

Selenium 在电话自动拨测系统中的应用

马天鸣¹, 梁琛²

(1. 上海工程技术大学 电子电气工程学院, 上海 201620;
2. 电信科学技术第一研究所有限公司, 上海 200032)

摘要:由于电话拨测系统存在大量的省市端局、端局用户信息和数以千计的拨测号码,以手工的方式做重复的功能测试会增加测试时间和工作成本,为了减少繁琐冗余的测试工作量,提高测试效率,现使用开源 web 自动化工具 Selenium 对电话自动拨测系统进行回归测试。对于电话自动拨测系统中端局管理等三个主要的模块,选择部分适合自动化测试的功能点来设计测试用例,以 python 为开发语言,unittest 为单元测试框架,编写测试代码并进行优化,在执行脚本后,能生成完整、易于分析和定位缺陷的测试报告并自动以电子邮件的形式发送给测试和开发人员,便于项目相关人员能及时获取系统的缺陷情况。该测试脚本可以每天不间断执行,不仅节省了测试的时间,而且提高了测试的准确性和有效性。

关键词:web 自动化; Selenium; 回归测试; python; unittest

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2021)10-0141-05

doi: 10.3969/j.issn.1673-629X.2021.10.024

Application of Selenium in Telephone Automatic Dial Testing System

MA Tian-ming¹, LIANG Chen²

(1. School of Electronic & Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China;
2. The First Research Institute of Telecommunications Technology Co., Ltd., Shanghai 200032, China)

Abstract: Due to the large number of provincial and municipal end office user information and thousands of dialing and testing numbers in the telephone dialing test system, repeated functional testing by hand will increase the test time and work cost. In order to reduce the tedious and redundant test workload and improve the test efficiency, Selenium, an open-source web automation tool, is used to conduct regression test of the automatic telephone dialing test system. For the three main modules of the telephone automatic dial test system, such as the end office management, select some function points suitable for automatic test to design test cases. With python as the development language and unittest as the unit test framework, the test code is written and optimized. After the script is executed, complete test reports that are easy to analyze and locate defects can be generated and automatically sent to the testers and developers in the form of E-mail, so that the project personnel can timely obtain the defect situation of the system. The test script can be executed continuously every day, which not only saves the test time, but also improves the accuracy and effectiveness of the test.

Key words: web automation; Selenium; regression test; python; unittest

0 引言

在项目的开发过程中,新版本的连续发布使回归测试的需求更为频繁,在项目周期长,模块功能复杂的情况下,采用手工方式对系统功能进行多次重复的验证时,会消耗大量时间,增加测试成本。因此,选择正确的回归测试策略来改进测试的效率和有效性是很有必要的。使用自动化测试^[1-3]的方式来进行回归测试,可以进行大规模、长时间和多周期的测试,既减少测试人员的工作量,又保证了测试的效率和准确性。

为了实现全国各局之间、局与各电信运营商网络

之间的通信畅通,实时监测全网通断情况,及时、准确定位故障线路,对全网设备、资源优化提供科学依据等运营目标,迫切需要建设一套覆盖全网的电话自动拨测系统。

根据电话自动拨测系统需求变动不频繁、项目周期长以及系统界面稳定、变动少的特点,引入 UI 自动化测试来达到扩大测试覆盖率,降低回归测试成本的效果。

Selenium^[4-8]是一个主要用于 web 应用程序自动化测试^[9]的工具集合,最初是为网站自动化测试而开

收稿日期:2020-10-28

修回日期:2021-03-02

基金项目:国家自然科学基金资助项目(61601296);上海工程技术大学人才行动计划资助项目(2018RC43)

作者简介:马天鸣(1983-),男,副教授,硕导,博士,研究方向为计算机移动通信。

发的^[10],它支持各种主流浏览器和多种编程语言,可以跨平台使用。该工具能模拟用户操作,在浏览器中直接运行,拥有足够好的用户体验^[11],其开源优势以及功能特点符合电话自动拨测系统对于自动化测试工具的要求。

