

基于 Selenium 的 Web 软件自动化测试

姜 文,刘立康

(西安电子科技大学 通信工程学院,陕西 西安 710071)

摘 要:随着互联网技术和云计算技术的发展,很多软件产品都通过 Web 页面来实现客户端操作。Web 软件的开发与测试已经成为软件开发过程中的重要组成部分。在常见的 Web 自动化测试工具中,Selenium 是一款开源的、表现非常优秀的自动化测试工具。Selenium 支持多种类型浏览器,可在多种操作系统上运行,支持多种语言的测试脚本。结合 Web 软件项目自动化测试工作实践,介绍了 Web 软件的开发技术和软件功能测试点;叙述了测试软件开发过程;详细叙述了基于 Selenium 与 Python 的自动化测试软件开发,其主要内容包括搭建测试环境、编写测试脚本、基于测试库框架优化测试脚本、自动化脚本连跑。最后介绍了测试结果分析。工作实践表明,采用 Selenium 进行 Web 软件测试,提高了测试效率,减少了测试工程师大量的重复测试验证工作。

关键词:Web 软件;自动化测试;测试脚本;Selenium;Python

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2018)09-0047-06

doi:10.3969/j.issn.1673-629X.2018.09.011

Automation Testing of Web Software Based on Selenium

JIANG Wen,LIU Li-kang

(School of Telecommunication Engineering,Xidian University,Xi'an 710071,China)

Abstract:With the development of Internet technology and cloud computing technology,a lot of software products have achieved client operation through the Web pages. Web software development and testing has become an important part of the software development process. In the common Web test automation tools,Selenium is an open source automation tool with excellent performances,which supports a variety of browsers,runs on multiple operating systems and supports test scripts in multiple languages. In combination with practice of Web software test automation project,we introduce the development of Web software technology and software function test points,and describe the testing software development process. In addition,we describe the test automation software development based on Selenium and Python in detail,including setting up test environment,writing test scripts,optimizing test scripts based on the test library framework and running automation scripts. Finally we introduce the test result analysis. Practice shows that using Selenium for Web software test can improve testing efficiency and reduce a lot of repetition test validation for engineers.

Key words:Web software;automation testing;test scripts;Selenium;Python

0 引 言

随着互联网技术和云计算技术的发展,很多软件产品都通过 Web 页面进行操作。Web 页面具有操作简单、操作场景繁琐等特点;之前 Web 页面测试大都使用手动的方法来进行测试,测试过程比较耗费人力,技术含量也较低。随着软件技术的发展,软件测试也逐渐从纯手动测试过渡到手动测试与自动化测试并存。

近年来,软件自动化测试技术发展很快,出现了很

多成熟的自动化测试工具,常见的 Web 自动化测试工具有:QTP、Silk Test、Selenium 等。其中 Selenium^[1-5]是一款开源的、表现非常优秀的自动化测试工具。它最大的特点就是可以完全模拟用户在浏览器上进行各种 Web 操作,拥有足够好的用户体验,是站在用户的角度对网站性能进行测试。Web 自动化测试工具 Selenium 的逐渐风行,为原本通过手动方式进行 Web 页面测试,提供了相对简单、廉价的自动化方法。

收稿日期:2017-11-18

修回日期:2018-03-22

网络出版时间:2018-05-16

基金项目:国家部委基础科研计划;国防预研基金项目(A1120110007)

作者简介:姜 文(1986-),女,工程师,硕士,CCF 会员(E200032324M),研究方向为图像处理与分析、文字信息分析处理、数据库应用和软件工程;刘立康,副教授,研究方向为数字通信、图像传输与处理、软件工程。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.TP.20180515.1702.080.html>

1 Selenium 与 Python

文中使用 Selenium 和 Python 工具开展 Web 软件自动化测试^[6-10],以下简单介绍这两个工具。

1.1 Selenium

Selenium 是一款开源的 Web 应用程序测试工具。Selenium 测试脚本直接运行在浏览器中,模拟用户操作网页的过程。支持在 IE、Chrome 和 FireFox 等通用浏览器上使用,可以在 Windows、Linux、Mac 和 Solaris 等多种操作系统上运行。

