

Software
Testing

软件测试丛书

精通自动化测试 框架设计

Proficiency in Testing Automation
Framework Design

陈冬严 邵杰明 王东刚 蒋涛 编著

- 对自动化测试框架层层剖析，对构成自动化测试框架的各个元素，包括基类、日志、计时、检查点、截屏、数据处理等娓娓道来，并实现了 UI/API 的分层自动化测试，更有效地应用于敏捷开发模式下的快速、灵活的持续测试。
- 本书对不同类型、格式的测试数据（如XML、Excel文件、数据库等数据源）的处理技术做了详细交待，能够满足对不同业务数据的测试需求。
- 一个随手可得的案例（开源TestLink项目）贯穿其中，从单点功能到多点功能、功能操作的封装、页面封装、控件封装、数据封装，到业务与具体实现的隔离、分层处理等，各种自动化具体技术或技巧都很好地被呈现出来，让读者可以边学习、边实践，效果倍增。



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

Software
Testing

软件测试丛书



精通自动化测试 框架设计

Proficiency in Testing Automation
Framework Design

陈冬严 邵杰明 王东刚 蒋涛 编著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

精通自动化测试框架设计 / 陈冬严等编著. — 北京:
人民邮电出版社, 2016.4
ISBN 978-7-115-41565-3

I. ①精… II. ①陈… III. ①软件工具—测试 IV.
①TP311.56

中国版本图书馆CIP数据核字(2016)第036830号

内 容 提 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。

-
- ◆ 编 著 陈冬严 邵杰明 王东刚 蒋 涛
责任编辑 张 涛
责任印制 张佳莹 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 24.25
字数: 464千字 2016年4月第1版
印数: 1—2500册 2016年4月北京第1次印刷
-

定价: 59.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第8052号

前言

在凌晨熬夜完成年度双 11 采购付款任务的同时，本书也将最终交付给出版社了。不知不觉间，互联网公司从一个“搅局者”成为市场的主流玩家，影响着数亿人的生活。

而互联网公司背后的软件开发模式也逐步成为行业的主流声音。伴随着敏捷时代的到来，互联网公司更催生出了测试开发工程师以及测试架构师这些新兴的职业。而传统的测试人员则面临着转型的挑战，职业生涯也更早地遇到了天花板。许多公司甚至都不再设立“测试经理”的职位，招聘人员也开始秉承着“不会做开发的测试不是一个好测试”的理念去遴选候选人，造成了很多测试人员将“转做开发”而不是“做好测试”作为自身职业发展的一个目标。另外一个方面，一般的自动化测试人员，在熟悉了 Webdriver 等自动化测试工具之后，如何从一个工具的使用者，仅仅完成测试步骤的自动化翻译工作，转变成为一个工具、框架或者平台的构建者、维护者，成为使用者们的支持人员，实现自我价值的提升与组织收益的共赢？这需要从业人员有很好的业务理解能力、新技术学习能力以及持续不断地自我激励与坚持。此外，还需要一个好的机遇，能得到项目实践的机会。而能在实际工作中得到这种系统锻炼，快速从“菜鸟”成长为“老法师”的无疑都是幸运儿。

本书希望凭借着作者们多年的自动化测试实践，将自动化测试可能的一些挑战、问题与处理的方式以及方法，通过选择合适的被测应用作为案例一步步进行展示的形式，介绍自动化测试框架的搭建以及后续演变过程。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，糅合进适当的工具，来尝试建设适合自身项目实际的自动化测试框架，并进而走出使用者的大门，并在踏上通往更高层级的台阶中走得更好。

本书的主要内容

第 1 部分包括第 1 章到第 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。本着兵马未动，粮草先行的策略，在第 2 章中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 章开始着手介绍如何分层构建 Solvent、SeMet 等底层框架。在第 3 章有关 Solvent 的构建中，提供了诸如用例基类、日志、计时、检查点、截屏、数据处理等基础功能。在第 4 章有关 SeMet 的构建中，结合自动化测试工具 WebDriver 以及被测应用 TestLink，介绍了 SeMet、TLTestCase 等类的实现。

第 5 章到第 8 章则是有关 TestLink UI 自动化测试框架的构建。在第 5 章中，笔者选取了 TestLink 的部分典型控件进行了封装，并提出了聚合而不是继承的 UI 页面切分原则。在第 6 章中，笔者引入了三段式页面封装法，并通过案例介绍了业务对象类、方法链、业务与具体实现的隔离、使用枚举类封装页面数据以及构建自给自足的测试用例等一些实用的方法。另外，在第 6 章到第 8 章中，分别介绍了测试项目管理、测试规约、关键字、测试用例集、测试计划、版本管理、平台、测试执行等单业务对象的 UI 自动化实现。在第 8 章中，还介绍了多业务对象之间进行交互的测试场景下的自动化实现。