1 Selenium 自动化测试的相关工作

1.1 电话自动拨测系统的介绍

电话自动拨测系统是 B/S(浏览器/服务器)架构的系统,由统计分析、端局状态、端局管理、号码管理、拨测管理和系统管理六个模块组成,如图 1 所示。

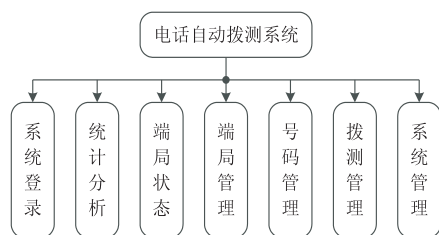


图 1 电话自动拨测系统的组成

其中,登录是进入系统的前提,端局管理和号码管理两个模块需要对大量的省市端局和号码进行增加、删除、修改、关联等操作,系统管理则需要添加各端局的登录用户、配置对应的权限,步骤重复繁琐,仅以手工的方式对这部分功能进行测试会消耗大量的时间和精力,增加测试人员的负担。故需要引入适合 web 应用程序的自动化工具对系统进行回归测试,从而提高测试效率。

1.2 Selenium 概述

Selenium 是一种基于浏览器的自动化测试工具,提供了一种跨平台、跨浏览器的端到端 web 自动化解决方案。Selenium 支持主流的浏览器,包括 Firefox、IE、Chrome、Safari 和 Opera 等,它可以模拟使用者与浏览器之间的交互,如访问网站、点击链接、填写表单、浏览网页等。Selenium 前后共经历了三个版本,分别是 Selenium1.0、Selenium2.0 和 Selenium3.0,现使用最新版本 3.0 来做测试,Selenium3.0 主要包括三个部分: Selenium IDE、Selenium WebDriver 和 Selenium Grid。

Selenium IDE 作为浏览器 Mozilla Firefox 的插件,不需要任何编程逻辑来编写测试脚本,它通过与浏览器的交互可以进行录制和回放,并能把录制的操作以多种编程语言的形式导出测试用例。

Selenium WebDriver 提供了 web 自动化所需的 API(应用程序接口),主要用作浏览器控制、页面元素选择和调试。使用 Selenium WebDriver 前要选择一种语言来编写自动化脚本且需要由 selenium client library 支持。

Selenium Grid 提供了在不同浏览器上运行

Selenium 的能力,它的作用是分布式执行测试,可以控制多台机器以及多个浏览器执行测试用例。在测试用例较多的情况下,通过 Selenium Grid 可以提高测试效率。

1.3 python 的优势

python 是一门被广泛应用的高级编程语言,它语法简单、简洁易懂、框架复用性高、可移植性强,并且提供了大量的内置库、函数以及用户编写的第三方库,众多开源的科学计算软件包都提供了 python 的调用接口^[12],从而可以容易地实现一些复杂的功能。python 支持 Selenium client library,既可以使用 C/C++ 进行扩展,又能嵌入其他编程语言中,故选择 python 来编写自动化测试脚本。

1.4 Selenium 自动化测试流程

Selenium 的测试过程分为六个步骤,分别是制定测试计划、编写测试脚本、优化测试脚本、执行测试脚本、生成测试报告和邮件发送报告,如图 2 所示。

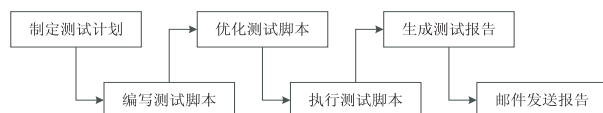


图 2 Selenium 自动化测试过程

1.4.1 制定测试计划

自动化项目的测试计划做得越详细,后期就越能按部就班地去执行,测试的成功率也就越高。对于系统而言,并非所有的功能点都适合使用自动化来进行回归测试,因此需要罗列所有符合 Selenium 测试条件的模块并对其功能点进行细化来设计测试用例,然后再实施后面五个步骤。