目前广泛使用的是 Selenium2.0 以上版本,Selenium2 = Selenium1 + WebDriver。

WebDriver 是一款自动化测试 Web 应用程序的工具,提供友好的 API,使测试更易于理解和维护。它不依赖于任何特定的测试框架,是完全独立的,应用它不需要启动任何额外的进程或者运行其他程序。

Selenium 2^[11-13](Selenium WebDriver)是将 WebDriver API 集成到 Selenium 1,解决提供替代编程接口所带来的局限性。Selenium 2 工具的主要功能包括:对 Web 页面进行功能测试,支持脚本录制,支持 Java、Python、Ruby 和 C#等语言的测试脚本。

1.2 Python

Python 是一款面向对象的脚本语言,语法比较简单但功能却很强大。Python 语言易读、易维护,受广大用户所欢迎、用途十分广泛。

Unittest 也称为 PyUnit^[14-16],是 Python 语言的单元测试框架。自 Python2.1 版本后,Unittest 已经成为了 Python 的标准库。Unittest 类库中包含许多方法可供编程时调用,提高了开发测试脚本的效率和质量。

2 Web 软件项目案例

在某视频直播转码软件项目中,与 Web 页面相关的部分作为一个单独的软件模块开发。该模块作为项目的业务功能模块之一,支持从 Web 页面对视频文件进行直播转码。

2.1 软件开发技术

该软件模块采用 MVC 设计模式,使用当前最流行的 Struts2 + Spring3 + Hibernate 框架进行开发。前台采用 JSP 进行页面开发和用户界面管理,使用 JavaScript 结合开源的前端插件 jQuery EasyUI 进行前端页面开发。后台采用 Java 与 MySQL 数据库进行开发,采用当前主流的 Java 开源开发工具 Eclipse 和 Tomcat 服务器进行程序开发和发布。

2.2 jQuery EasyUI

jQuery 是一套简洁、灵活的 JavaScript 脚本库,可以帮助 Web 页面开发工程师简化 JavaScript 代码。jQuery 由于简便易用的特点,在 Web 前端开发中被广

泛使用。

EasyUI 作为一款开源前端插件集合,功能十分强大。应用这些插件可以极大地减少 Web 开发方面的工作量。EasyUI 与 jQuery 结合在一起应用,jQuery 负责把 EasyUI 的各种插件搬运到 Web 项目合适的地方,并且让它真正运转起来。

2.3 软件功能测试点

Web 软件功能测试点覆盖 Web 页面上各场景的正常与异常功能测试点。Web 页面的功能测试包括以下场景:Web 页面登录与注销;Web 页面登录密码修改;直播频道增、删、改、选中、多频道翻页;频道显示条数、频道各参数配置;直播频道启用和禁用;浏览器兼容性测试,等等。

3 自动化测试软件开发过程

自动化测试软件开发过程主要包括:仔细研读软件需求说明书、分析测试需求、提交测试策略、设计测试用例、搭建测试环境。手动执行测试、选择自动化测试用例、开展自动化测试工作。具体过程如图 1 所示。

