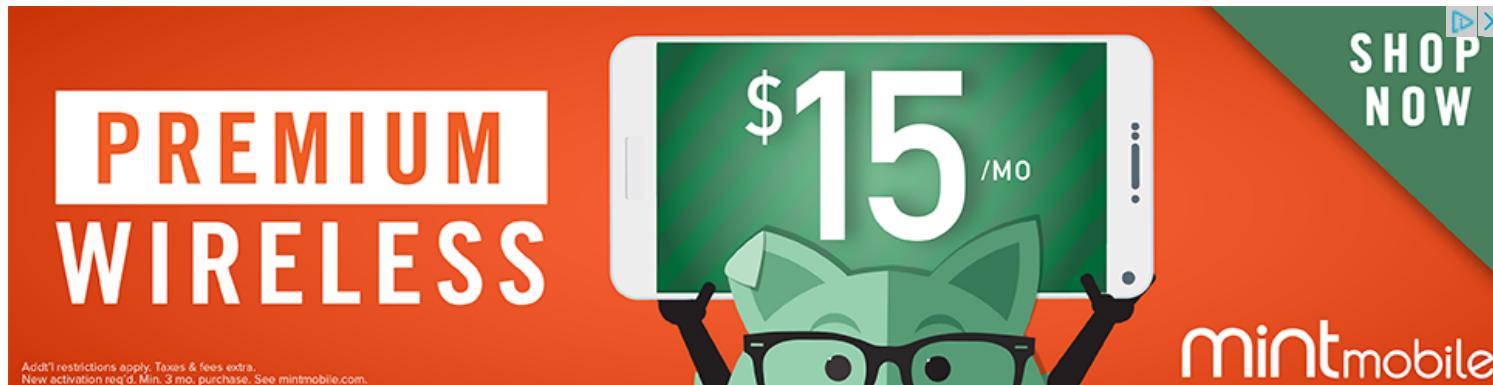


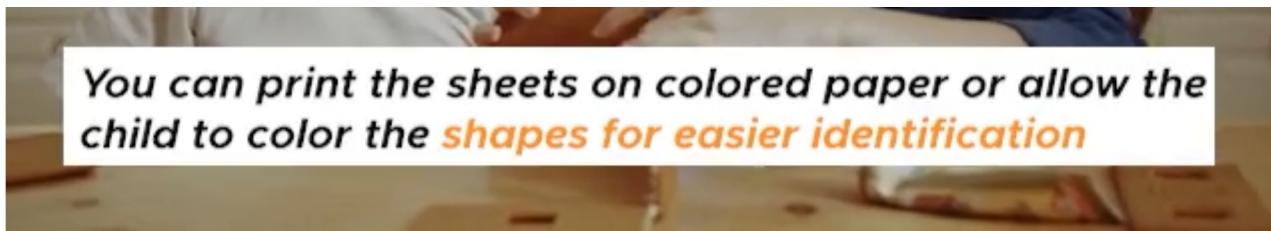
ProgrammerSought

 search

Selenium + web-based automated testing framework for Python

Add'l restrictions apply. Taxes & fees extra.
New activation req'd. Min. 3 mo. purchase. See mintmobile.com.

TOP



First, what is Selenium?