1.4.2 编写测试脚本

单元测试框架是编写自动化测试用例的基础^[4]。unittest 一般称为 PyUnit,是从 Java 程序中广泛应用的 JUnit 启发而来的,可以使用 unittest 为任何项目创建全面的测试套件,unittest 也是 python 中用来测试各种标准类库模块的^[13]。现使用 unittest 单元测试框架来创建基于 python 的 Selenium WebDriver 测试脚本。unittest 适用于 web 自动化测试用例的开发和执行,它可以用来组织和执行系统各模块的测试用例。

1.4.3 优化测试脚本

编写的脚本只包含元素定位和页面操作,没有校验测试用例中预期结果和程序返回的实际结果是否一致,因此需要对脚本进行优化,通过断言的方式来判定测试用例执行成功与否,如等价校验、逻辑校验和异常校验等。

1.4.4 执行测试脚本

测试脚本是以 py 为后缀的文件,在代码编辑器中运行当前文件,脚本便自动开始执行,Chrome 浏览器

立即弹出并按测试用例中的步骤依次执行,在结束时浏览器的窗口会自动进行关闭。

1.4.5 生成测试报告

使用 unittest 的拓展 HTMLTestRunner 生成格式友好的 HTML 测试报告,既能获取测试结果的概况,又能深入查看报告的细节。

1.4.6 邮件发送报告

以邮件附件的形式自动发送测试报告至相关技术人员的邮箱,这样能让他们及时了解系统缺陷的详细信息,有助于其后续工作的安排和开展。

2 Selenium 测试的实例分析

2.1 制定测试计划

电话自动拨测系统中的系统管理、端局管理和号码管理分别牵涉到大量的端局用户、省市端局和拨测号码,用手动测试的方法进行重复的操作会消耗不少时间,而且降低了测试人员对于工作的积极性,使用 Selenium 工具对这三个模块中部分适合自动化测试的功能点进行回归测试,比如对拨测号码以及端局用户信息的新增,删除和修改等,这样可以减轻重复操作的负担,从而提高测试人员的工作积极性。

系统登录虽然不在系统管理、端局管理和号码管理三个模块的范围内,但由于操作任何一个模块都无法逾越这个步骤,因此该部分也纳入了自动化测试的范围。

2.2 编写测试脚本

2.2.1 系统登陆验证码

登录电话自动拨测系统的前提是输入正确的用户名、密码和由四个随机数字组成的验证码,用户名和密码是固定的,验证码则是随机生成的,需要正确识别才能进行后续操作,系统登录框如图 3 所示。



图 3 登录框的显示

主要有四个方法来解决通过验证码的问题:(1)去掉验证码部分的代码;(2)使用一个万能验证码;(3)通过 cookie 直接登录系统,从而避开验证码这一步骤;(4)识别验证码中的数字并登录系统。第一种

方法去掉验证环节会给系统带来安全风险,第二种方法则需要通过开发的协助才能完成,第三种方法不能对验证码进行自动化测试,因此只有最后一种才是安全,可行的方法。

要识别验证码首先要对数字区域进行截取,webdriver 能对浏览器页面进行截图,但是整个浏览器页面过大,需要具体定位到验证码图片的所在位置。通过验证码的标签 id 定位到该标签所在页面的像素位置,再获取标签大小,确定图片的宽和高,从而获得验证码的位置和大小。在截取验证码图片后,使之转化成灰度图并增强它的饱和度,此时保存后的图片如图 4 所示。



图 4 处理后的验证码图片

从图 4 可以看出,此时验证码的辨识度比较高,通过把图片中的内容转化为字符串的操作,成功地辨识出验证码中的数字,使得系统登录的整个过程得以自动完成。

2.2.2 单元测试框架

单元测试是一项对技术要求很高的工作,只有白盒测试人员和软件开发人员才能胜任,但用单元测试框架做单元测试却十分简单,而且单元测试框架不仅可以用来做单元测试,还适用于不同类型的自动化测试^[14]。

