试项目管理	142
6.1	登录——三段式页面封装法	142
6.2	测试项目对象TestProjectBean-使用方法链	144
6.3	创建测试项目类CreateProjectSeMet	146
6.3.1	页面定位	146
6.3.2	页面元素定位	146
6.3.3	元素获取与简单操作	146
6.3.4	高级业务操作—隔离业务与具体实现	148
6.4	统一测试项目操作类ProjectMgrCRUSeMet	149
6.4.1	重构的CreateProjectSeMet类	150
6.4.2	零代码的EditProjectSeMet类	150
6.5	测试项目管理TestProjectMgrSeMet	150
6.5.1	测试项目列表	151
6.5.2	查询与创建	152
6.6	登录后默认主页面——使用枚举类封装框架数据	153
6.7	第一个完整的端到端用例	154
6.7.1	测试场景	154
6.7.2	自给自足的测试用例	155
6.8	小结	156
第7章	页面封装—测试规约的自动化	158
7.1	导航栏	158
7.1.1	导航栏页面分析	158
7.1.2	元素分析与自动化实现	159
7.1.3	测试上下文	160
7.2	关键字管理	160
7.2.1	页面分析	161
7.2.2	自动化测试用例	161
7.2.3	关键字管理类的实现	162
7.2.4	文件导入关键字	164
7.3	测试规约	166
7.3.1	进入测试规约页面	166
7.3.2	测试用例集操作	166
7.3.3	创建测试集	168
7.3.4	创建测试用例	169
7.4	完整的用例	174
7.5	小结	176
第8章	测试计划	177
8.1	从主页面到测试计划管理	177
8.1.1	页面定位	177
8.1.2	静态跳转方法	178
8.2	测试计划管理	179
8.3	创建测试计划	179
8.4	测试计划对象	181
8.5	版本管理	182
8.5.1	从主页面到版本管理	182
8.5.2	从版本管理类到版本管理对象	183
8.6	添加测试用例至测试计划	186
8.6.1	发起添加测试用例	187

- 8.6.2 测试用例树操作 187
- 8.6.3 添加用例至测试计划 188
- 8.7 添加平台至测试计划 190
 - 8.7.1 发起添加平台 190
 - 8.7.2 完成平台添加 191
- 8.8 测试执行 192
 - 8.8.1 测试执行的入口 192
 - 8.8.2 测试用例集的设置 192
 - 8.8.3 执行结果录入 194
- 8.9 小结 196
- 第2部分 UI自动化框架横向拓展
- 第9章 Web兼容性测试 198
 - 9.1 浏览器简史 198
 - 9.2 内核之母 201
 - 9.3 内核之意 202
 - 9.3.1 Trident 202
 - 9.3.2 Gecko 203
 - 9.3.3 Presto 204
 - 9.3.4 WebKit 204
 - 9.3.5 Chromium & Blink 205
 - 9.4 渲染与解析 205
 - 9.5 容错处理机 210
 - 9.6 布局与绘制 210
 - 9.7 兼容性问题案例 215
 - 9.7.1 网银中使用的jQuery版本问题 215
 - 9.7.2 IE 10在响应式时间中CSS样式布局混乱 215
 - 9.7.3 CSS重构以及JS文件分割优化 216
 - 9.7.4 IE 6浏览器上页面加载的性能问题（可导致IE 6崩溃） 217
 - 9.7.5 兼容性Bug错误具体分析 223
 - 9.8 移动端浏览器兼容性 226
- 第10章 轻量级本地化测试解决方案 233
 - 10.1 什么是本地化、国际化和全球化 233
 - 10.2 本地化测试自动化的挑战 233
 - 10.3 解决思路 234
 - 10.4 伐谋：一码走天下 234
 - 10.4.1 TestLink的样例分析 234
 - 10.4.2 使用本地化中性资源进行自动化 235
 - 10.5 伐兵：使用本地化资源 238
 - 10.5.1 TestLink本地化资源文件分析 238
 - 10.5.2 本地化支持的简要流程 239
 - 10.5.3 本地化案例 240
 - 10.6 基于HSQL的数据字典 241
 - 10.6.1 为什么使用HSQL 241
 - 10.6.2 HSQLDB数据导入及维护 241
 - 10.6.3 将HSQL集成进自动化测试框架 243
 - 10.7 结合Java Annotation的解决方案 248
 - 10.7.1 思路 248
 - 10.7.2 定义Annotation 249
 - 10.7.3 定义工具类，通过Java反射机制动态获取本地化资源 249