本书第 2 部分则是有关 UI 自动化测试框架横向拓展的主题。

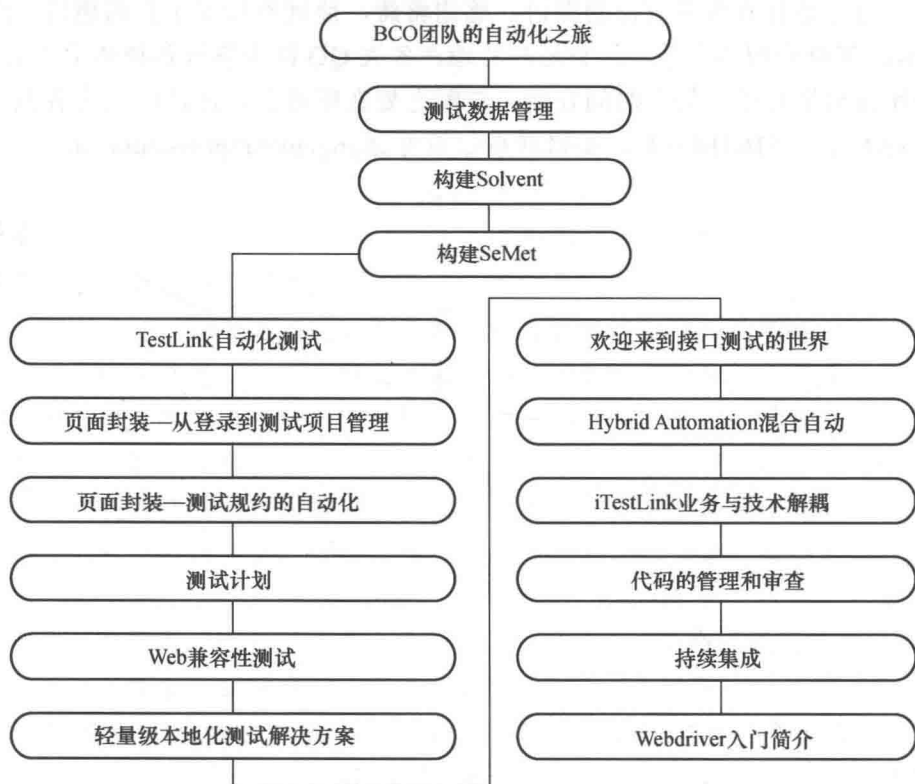
第 9 章关注浏览器兼容性问题，对浏览器端 Web App 的渲染、容错、布局等内容进行了介绍，并列举了许多兼容性测试案例。

在第 10 章中介绍了本地化测试的自动化课题。在提出了“一码走天下”的愿景之后，也根据实际需要介绍了一种利用轻量级数据库来存储数据字典，结合 Java 反射机制，完成页面定位信息动态生成的方式。

本书第 3 部分将引入 API 测试，提出多层次自动化测试的课题。

在第 12 章中介绍了如何使用第三方提供的 TestLink Java API 通过 XML-RPC 协议操作 TestLink，并简要介绍了一种接口测试用例自动化生成的方法。在第 13 章中，介绍如何在 SeMet 层混合使用接口服务，实现局部的跨层混合自动化，在一些数据和结果验证的场景中通过用例无缝切换，实现速度和鲁棒性的大幅提升。在随后的章节中，引入了与 SeMet 层并列的 APIMet 层，从而实现了 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。在第 14 章中，围绕代码管理和审查的课题，结合代码检出、提交、合并等日常工作场景，介绍了 Git/Gerrit 代码工具的使用。在第 15 章中，结合 Jenkins 这一持续集成工具，介绍了持续集成的各项工作、常见问题和最佳实践。第 16 章介绍了 Webdriver 的基础概念以及简单元素封装、自定义 Page Object 等内容，供初次接触 Webdriver 自动化测试的读者阅读参考。

本书的总体结构如下图所示。



致谢

感谢朱少民老师欣然应允为本书写序，相信很多读者是读着朱老师的书入行的。感谢人民邮电出版社的编辑张涛老师。没有他的热心指导，本书无法从一个想法落地实施，并最终与读者见面。

最需要感谢的是我们的家人。没有他们在过去一年中默默的付出和谅解，以及不断的鼓励，本书是无法写成的。

还要感谢前同事王哲、牛锐，本书很多的内容都源自于和他们之前探讨的话题，与他们的一次次午餐闲聊最终促成了本书早期提纲的形成。感谢 SPR Tracker 的作者方建新对本书第1章初稿的评阅。

感谢我的合作伙伴们。依稀记得当年在杭州学院路的办公室，王东刚每周忍着烟瘾编写程序的神情。后来才知道他编著了《软件测试与 Junit 实践》一书，是 Junit 在国内的早期传播者与践行者之一。这次，他又同意将有关兼容性等最近的研究成果拿出来与大家分享。感谢邵杰明，承担了有关测试项目工程管理部分的撰写工作，在我负责撰写