单元测试框架是构建自动化测试用例的基础,在 python 中有多种单元测试框架,如 unittest、doctest、pytest 和 nose 等,电话自动拨测系统使用的是 unittest,它可以为任何项目创建全面的测试套件,主要由 Test Case、Test Suite、Test Runner 和 Test Fixture 四个核心组件构成。在执行 Test Suite(测试套件)后,unittest 在命令行会输出对应的测试结果,使用 unittest 的拓展 HTMLTestRunner 可以进一步生成 Test Report(测试报告),如图 5 所示。

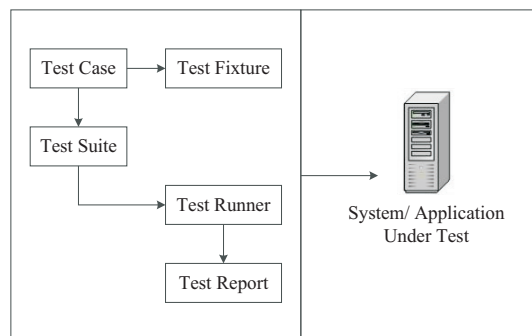


图 5 unittest 单元测试框架

(1)Test Case^[15-18](测试用例)。一个测试用例是在 unittest 中执行测试的最小单元,它通过 assert 方法来验证一组特定的操作和输入以后得到的具体响应。

unittest 提供了 TestCase 基类,可以用来创建测试用例。

(2)Test Suite(测试套件)。一个测试套件是多个测试或测试用例的集合,是针对被测程序的对应的功能和模块创建的一组测试,一个测试套件内的测试用例将一起执行。

(3)Test Runner(测试执行器)。测试执行器负责测试执行调度并且生成测试结果给用户。测试执行器可以使用图形界面、文本界面或者特定的返回值来展示测试执行结果。

(4)Test Fixture(测试夹具)。通过使用测试夹具,可以定义在单个或多个测试执行之前的准备工作和测试执行之后的清理工作。

(5)Test Report(测试报告)。测试报告用来展示所有执行用例的成功或者失败状态的汇总,执行失败的测试步骤的预期结果与实际结果,还有整体运行状况和运行时间的汇总。

2.2.3 页面定位元素

页面元素都是由 HTML 代码组成的,它们之间有层级地组织起来,每个元素有不同的标签名和属性值,WebDriver 就是根据这些信息来定位元素的。Selenium 主要有 id、name、tag、class、link_text、partial link、XPath、CSS_selector 八种元素定位的方法,这些方法根据一定的标准取查找元素,如果元素被正常定位,那么 WebElement 实例返回,否则抛出 NoSuchElementException 的异常。

电话自动拨测系统主要使用 id、name 和 class 属性来查找元素,这三种方式是最为普遍和快捷的,当上述方法无法生效时,使用 XPath 的方式可以灵活地运用绝对或相对路径来进行定位。

基于元素 id 的定位方法是通过查找 Web 页面上的元素 id 值来获取元素,比较简单,适合 Web 元素 id 固定的情况来使用,对于动态分配 Web 元素 id 的情况不能使用^[11]。由于页面中元素的 id 是唯一的,因此通过 id 定位元素是最佳方法。

基于元素 name 的定位方法是另一种常见的查找元素的方式,可以通过匹配 name 的值来定位一个或者一组元素,如果匹配成功则返回查找的元素,匹配失败则抛出异常。

基于元素 class 的定位方法用来关联 css 中定义的属性,class 可以指定元素的类名,其用法和 id、name 相类似。

基于元素 XPath 的定位方法是通过元素的路径来完成对元素的查找,XPath 是 XML 路径语言,可以用来定位 XML 文档中元素的位置。Xpath 分两种定位方法,一种是用绝对/相对路径来定位,另一种是使用

元素属性来进行定位。

2.2.4 操作页面元素

WebDriver 提供很多用来与浏览器交互的功能和设置,可以通过使用 WebDriver 的功能和方法来实现与浏览器窗口、警告、框架和弹出窗口的交互。自动化过程中与页面元素进行交互的方法都是由 WebElement 接口提供的,页面元素包含文本框、文本域、按钮、单选框、多选框和下拉列表等。