10.8	部署	251
10.8.1	SolventTestCase	251
10.8.2	Solvent	251
10.8.3	框架类	251
10.8.4	测试用例	252
10.9	总结与展望	252
第3部分 Hybrid Automation—实现UI/API测试一体化		
第11章 欢迎来到接口测试的世界 254		
11.1	集成测试与接口测试	254
11.1.1	集成测试	254
11.1.2	接口测试与测试接口	254
11.2	XML-RPC简介	255
11.3	使用原生的TestLink XML-RPC接口	256
11.3.1	启用XML-RPC接口	256
11.3.2	获取API KEY	256
11.3.3	sayHello样例	257
11.3.4	TestLink XML-RPC简析	259
11.4	使用TestLink Java API	260
11.5	测试用例自动化生成初探	263
11.5.1	重温等价类划分	263
11.5.2	组合策略与工具	265
11.5.3	数据集与组合算法	265
11.5.4	生成全组合用例	267
11.5.5	生成健壮性用例	271
11.6	小结	272
第12章 Hybrid Automation混合自动 274		
12.1	修改SeMet层的基础类	274
12.1.1	在Webdrivers类中存储TestLinkAPI实例	274
12.1.2	在TLTestCase中完成TestLinkAPI初始化	274
12.2	在测试用例中直接使用接口	276
12.3	将接口转移到业务SeMet类中	278
12.4	用例无缝切换	280
第13章 iTestLink业务与技术解耦 281		
13.1	问题与挑战	281
13.2	Webdriver带来的启示	281
13.3	实现iTestLink	282
13.3.1	定义TestLink业务接口	282
13.3.2	定义业务对象类	283
13.3.3	增加TLSession类获取执行方式	284
13.3.4	枚举类TestLinkDriver的实现	285
13.3.5	新建TLTestCase	286
13.4	iTestLink的测试用例	286
13.5	改造APIMet支持iTestLink	288
13.5.1	引入TestLinkJavaAPIAdaptor	288
13.5.2	接口间TestProject对象转换	290
13.5.3	接口层测试用例执行	291
13.6	改造SeMet支持iTestLink	292
13.6.1	实现Service类	293
13.6.2	改造相关SeMet	294

13.6.3	实现TestLinkSeMetAdaptor	296
13.6.4	UI层测试用例执行	297
13.7	小结	298
第14章	代码的管理和审查	299
14.1	简介	299
14.2	代码管理工具	299
14.2.1	GIT介绍	299
14.2.2	安装过程	300
14.2.3	常用命令	300
14.3	代码审查工具	302
14.3.1	Gerrit是什么	302
14.3.2	如何去安装	302
14.3.3	基本功能介绍	306
14.4	检出代码	310
14.4.1	命令介绍	310
14.4.2	工作场景一	310
14.5	提交代码	312
14.5.1	命令介绍	312
14.5.2	工作场景	314
14.6	合并分支	316
14.6.1	命令介绍	316
14.6.2	工作场景	317
14.6.3	分支合并冲突	319
14.7	最佳实践	320
第15章	持续集成	322
15.1	简介	322
15.2	持续集成工具	322
15.2.1	Jenkins介绍	322
15.2.2	主要功能	323
15.2.3	如何安装	324
15.2.4	建立新的Job	324
15.2.5	注意事项	325
15.2.6	系统配置	326
15.2.7	设置Job	330
15.2.8	成功连通	331
15.2.9	工作场景	333
15.3	执行自动编译和测试	334
15.4	合并代码	335
15.5	常见问题	336
15.6	最佳实践	337
第16章	Webdriver入门简介	339
16.1	Webdriver的基本结构	339
16.2	Webdriver的工作原理	339
16.3	在启动浏览器的时候发生了什么	340
16.3.1	Firefox	340
16.3.2	Chrome	341
16.3.3	IE	341
16.4	Webdriver常用API简介	342
16.4.1	Webdriver API结构浅析	342

- 16.4.2 启动与关闭浏览器 344
- 16.4.3 提取文本getText 344
- 16.4.4 提取元素属性getAttribute 345
- 16.4.5 模拟浏览器的前后跳转forward和back 345
- 16.4.6 模拟鼠标右键Actions和contextClick 346
- 16.4.7 截图TakesScreenshot 346
- 16.4.8 操作下拉菜单Select控件 347
- 16.4.9 操作frame里面的元素switchTo 348
- 16.4.10 操作单选按钮radio button 349
- 16.4.11 操作多选框checkbox 350
- 16.4.12 切换窗口getWindowHandle 350
- 16.4.13 等待 351
- 16.4.14 JS弹出框 352
- 16.4.15 弹出框的扩展 353
- 16.5 Page Object介绍 355
 - 16.5.1 创建LoginVariables.java文件 357
 - 16.5.2 创建LoginPage文件 358
 - 16.5.3 创建GloablaFunction.java文件 359
 - 16.5.4 创建WebdriverTest.java文件 359
 - 16.5.5 创建LoginTest.java文件 360
- 16.6 选择Webdriver而不是Selenium RC 361
- 16.7 如何从Selenium RC迁移到Webdriver 362
- 16.8 Webdriver如何执行Javascript 364
- 16.9 如何定制浏览器 364
- 16.10 小结 365

《精通自动化测试框架设计》

精彩短评

1、内容安排上不算太好，且框架的代码写的很零散。

《精通自动化测试框架设计》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com