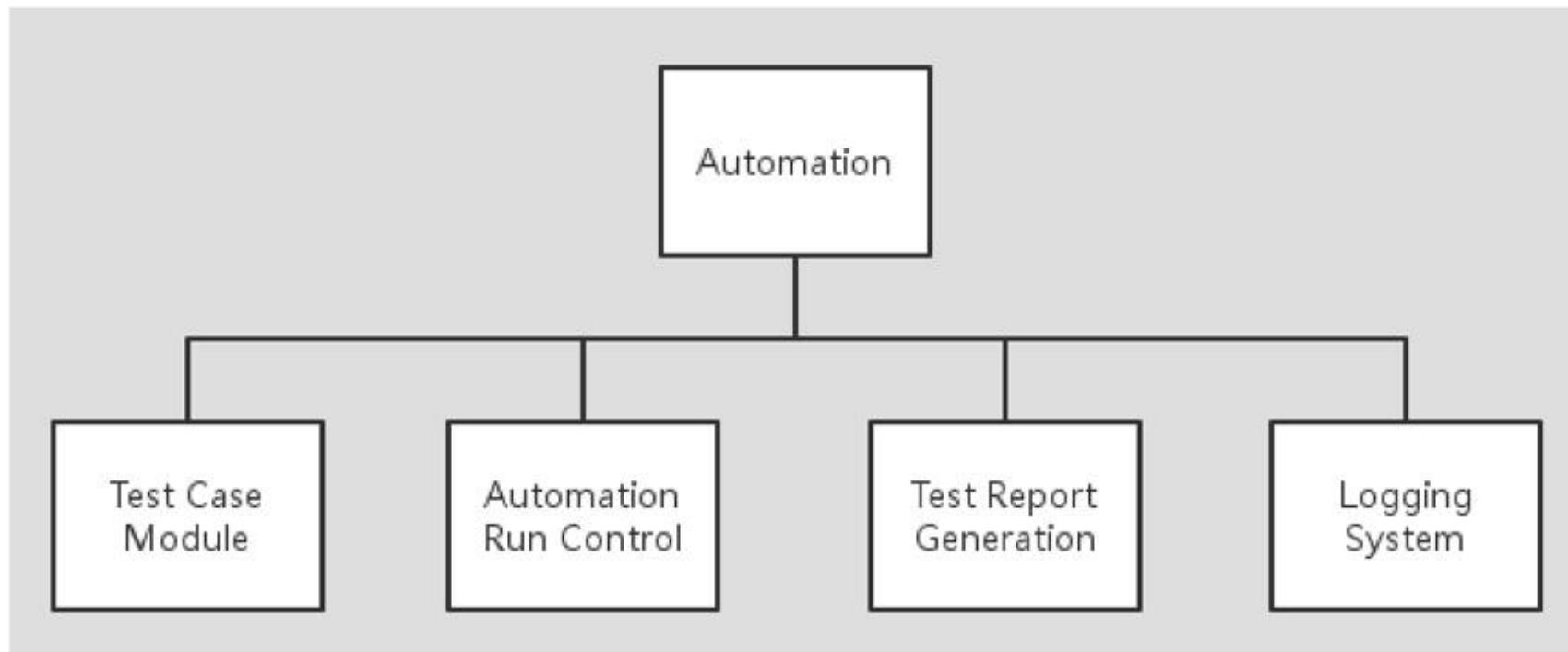
Selenium is a browser-based automated testing tool that provides a cross-platform, cross-browser web-to-end automation solutions. Selenium mainly includes three parts: Selenium IDE, Selenium WebDriver and Selenium Grid.

- Selenium IDE: a Firefox extension, it can play back the recording, and the recording operation (e.g., Java, Python, etc.) into the form of test cases derived languages.
- Selenium WebDriver: to provide the necessary automation Web API, mainly used browser control, select the page elements and debugging. Different browsers require different WebDriver.
- Selenium Grid: provides the ability to run selenium tests on different machines in different browsers.

This article details how to use Python to build a library in conjunction with Selenium WebDriver web automated testing framework.

Second, the automated testing framework

A typical automated testing framework embodiment generally comprises management module, automatic execution controller, and a report generation module log modules that complement each other.



Next comes the logic unit of each module:

1, with the management module cases

Use Case Management module includes add, modify, or delete operation units, which will involve the use case writing mode, the test database management, reusable libraries.

2, automation controller

The controller module is the automated organizational use cases executed, what method is mainly responsible for the implementation of our test cases.

3, report generation module

Mainly responsible for generating reports after execution of use cases, mostly general in HTML format, information is mainly the implementation of the use cases. It also can send e-mail configuration function.

4, log module

Primarily used for recording the implementation of the embodiment, in order to investigate the failure information for efficient tracking implementation examples and use cases.

Third, the design and automation framework

1, needs analysis

First, we test object is a web platform, based on this platform design framework to include test case management, test execution controller generation, test reports and test logs.

2, design and implementation

Page Management

Hypothesis testing web object to a typical single-page application, so we use page mode. Is the link between the page mode and the test pages, each page will be abstracted into a single class, to provide the operation and positioning of page elements to test.

As BaseClass parent driver member comprises only variable that identifies Selenium in the WebDriver, page elements to be positioned in a subclass. LoginClass and PageClass as a subclass, and operation method can provide positioning of page elements. For example, the login page.