的章节中，也与他有着很多有益的探讨。感谢蒋涛，他欣然接受了我的邀请，撰写了有关 Webdriver 基础介绍的章节，而且还热心地在各大 QQ 群中解答各种初学者的问题。

由于作者水平有限，加之时间仓促，疏漏之处在所难免，还请广大读者批评指正。本书答疑 QQ 群：(516414684)，编辑联系邮箱为 zhangtao@ptpress.com.cn。

陈冬严
于上海

序

2015 年国庆节长假最后一天，打开 To Do List，里面显示还有一项工作没有完成，那就是应本书作者的邀请为本书写序，节前已经承诺，我会在“十一”假期期间完成这一任务。但假期快要结束了，还没有开始写，能完成吗？如果有一个机器人，我只要把想表达的主题、关键字（主要想法）告诉它，接着它就能很快帮我完成写作。最后，我只要校对一遍，稍作修改，就大功告成，那该多好啊！这样的梦想可能在不久的将来会成为现实，因为半个多月前，新闻写作机器人 Dream writer 就已经出现，它只花了 1 分钟，就炮制出了一篇新闻稿。但“序”的写作还属于高难度动作，Dream writer 目前还干不了。由此想到软件测试，在软件发布前，留给回归测试的时间一般也很少，人们也希望有机器人助自己一臂之力，能够快速自动地执行测试。为此，有些著名的测试工具乐意取名为 Robotium、Robot Framework，和机器人（Robot）扯上关系。在敏捷开发模式流行的今天，没有自动化就无法“敏捷”起来；没有自动化测试就无法做到持续集成，没有持续集成就不能做到持续交付。所以，人们一直致力于提高自动化测试水平，例如，在移动 App 的安装/卸载测试、兼容性测试、基本功能测试、“闪退”测试等方面，人们已经享受到自动化测试的云服务，需要什么测试就能获得相应的服务，甚至可以做到一键式测试，让测试自动化达到极致。

但是，当我们面对企业的软件应用产品时，其业务逻辑复杂，用户交互（UI）界面也不简单，仅仅完成基本功能的验证是不够的、也是很难的，而且需要针对各种业务场景、业务流程、业务数据等进行验证。这时，需要我们针对 UI、API 开发大量的自动化测试脚本，自动化测试会面对更大的挑战。虽然针对 API 进行自动化测试的效率高、执行稳定，维护成本也低，但没有 UI 的自动化测试配合，测试还不够彻底，还需要手工执行部分 UI 测试。所以，UI 自动化测试再难，也必须克服这个困难，才能比较彻底地实现自动化测试。

本书就是从 UI 自动化测试入手，逐步展开各项自动化测试技术的讨论，包括面向接口的自动化测试、数据驱动的自动化脚本、兼容性测试和本地化测试的自动化实现等。全书文字流畅、幽默，案例丰富，技术也很有深度，亮点也不少，例如：

- 对自动化测试框架层层剖析，对构成自动化测试框架的各个元素，包括基类、日志、计时、检查点、截屏、数据处理，娓娓道来，并实现了 UI/API 的分层自动化测试，更有效地应用于敏捷开发模式下的快速、灵活的持续测试。

- 多数业务测试是数据驱动的，本书对不同类型、格式的测试数据（如 XML、Excel 文件、数据库等数据源）的处理技术做了详细交待，能够满足不同业务数据的测试需求。

- 一个随手可得的案例（开源 TestLink 项目）贯穿其中，从单点功能到多点功能、功能操作的封装、页面封装、控件封装、数据封装，到业务与具体实现的隔离、分层处理等，各种自动化具体技术或技巧都很好地被呈现出来，让读者可以一面学习、一面实践，效果倍增。

- 本书构建的自动化测试框架是从底层 Solvent、SeMet 开始，一方面是基于 Selenium+Webdriver 实现，基础可靠；另一方面，又能脱离 Selenium+Webdriver，为 API 等其他不同技术实现的自动化测试服务。我们知道，Selenium 是众多开源测试工具的佼佼者，2006 年我就向自己的团队介绍 Selenium 并开始试用，将近 10 年过去了，Selenium 依旧光彩照人，在与 Webdriver 结合之后，其能力和影响更大了，基于 Selenium+Webdriver 实现的自动化测试框架值得我们信赖。

本书是作者多年的自动化测试实践经验的积累，且不是停留在工具应用上，而是作为自动化测试框架的创建者、开发者来分享经验的。所以，本书不仅适合刚入门的测试人员阅读，也适合资深测试工程师阅读，在阅读时，会有思想火花的碰撞、会从中得到启发，能从中学到不少东西。本书也不仅适合测试工程师阅读，而且适合开发人员阅读，这正和目前开发的敏捷潮流吻合，未来的测试工具开发或维护工作，可能更多由开发人员来承担起来。希望本书的出版，给软件测试图书世界带来一丝清新的气流，让更多的读者能呼吸到这股新鲜的空气，舒畅快乐！

朱少民

同济大学软件学院

作者介绍

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。

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