WebElement 提供了一些功能和方法来实现与网页元素的交互,WebElement 的功能如 size(获取元素大小)、tag_name(获取元素的标签名称)和 text(获取元素的文本值)等,方法如 clear(清除文本)、send_keys(*value)模拟按键输入和 click()(单击元素)等。WebElement 与各种 HTML 控件的自动化交互,例如在输入框内写入文本、单击一个按钮和选择单\复选框等。

2.2.5 页面元素等待

当没有定位到标签元素时,除了找错标签这个原因以外,还有一种可能就是标签由于网速等因素没有及时加载出来,直接获取会导致程序运行时报错,因此需要设置线程等待,Selenium 提供了两种等待机制:隐式等待和显式等待。

隐式等待对于解决网络延迟或者利用 Ajax 动态加载元素所导致的程序响应时间不一致,是非常有效的,WebDriver 类提供了 implicitly_wait()方法来设置超时时间;显式等待可以只作用于仅有同步需求的测试用例,WebDriver 提供了 WebDriverWait 类和 expected_conditions 类来实现显式等待^[13]。

在处理同步问题上,显式等待的可操控性更佳,处理方式更灵活,故在电话自动拨测系统脚本的编写中使用显式等待的机制。

2.2.6 控制浏览器

浏览器的工具栏中一般都有前进、后退以及刷新的按钮,这些功能比较实用,在网页操作中经常会用到,Selenium WebDriver 提供了这些操作浏览器的方法。

浏览器全屏模式可以用 maximize_window()的方法来实现,该方法无需设置参数;当想要让浏览器以某种特定的大小运行时,可以用 set_window_size()的方法来设置浏览器的窗口大小。

WebDriver 提供了 forward(),back()和 refresh()来模拟浏览器前进、后退以及刷新/重新加载的按键功能。

2.3 优化测试脚本

测试脚本中验证实际和预期结果是否一致是很重要的,unittest 的 TestCase 类提供了很多实用的方法来

校验预期结果与程序返回的实际结果是否一致,如 `assertEqual/assertNotEqual`、`assertTrue/assertFalse`、`assertIn/assertNotIn` 等,这些方法要求必须满足某些条件才能继续执行接下来的测试。如果给定的断言通过了,下面的测试代码则被执行,否则将会导致测试立即停止并且给出异常信息。

2.4 生成测试脚本

执行测试套件后, `unittest` 在命令行输出测试结果,但是给相关技术人员发送命令行日志并不是一个明智的选择,生成一个包含所有用例执行情况的测试报告是很有必要的。 `unittest` 没有相应的内置模块能创建格式友好的测试报告,可以通过使用 `unittest` 的拓展 `HTMLTestRunner` 来实现,脚本生成的测试报告如图 6 所示。

Test Group/Test case	Count	Pass	Fail	Error	View
usermangertest.UserMangerTest	3	3	0	0	Detail
rolermangertest.RoleMangerTest	3	3	0	0	Detail
test_add_and_delete_role				pass	
test_adjust_authority_select_and_unselect				pass	
test_basic_operation_part				pass	
departmentmangertest.DepartmentMangerTest	3	3	0	0	Detail
numberrangertest.NumberMangerTest	4	3	1	0	Detail
test_add_number_function				pass	
test_delete_number_function				pass	
test_search_number				pass	
test_select_status				fail	
numbergroupmangertest.NumberGroupMangerTest	2	2	0	0	Detail
agentmangertest.AgentMangerTest	6	6	0	0	Detail
Total	21	20	1	0	

```

File: "python(testid)_test_system(numbrermangertest.py", line 96, in test_select_status
self.assertEqual(1, len(operator.optimz))
AssertionError: 4 != 0
Traceback (most recent call last):
File: "python(testid)_test_system(numbrermangertest.py", line 96, in test_select_status
self.assertEqual(1, len(operator.optimz))
AssertionError: 4 != 0
  