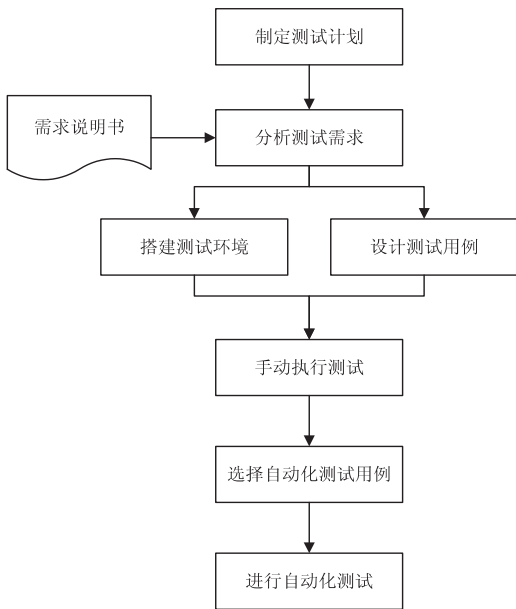


图 1 自动化测试软件开发过程

3.1 制定测试计划

明确自动化测试的对象、目的、内容、方法、进度要求,并确保测试所需的人力、软硬件设备、数据等资源准备充分。

3.2 分析测试需求

根据需求说明书,测试架构师(TSE)编写测试策略,设计测试功能点。Web 功能测试需要覆盖页面链接、控件、页面功能、数据处理、模块业务逻辑等多种功能的测试。

3.3 设计测试用例

TSE 针对正常和异常情况进行测试用例设计,测

试用例覆盖所有测试功能点,共涉及测试用例 288 个,并编写测试用例文档。

3.4 搭建测试环境

搭建测试所需的各种软硬件环境,如数据、网络、测试工具的安装和设置等。

3.5 手动测试

手动测试可以发现 Web 软件存在的各种问题,同时也可以检查测试用例设计中存在的问题,为开展自动化测试奠定坚实的基础。

3.6 选择自动化测试用例

选择可以自动化测试的测试用例,共选定测试用例 238 个,自动化率为 82.64%。通常兼容性测试用例不能实现自动化测试。

3.7 编写测试脚本开展自动化测试

采用 Python 与 Selenium 工具编写自动化测试脚本,脚本基于 Unittest 自动化测试框架。完成用例的测试脚本编写工作后,采取测试库框架优化测试用例脚本,从众多的脚本中提取出公共函数,供测试用例调用。通过脚本连跑实现对 Web 软件的自动化测试。

4 基于 Selenium+Python 的自动化测试软件开发

测试软件开发选择 Python 作为开发语言,使用 Selenium 提供的资源进行测试软件开发。工作内容包括安装各种测试工具、搭建测试环境;录制、编写测试脚本;基于测试库框架优化软件代码;自动化脚本连跑;测试结果分析。

4.1 安装测试工具

在执行测试任务的 PC 机(执行机)安装所需的各种测试工具。

4.1.1 Python 工具

安装包为 python-2.7.3.msi。安装完成后,在环境变量中增加 Python 工具的相关目录路径。Python 工具自带了一个轻量级的集成开发工具 IDLE, IDLE 可以进行测试脚本的编辑、调试和执行。

4.1.2 Selenium 工具

网上下载基于 Python 语言的软件包 selenium-2.50.1-py2-none-any.whl,执行 pip install selenium-2.50.1-py2-none-any.whl 进行软件包安装。安装完成后 Selenium 软件包安装在 Python 工具的 lib 目录下。

4.1.3 安装 Firefox 浏览器和 Selenium IDE 工具

首先安装 Firefox 浏览器,然后在该浏览器上安装 Selenium IDE 工具。Selenium IDE 是 Firefox 浏览器的插件,只能安装在 Firefox 浏览器上。

4.1.4 安装 MySQL 数据库的相关工具

在 Python 数据库中读写 MySQL 数据库,需要安装相

关的工具。

(1)安装 MySQL-python 工具,安装包为 MySQL-python-1.2.5.win32-py2.7.exe。

(2)安装 MySQL 连接器,安装包为 mysql-gui-tools-5.0-r17-win32.msi。

安装完成后,可以通过 SQL 语句对 MySQL 数据库表单进行操作。

4.1.5 pycharm 工具安装

pycharm 工具的安装包为 pycharm-community-5.0.3.exe。pycharm 工具是 Python 重量级 IDE,功能强大,自动检测语法,可以帮助开发者写出更规范的 Python 代码。

4.2 搭建测试环境

Java Web 软件^[17]基本组成如图 2 所示。搭建测试环境需要安装浏览器和 Web 应用软件。文中的 Web 软件选择 Chrome 或者 Firefox 浏览器作为客户端浏览器。

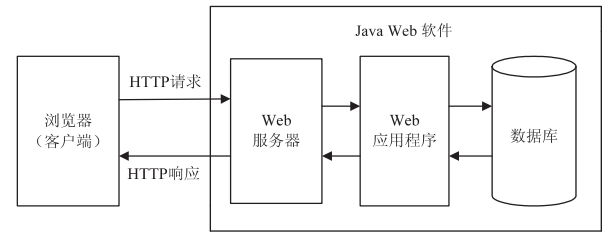


图 2 Java Web 软件基本组成

4.2.1 安装浏览器

在测试执行机上安装浏览器,目前安装的 Python 与 Selenium 工具要求谷歌浏览器的版本不能低于 43.0.2357.0。谷歌浏览器安装包为 48.0.2564.116_chrome_installer.exe。安装完谷歌浏览器后,还需要安装谷歌浏览器驱动程序 chromedriver.exe,将其安装在指定的目录下。