登录博客园 - 代码改变世界

登录用户名(找回)

密码(重置)

下次自动登录

登 录

» 立即注册

» 反馈问题

From the page to see the elements need to operate are: login user name, password, and automatically log the next login button. Specific codes are as follows:

Page parent BaseClass.py

```
1  class BaseClass(object):
2      # 初始化
3      def __init__(self, driver):
4          self.driver = driver
5
6
```

LoginClass inherited from BaseClass, and element positioning login and operation implementation. And locating the code username password, and adds the user name and operator password.

```
1  from BaseClass import BaseClass
2      from selenium.webdriver.common.by import By
3      from selenium.webdriver.common.keys import Keys
4
5
6  class LoginClass(BaseClass):
7      # 定位元素
8      username = (By.ID, 'input1')
9      password = (By.ID, 'input2')
10     sign_in = (By.ID, "submit")
11
12     # 设置用户名
13     def set_username(self, username):
14         name = self.driver.find_element(*LoginClass.username)
15         name.send.keys(username)
16
17     # 设置密码
18     def set_password(self, password):
19         pwd = self.driver.find_element(*LoginClass.password)
20         pwd.send.keys(password + Keys.RETURN)
21
22     # 提交登录信息
23     def sign(self):
24         submit = self.driver.find_element(*LoginClass.sign_in)
25         submit.click()
26
```

Public library module

TOP

Public library modules to create test cases for services, including constants, public functions, logs, reports, and so on.

Common.py

```
1  from datetime import datetime
2
3
4  def base_url():
5      return 'https://passport.cnblogs.com/user/signin'
6
7
8  def current_time():
9      time = "%a %b %d %H:%M:%S %Y"
10     return datetime.now().strftime(time)
11
12
13 def time_diff(start, end):
14     time = "%a %b %d %H:%M:%S %Y"
15     return datetime.strptime(end, time) - datetime.strptime(start, time)
16
```

Test to identify test class information, and includes cases execution, an execution result information, including the following fields.

```
1 class TestCaseInfo(object):
2
3     def __init__(self, test_id="", name="", owner="", result="Failed", start="", end="", error_info ""):
4         self.id = test_id
5         self.name = name
6         self.owner = owner
7         self.result = result
8         self.start = start
9         self.end = end
10        self.info = error_info
```

Log mainly used to perform test recording step and the error information generation, we have different information of different levels of logging, such as Information, Warning, Critical and Debug. Because fewer log entries for each test case generation, so the test framework uses only the highest level of print logs that Debug level, which will also log all the other levels of information is also printed. Referenced Python standard library logging library In a particular implementation, in order to more easily control the log output.

Test Case warehouse

Use Case warehouse is mainly used to organize automated test cases. Each test case is abstracted into a separate class, and class inherit from unittest.TestCase. The Python unittest library provides a rich framework to support testing, including test cases setUp and tearDown methods, in the course of the use case implementations can be rewritten. Relying on page method and functions of public administration and public library modules page implementation, each test case script writing will be very clear and concise.

```
class TestLogin(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.base_url = C.base_url()
        self.testCaseInfo = TestCaseInfo(test_id=1, name='Test case name', owner='wu')

    def test_A(self):
        try:
            self.testCaseInfo.start = C.current_time()
            # 打开网页
            Logger("Open Base site" + self.base_url)
            self.driver.get(self.base_url)

            # 登录界面
            login_page = LoginClass(self.driver)

            # 输入用户名和密码
            Logger("Login web using username")
            login_page.set_username("username")
            login_page.set_password("password")

            time.sleep(2)

            login_page.sign()

        except Exception as err:
            self.testCaseInfo.error_info = str(err)
            Logger(("Got error: " + str(err)))
        finally:
            self.testCaseInfo.end = C.current_time()
            self.testCaseInfo.secondsDuration = C.time_diff(self.testCaseInfostarttime, self.testCaseInfoendtime)

    def tearDown(self):
        self.driver.close()
```

From this test case, we can see

- Setup defines some examples of work before the implementation of test cases
- tearDown for executing the test and do a clean-up work write log files

- Test procedures, test data and test check point very clear, easy to modify (such as user name and password)
- Just log level Debug, so just write the log using the same method Log

3. for example, execution module

Module is mainly used to control execution of a test case script batch execution, to form a test set.

REFERENCE EXAMPLE execution of the Python standard library subprocess nosetests shell commands to execute to perform a given test set cases. Test suite is a simple text file, use the process to achieve a .txt file testcases.txt.

```
1  Test_Login_pass.py
2  Test_Login_Fail.py
3  #Test_MainPage_CheckSecurityTableInfo.py
4  Test_MainPage_EditSecurityInfo.py
5
6
```

With the former cases there is no "#" mark test script will be executed, and the "#" mark will be ignored, so you can easily control the execution of the test set, of course, can create different files to perform different tests set.