```

图 6 HTML 测试报告

测试报告为 HTML 文件,可以用浏览器进行读取,它包含测试开始时间、持续时长、状态统计以及每个测试用例执行结果的详细描述。图 6 为系统管理、端局管理和号码管理三个模块选取部分用例所生成的测试报告,其中执行失败的用例可以通过展开 `Detail` 来对问题进行具体的定位和分析。

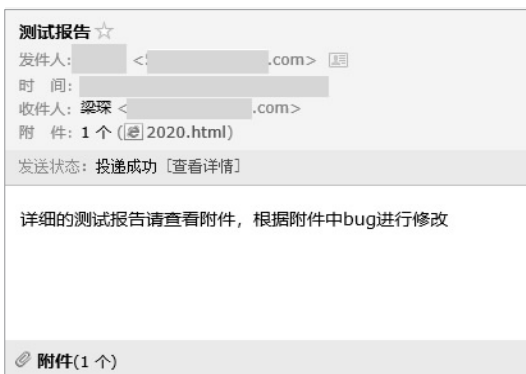


图 7 邮件自动发送测试报告

2.5 邮件自动发送测试报告

以邮件的形式自动发送测试报告是电话自动拨测系统的重要需求之一,当测试脚本执行完成以后,便会

自动向项目相关人员的邮箱发送测试报告。 `yagmail` 是 `python` 的一个第三方库,通过使用 `yagmail` 能以比较简单的方法来实现自动发送邮件的功能。在发送邮件时,除了填写主题和正文外,还可以增加抄送人和添加附件等。

由于 `HTMLTestRunner` 报告在展示时引用了 `Bootstrap` 样式库,当作为正文写在邮件中时,会导致样式丢失,所以作为附件发送更为合适,如图 7 所示。

3 结束语

为减少手工测试工作量,提高测试效率,使用 `Selenium` 自动化测试工具对拨测系统中三个主要模块进行回归测试。测试结果表明,该方法可以模拟手工操作反复地执行并生成直观、易于定位 `bug` 的 HTML 测试报告,最后以邮件附件形式每天定时发送给相关技术人员。相比手工测试,此方法节省了重复执行测试用例的时间,提高了测试效率,为开发能尽早修复缺陷,相关人员能及时了解测试情况提供了便利。

参考文献:

- [1] CAREY D R, ROMANIA J. Automated test systems test program software—making the invisible visible[J]. IEEE Instrumentation & Measurement Magazine, 2020, 23(5): 11–18.
- [2] HE W, ZHAO R, ZHU Q. Integrating evolutionary testing with reinforcement learning for automated test generation of object-oriented software[J]. Chinese Journal of Electronics, 2015, 24(1): 38–45.
- [3] ANGIUS N. Model-based abductive reasoning in automated software testing[J]. Logic Journal of the IGPL, 2013, 21(6): 931–942.
- [4] 吴晓华. Selenium WebDriver 实战宝典[M]. 北京: 电子工业出版社, 2015: 49–64.
- [5] 龚智勇. 基于 Selenium 的 OpenStack Horizon 自动化测试的实现[J]. 国外电子测量技术, 2017, 36(5): 45–49.
- [6] 叶鹏. 基于 CTI-TET 和 Selenium Webdriver 的 Web 应用自动化测试框架的设计与实现[J]. 软件工程, 2016, 19(8): 9–13.
- [7] BRUNS A, KORNSTADT A, WICHMANN D. Web application tests with selenium[J]. IEEE Software, 2009, 26(5): 88–91.
- [8] MUSTOFA K, FAJAR S P. Selenium-based multithreading functional testing[J]. Indonesian Journal of Computing and Cybernetics Systems, 2018, 12(1): 63–72.
- [9] STOCCO A, LEOTTA M, RICCA F, et al. APOGEN: automatic page object generator for web testing[J]. Software Quality Journal, 2017, 25(3): 1007–1039.
- [10] 黄孝伦, 王东. 以 Selenium+Chrome 为核心的数据采集系统设计[J]. 计算机技术与发展, 2020, 30(9): 216–220.