4.2.2 Web 软件测试环境部署

Web 软件测试环境部署在操作系统为 Windows 7 的 PC 机上。

(1)安装 JDK、Tomcat 与 MySQL。

JDK 安装包为 jdk-7u15-windows-is86.exe(1.7.0_15 版本)。Tomcat 安装包为 apache-tomcat-7.0.29-windows-x86.zip,直接解压、安装。MySQL 数据库安装包为 mysql-5.5.8-win32.msi。完成三个软件安装后,在环境变量中添加 JDK、Tomcat、MySQL 工具的相关目录路径。

(2)安装 Web 产品软件包。

Web 产品软件包为 *.war 格式的软件包,将软件包部署到 Tomcat 安装路径下的 webapps 文件夹中,部署完成之后,切换目录到 Tomcat 安装路径下的 bin 文件夹下,执行 startup.bat 文件启动 Tomcat 服务。网页

的 IP 地址为部署 Web 软件 PC 机的 IP 地址。

4.3 录制测试脚本

在 Firefox 浏览器上启动 Web 软件程序,找到菜单栏“工具”下的 Selenium IDE,单击后出现录制页面,点击红色录制按钮,开始录制。根据测试用例进行手动测试操作,测试操作完成后,按下红色按钮停止录制。从 Selenium IDE 工具导出录制的 Python 脚本,该脚本文件名与测试用例编号相对应。

目前 Web 页面的登录、注销以及修改密码等简单的场景采用 Selenium IDE 工具录制之后,可以导出正确可用的测试脚本。其他包含复杂测试场景的 Web 自动化测试用例录制后并不能生成真正可用的脚本。某些测试场景不支持通过 Selenium IDE 工具录制脚本。

脚本录制完成之后通常还需要修改、完善。录制的测试用例脚本拥有 Unittest 测试框架,为脚本的进一步编写工作提供了良好的基础。

4.4 编写自动化测试用例脚本

测试用例脚本采用 Python 进行程序设计。对于 Python 而言,代码缩进是一种语法,Python 没有像其他语言一样采用 {} 或者 begin...end 分隔代码块,而是采用代码缩进和冒号来区分代码之间的层次。缩进的空白数量是可变的,但是所有代码块语句必须包含相同的缩进空白数量,这个必须严格执行。Python 脚本中出现未对齐的行,该行下方就会出现红色曲线。可以使用空格键调整格式,直到红色曲线消失。

当脚本中一行的字符数过多时,会报 PEP 8 的告警。应该将长字符串分隔成多行显示。

4.4.1 自动化测试脚本框架

下面是一个简单测试用例脚本示例:

```
#- *- coding:utf-8 - *-
from selenium import webdriver
import unittest
.....
class BaiduTest(unittest.TestCase):
def setUp(self):
self.driver = webdriver.Chrome()
self.base_url = "http://www.baidu.com"
def test_baidu_search(self):
driver = self.driver
driver.get(self.base_url)
self.assertEqual(driver.title, u"百度一下,你就知道")
driver.find_element_by_id("kw").clear()
.....
def tearDown(self):
self.driver.quit()
if __name__ == "__main__":
unittest.main()
```

从脚本示例中可以看出,通常测试脚本由 5 部分组成:

(1) 导入需要调用的程序模块。

(2) 定义继承 unittest.TestCase 的测试类,从而可以使用 unittest.TestCase 类中的各种方法。

(3) 在初始化方法 setUp 中,初始化 webdriver,设置测试环境。该方法自动执行。

(4) 定义测试方法,首先定位 Web 元素,然后开展相关测试场景的操作。

(5) 在清理方法 tearDown 中,执行测试用例运行后的清理工作,如退出 webdriver、关闭驱动、恢复用例执行前的状态等。该方法自动执行。

4.4.2 Web 元素定位

Web 元素定位就是在 Web 页面上找到待测试的 Web 元素。手工测试时,测试人员容易在 Web 页面上找到 Web 元素开展测试工作。自动化测试通过测试程序在 Web 页面对应的 HTML 文件中定位 Web 元素,这是进行 Web 元素测试的第一步。Web 元素定位有多种方法,文中的软件项目测试过程中主要采用如下四种方法。