Fourth, the need to improve the module

For existing testing framework to achieve, we have to meet the automation needs of web objects, but some can be improved to improve the place, such as:

- EXAMPLE portion may attempt to use the data driver.
- By encapsulating the secondary function of selenium, for more efficient targeting elements.
- Integration is not continuous.

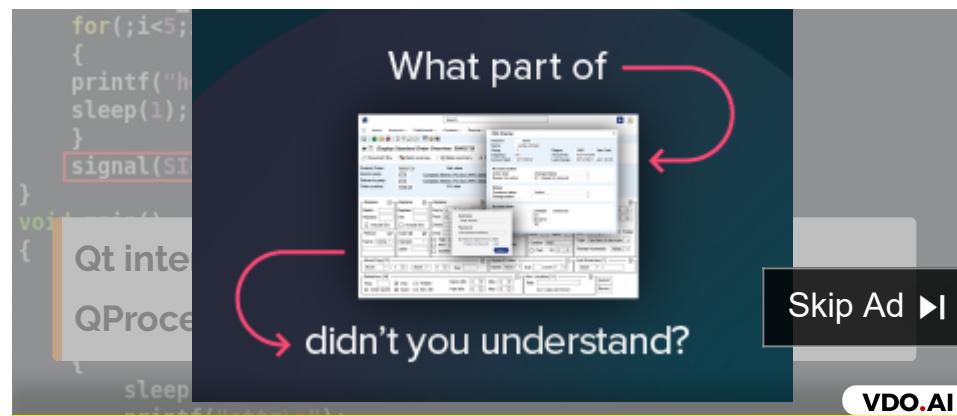
V. Summary

Help you design and implement future web automation framework web-based automated testing framework Selenium achieve not only lightweight but also flexible and can quickly develop automated test cases, binding framework design Benjian and some good practices, hope .

Author: Wu Guan Xiang

Source: CreditEase Institute of Technology





Intelligent Recommendation

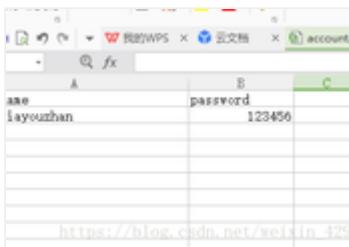
Python + selenium for automated testing - Web UI automated testing

1.1 Python's operating environment When I decided to learn a new language Python, Python3 has come out. Currently, Python2 and Python3 are maintained simultaneously. Python2 has a large number of class...



Web automated testing--python+selenium automated testing environment

Python+selenium automated test environment construction 1. Download and install python 3
2. Download and install Selenium 3 (1) Online installation method (recommended) (2) Offline installation 3. Dow...



Python+selenium automated testing framework (1)

Python+selenium automated testing framework (1) I wrote my blog for the first time. On the one hand, I wanted to improve my writing ability. On the other hand, I took time to solve the problem. I reco...

selenium + python unittest automated testing framework

1, first, the code library introduced unittest 2, inherited unittest code base test class TestCase subclass inherits all the properties and methods of the parent class, and may expand their property m...



Python + Selenium (1) - automated testing framework

1. What is the automated testing framework In short, automated testing framework by a few standards, protocols, rules of composition, provides an environment to run the script. Automated testing frame...

TOP

ANALYZING/PRESENTING DATA/INFORMATION

Fundamental design strategies for displaying information.
Taught entirely by Edward Tufte and includes all 5 books.
Generous Faculty, Student, Postdoc discount.

WWW.TUFTECOM 800 822-2454
"The daVinci of data" NY Times "The Galileo of graphics" Bloomberg

More Recommendation

```

3 Dec 27 14:00:00
ad at 18:18 ...
www(desktop): user "C:\Users\Lee\PyCharm 2018.2\helpers\pycharm
tests with arguments python -m unittest test_E2E_PyTh
passed!
finished!
31.44ms.

bed with exit code 0

```

Selenium+python+unittest automated testing framework (1)

background Recently, I used my spare time to learn some basic knowledge of selenium, but in the actual work project, it has not yet been applied. I still think about it and practice more. Everyone can...

Python pytest selenium automated testing framework

TOP

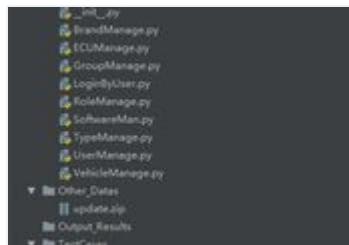


Python pytest selenium automated testing framework The company has always had this need for automated testing. In the early stage, it used c++ or python to write some automated scripts. There has been...