(1) 基于元素 ID 的定位。

该方法通过查找 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。

(2) 基于元素 XPATH 的定位方法。

元素 XPATH 的定位类似查找一个人的地址,通过页面 html 文件中的标签名的层级关系定位元素的绝对路径。这种定位元素方法在实际使用过程中,可以针对元素的 XPATH 进行修改,添加上、下层级的脚本标示,增加元素获取的准确性。

(3) 基于元素 CSS 的定位方法。

该定位方法查找 Web 元素的 CSS 属性来获取元素。CSS(cascading style sheets)层叠样式表是用来表现 HTML 或 XML 的标记语言。CSS 使用选择器来为页面元素绑定属性。元素 CSS 定位方法,相对前两种方法要复杂一些,元素的 CSS 属性选择比较灵活多样。

(4) 基于元素 LINK 的定位。

该定位方法仅适用于文本链接的场景,通过元素标签对应的文本信息来定位元素,相对其他的元素定位方法,适用的场景比较有限。

4.4.3 测试方法

完成 Web 元素定位后,需要对相关的场景进行各种操作,开展测试工作。文献[14]介绍了采用 Python 语言在各种常见场景中的操作方法。文献[15]采用 Java 语言介绍了同样的内容。网上也可以查到大量测

试场景操作方法。编写测试方法是脚本编写的核心内容。

下面介绍测试脚本编写过程中两种场景的处理方法:

(1) Web 页面中包含 iframe 框架。

Web 页面中包含 iframe 框架时,如果要提取 iframe 框架中的元素,需要在脚本中使用 swtich_to_iframe 实现框架切换。在该 iframe 框架上的相关操作完成后,需要从该 iframe 框架切换回来,在脚本中使用 swtich_to_default_content 来实现。

(2) 文本框或数字框输入非法值场景。

Web 页面上很多元素是使用 EasyUI 制作的,其中包含大量的文本框和数字框场景。由于填入文本或数字都有一定的取值范围限制,当文本框或数字框中填入非法值时,文本框或数字框的右侧通常会弹出 tooltip 类型的提示框,提示填入的内容非法。填入非法值之后,tooltip 提示框会在鼠标移动出对应文本框或数据框之后消失。测试脚本需要定位 tooltip 类型的提示框。

解决办法:当文本框与数字框中输入非法值时,可以通过基于元素 CSS 的方法来定位 tooltip 元素,检测 tooltip 中所提示的文本内容。具体方法如下:

```
Tip=driver.find_element_by_css_selector(“div[ class ‘ tooltip tooltip-right’ ]”).text
```

4.4.4 MySQL 数据库操作

测试脚本对 Web 软件进行频道添加、频道修改以及各类频道参数配置后,相关数据会写入 MySQL 数据库中对应的数据表中。在编写测试脚本时,需要通过 MySQL-python 工具来实现对 MySQL 数据库的操作,脚本实现时首先需要在脚本的最前面加上 import MySQLdb,然后在测试脚本的测试方法中添加对 MySQL 数据库的操作语句,实现对数据库的读写操作。代码如下所示:

```
conn = MySQLdb. connect ( host = ‘ 10. 185. 31. 52 ’ , port = ‘ 3306 ’ ,
user= ‘ root ’ ,password= ‘ huawei ’ ,db= ‘ hwwebom ’ ,charset= ‘ utf8 ’ )
cur=conn. cursor()
cur.execute(“select name from hwwebom. t_channel where name= ‘ %s ’ ” % channel_name)
for cha_name in cur.fetchall():
print cha_name
cur.close()
conn.commit()
conn.close()
```