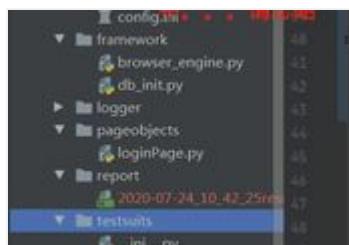
Implement a Python+Selenium automated testing framework

Personal blog navigation page (clickRightlinkTo open the personal blog): Daniel takes you to the technology stack First of all you have to know what is Selenium? Selenium is a browser-based autom...



Python+selenium automated testing framework design

Project Introduction The menu list is displayed as follows: There are multiple sub-menus after each parent menu is expanded. The functions in the sub-menus are similar, including new, edit, delete, se...



python+selenium+mysql+HTMLTestRunner automated testing framework

I wrote a simple, but basic function of automated testing framework, to share with you First select the tool. After analysis, the company's project manpower and time do not allow us to carry out...

TOP



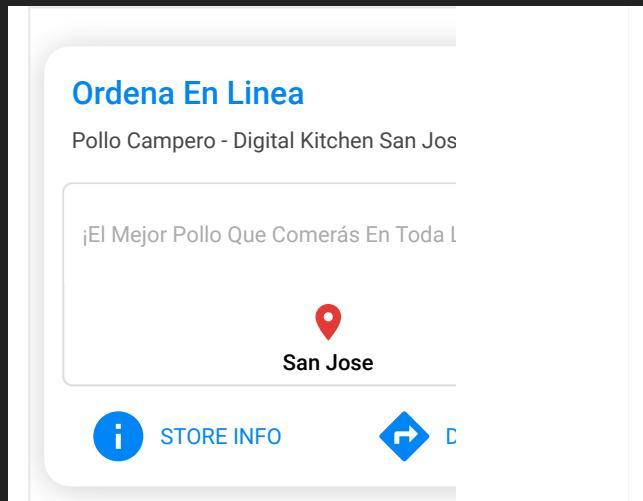
A job advertisement for Hexaware Technologies Ltd. It features a blue and orange design with the text "Contribute to develop the code for tomorrow" and "Become a JAVA Full Stack Developer at Hexaware". A hand holding a smartphone displays the Java logo. The Hexaware logo is in the bottom left. A white overlay box contains the text "Prepare to develop the code for your success". At the bottom, it says "Hexaware Technologies Ltd" and "Open >".

Related Posts

- Python3 Selenium automated web testing: Python unit testing framework unittest
- Based on the automated testing framework Selenium
- A web-based automated testing framework selenium webdriver of Patatium\WebUi
- Selenium Web automated testing-a PO design pattern based on the unittest framework
- python+selenium automated testing framework
- python + selenium web automated testing
- Web-based automated testing based on Selenium
- Practice of Selenium Web Automated Testing Framework
- Pyse automated testing framework based on selenium

TOP

- Analysis of automated testing framework based on Selenium and TestNG



Popular Posts

- Test Questions Basic Exercises Hexadecimal to Decimal
- Tools (to be completed)
- Enter two monotonically incremented linked lists, output two linked lists after two linked lists, of course, we need to synthesize the lathecules to meet the monotonicity.

TOP

- EFI secure boot
- leetcode first brush _Unique Binary Search Trees II
- [Ceph] Design and use of Bufferlist in Ceph
- After the DAO layer is encapsulated, you can realize the corresponding JavaBean object in accordance with a certain property.
- vue: Menu shrink function
- DesignPattern Series __03 Dependency Inversion Principle
- AcWing 1082 Number Game





Recommended Posts

- Android content observer's usage notes
- Use of Linux-rpm package and build third-party software warehouse
- Simple neural network processes
- SpringBoot2.0 integrates redis
- Stack
- How does the secret whale "liquidator" that helps DeFi run smoothly?
- Php tutorial DOM parsing xml document instance
- C# implements udp broadcast to collect all device information of LAN
- Swagger automatically generates interface documentation
- Python operation and maintenance tool introduced 1-fabric



Benefits of ballroom dancing

West Valley Dance Company
San Jose

Related Tags

software test

automated test

unittest

python

Design Patterns

java

selenium

vue

Python automated testing

Test development

TOP

Advertising Sponsor Link

- I'm Feeling Lucky
- Hard work, too tired, take a break
- Contact us to advertise

Copyright 2018-2021 - All Rights Reserved - www.programmersought.com