4.4.5 执行用例脚本生成测试报告

测试用例脚本为 *.py 类型文件,在执行机上点

击该文件就可以自动运行。希望在测试用例脚本执行之后生成测试报告,需要使用 Unittest 的第三方测试报告模块 HTMLTestRunner(该模块不在标准库中,需要自行下载),加载该模块后可以生成 HTML 格式的测试报告。将 HTMLTestRunner.py 文件保存 Python 的 Lib 目录下,就可以在测试脚本中调用该模块,具体实现过程如下所示:

```
if __name__ == “__main__”:
now = time. strftime(“% Y-% m-% d-% H-% M-% S”)
filename = report_dir+ ‘ / ’ +now+ ‘ _Web_Login_result. html ’
fp = open( filename, ‘ wb ’ )
runner = HTMLTestRunner(
stream = fp,
title = unicode( ‘ 测试报告 ’ ),
description = Unicode( ‘ 测试执行情况: ’ )
)
runner. run( discover )
fp. close()
```

在生成测试报告文件时,将测试报告生成的时间作为测试报告文件名的一部分,并将测试报告文件生成的时间、测试用例运行结果写入测试报告。

4.5 基于测试库框架优化测试软件

测试库框架(test library framework)把应用程序分成过程和函数。该框架创建应用程序函数的库文件,库文件中的函数供应用程序调用。测试库架构框架的优点在于能够删除明显重复的部分,优化脚本代码。

独立编写的测试用例脚本,整体结构凌乱。需要将很多的公共场景与 Web 页面上的公共元素参数提取出来作为公共函数与共享参数;重新修改测试脚本,规范优化测试脚本;减少代码冗余,提高脚本质量;同时也方便业务扩展之后的脚本修改和维护。

4.5.1 测试软件目录结构

测试软件的目录结构如图 3 所示。测试脚本按功能分类存放在各自文件夹中。公共函数与公共配置文件存放在 Public 文件夹中。执行连跑任务的 Python

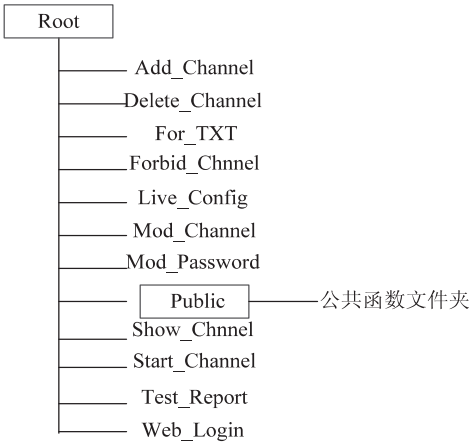


图 3 测试软件目录结构

脚本存放在根(Root)目录中。测试报告存放在 Test_report 目录中。

4.5.2 公共函数的功能

Public 文件夹中共包含了 5 个公共函数文件。其他文件夹中测试脚本可以调用 Public 文件中公共函数文件中的函数,在 Public 文件夹中需要添加__init__.py 文件。5 个公共函数文件实现的功能如表 1 所示。

表 1 公共函数的功能

公共文件的文件名	实现的功能
login.py	Web 页面登录与注销场景
password_change.py	Web 页面登录密码修改等场景
channel_operation.py	频道增、删、改、选中,多频道翻页,频道显示条数等场景
channel_param_config.py	频道各参数配置场景
mysql_operation.py	数据库查询与删除的场景

4.5.3 公共配置文件

Python 标准库的 ConfigParser^[18] 模块提供一套 API 来读取和操作配置文件。ConfigParser 是用来读取配置文件的包,可以把一些常用的参数放在配置文件中,供多个脚本共同使用。

公共配置文件中主要写入元素定位参数。在测试脚本编写中,多个测试脚本需要对同一 Web 元素定位,将该元素的定位参数值写入配置文件,方便编程时共同使用。

以场景登录(Login)时输入“用户名”和“密码”为例,叙述公共配置文件的使用方法。

(1) 在脚本开头添加:import ConfigParser。

(2) 打开公共配置文件。

```
all_dir=os.path.split(os.getcwd())
ini_dir=all_dir[0]
css_selector_filename = os.path.join(ini_dir, 'css_selector.ini')
```

(3) 获取配置文件中指定的配置信息。

```
[global]
Tip_username_Null=/html/body/div[2]/div[3]/form/div[1]/div/small[1]
Tip_Password_Null=/html/body/div[2]/div[3]/form/div[2]/div/small
```

(4) 将获取的元素定位信息赋值给元素定位函数的参数变量,进行定位场景操作。

4.6 测试脚本连跑

测试脚本连跑是指全部测试脚本在程序控制下依次自动运行。

4.6.1 测试脚本连跑实现方法

脚本连跑使用 discover 方法,具体实现方式如下所示:

```
report_dir = './Test_Report'
#discover 方法定义
discover = unittest.defaultTestLoader.discover(test_dir, pattern='*.py')
```

定义连跑路径变量和测试结果报告存放路径变量,然后调用 discover 方法执行连跑路径下所有的测试脚本,连跑完成之后生成每个测试脚本的测试报告。

4.6.2 设定定时脚本连跑任务

采用相关工具设定定时连跑任务。

(1) BAT 脚本启动连跑任务。

编写 BAT 脚本,调用根(Root)目录下的 Python 连跑脚本,启动连跑任务。脚本内容如下所示:

```
@echo off
set Script_path=D:\Python_27_3\Project_Live_New
set Report_path=D:\Python_27_3\Project_Live_New\Test_Report
taskkill /f /im Firefox.exe /t
cd /d %Script_path%
::调用 Python 脚本执行连跑
python Report_Test_Mod_Password.py -v
```

(2) Windows 的任务计划程序定时连跑。

若执行机的操作系统是 Windows,在操作系统的附件中启动任务计划程序,设置 Bat 脚本定时运行任务。

4.6.3 持续集成工具定时连跑

在执行机上安装持续集成工具 Jenkin 或者 ICP-CI,编写 Ant 脚本,创建持续集成工程,进行 Web 自动化脚本定时连跑。

5 测试结果分析

基于 Selenium+Python 的自动化测试软件开发完成后,使用效果良好。

(1) 提高测试效率。

手动执行 288 个 Web 测试用例耗时 7 人/天,且无法实现反复测试,整理测试报告也需要花费大量时间与人力。Web 自动化测试脚本实现了 82.64% 的测试场景自动化,238 个自动化测试用例连跑大约需要 3 个小时,同时可以实现测试报告自动输出,可以利用非工作时间反复执行测试任务。

(2) 预测试。

可用于软件集成测试、系统测试、验收测试的预测试。

6 结束语

结合 Web 软件项目详细叙述了基于 Selenium+Python 的自动化测试软件开发过程,介绍了开发过程中

- [8] 周世兵,徐振源,唐旭清. K-means 算法最佳聚类数确定方法[J]. 计算机应用,2010,30(8):1995-1998.
- [9] 袁方,周志勇,宋鑫. 初始聚类中心优化的 k-means 算法[J]. 计算机工程,2007,33(3):65-66.
- [10] 邓海,覃华,孙欣. 一种优化初始中心的 K-means 聚类算法[J]. 计算机技术与发展,2013,23(11):42-45.
- [11] 田诗宵,丁立新,郑金秋. 基于密度峰值优化的 K-means 文本聚类算法[J]. 计算机工程与设计,2017,38(4):1019-1023.
- [12] HIND J. Organizational patterns in discourse, syntax and semantics; discourse and syntax [M]. New York: Academic Press,1979.
- [13] 刘秋水. Web 信息抽取与网页摘要的研究与应用[D]. 大连:大连理工大学,2008.
- [14] 曾利沙. 主题与主题倾向关联下的概念语义生成机制——也谈语篇翻译意识与 TEM8 语段翻译教学[J]. 外语教学,2007,28(3):83-87.
- [15] 林鸿飞,战学刚,姚天顺. 文本层次分析与文本浏览[J]. 中文信息学报,1999,13(4):7-15.
- [16] 郭庆琳,李艳梅,唐琦. 基于 VSM 的文本相似度计算的研究[J]. 计算机应用研究,2008,25(11):3256-3258.
- [17] 黄磊,伍雁鹏,朱群峰. 关键词自动提取方法的研究与改进[J]. 计算机科学,2014,41(6):204-207.
- [18] 陈磊磊. 不同距离测度的 K-Means 文本聚类研究[J]. 软件,2015,36(1):56-61.
- [19] 沈斌. 基于分词的中文文本相似度计算研究[D]. 天津:天津财经大学,2006.
- [20] MACQUEEN J. Some methods for clustering and analysis of multivariate observations [C]//Proceedings of 5th Berkeley symposium on mathematical statistics and probability. Berkeley: University of California Press,1967:281-297.
- [21] 索红光,王玉伟. 一种用于文本聚类的改进 k-means 算法[J]. 山东大学学报:理学版,2008,43(1):60-64.
- [22] REZAEI M R, LELIEVELDT B P F, REIBER J H C. A new cluster validity index for the fuzzy c-means[J]. Pattern Recognition Letters,1998,19(3-4):237-246.
- [23] 姚清标. 基于向量空间模型的中文文本聚类方法的研究[D]. 上海:上海交通大学,2008.
- [24] HALKIDI M, BATISTAKIS Y, VAZIRGIANNIS M. On clustering validation techniques [J]. Journal of Information Systems,2001,17(2-3):107-145.
- [25] 常兴治. 基于全局评价的文本分割技术研究[D]. 沈阳:东北大学,2004.
- [26] 刘务华,罗铁坚,王文杰. 文本聚类算法的质量评价[J]. 中国科学院研究生院学报,2006,23(5):640-646.

(上接第 52 页)

遇到的许多技术细节。工作实践表明,该测试软件在使用过程中效果良好,可以提高软件开发与测试的效率,降低软件的开发成本。同时弥补了手工测试时重复劳动的缺点,减少了测试人员大量的重复测试验证工作;也有助于做好软件项目的管理工作。

参考文献:

- [1] 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.
- [2] 吴伶琳. 基于 Selenium 的软件自动化测试的研究与应用[J]. 计算机与现代化,2013(2):65-68.
- [3] 曹磊,董科军,袁博文. 一种基于 Selenium 的 Web 应用软件自动化测试平台设计与实现[J]. 科研信息化技术与应用,2014,5(6):44-52.
- [4] 李艳,任洪敏,刘芳. 基于 Selenium JS UI 的自动化测试框架设计与实现[J]. 微型机与应用,2017,36(17):24-26.
- [5] 卢晨. 基于 Selenium 进行 Web 应用测试研究[J]. 软件导刊,2015,14(1):154-155.
- [6] 赵金丹. 基于 selenium 的 web 自动化测试脚本设计研究[J]. 科技传播,2014(1):94.
- [7] 柏莹. 基于 .NET 平台下 Web 自动化测试的研究与设计[D]. 西安:西安电子科技大学,2013.
- [8] 李潇烨. 企业项目管理系统的 Web 自动化测试研究与实现[D]. 西安:西安电子科技大学,2015.
- [9] 刘军. 基于 Selenium 的网页自动化测试系统设计与实现[D]. 武汉:华中科技大学,2014.
- [10] BRUNS A, KOMSTADT A, WICHMANN D. Web application tests with selenium[J]. IEEE Software,2009,26(5):88-91.
- [11] LEOTTA M, CLERISSI D, RICCA F, et al. Repairing selenium test cases: an industrial case study about web page element localization [C]//IEEE sixth international conference on software testing, verification and validation. Luxembourg: IEEE,2013:487-488.
- [12] XU Dianxiang, XU Weifeng, BAVIKATI B K, et al. Mining executable specifications of web applications from selenium IDE tests [C]//IEEE sixth international conference on software security and reliability. Gaithersburg, MD, USA: IEEE, 2012:263-272.
- [13] GUNDECHA U. Selenium testing tools cookbook [M]. Birmingham, UK: Packt Publishing Ltd., 2012.
- [14] 虫师. Selenium 2 自动化测试项目实战: 基于 Python 语言 [M]. 北京: 电子工业出版社, 2016.
- [15] 吴晓华. Selenium WebDriver 实战宝典 [M]. 北京: 电子工业出版社, 2015.
- [16] 张秋杰. 基于 pyUnit 框架的企业级软件自动化测试技术的研究 [D]. 北京: 北京邮电大学, 2010.
- [17] 孙利. Java Web 案例教程 [M]. 北京: 电子工业出版社, 2015.
- [18] 齐伟. 跟老齐学 Python 入门到精通 [M]. 北京: 电子工业出版社, 2016.