Web Automation with Selenium WebDriver

Andrew Malone
Harvard abcd-www
November 13, 2013
SELENIUM
Selenium

Selenium is a suite of tools to automate web browsers across many platforms.
Confusion!

Selenium
Selenium IDE
Selenium RC
Selenium 1

Selenium 2
Selenium Webdriver
Selenium Server
Selenium Grid
Selenium Webdriver makes direct calls to the browser using each browser’s native support for automation.
Current browser support

Google Chrome 12+
Internet Explorer 6, 7, 8, 9
Firefox 3+
Opera 11.5+
HtmlUnit 2.9
Android – 2.3+
iOS 3.2+
Safari 5.1+
<table>
<thead>
<tr>
<th>Language bindings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
</tr>
<tr>
<td>C#</td>
</tr>
<tr>
<td>Ruby</td>
</tr>
<tr>
<td>Python</td>
</tr>
<tr>
<td>Javascript (Node)</td>
</tr>
<tr>
<td>Perl</td>
</tr>
<tr>
<td>PHP</td>
</tr>
<tr>
<td>Haskell</td>
</tr>
<tr>
<td>Objective-C</td>
</tr>
</tbody>
</table>

Examples in this presentation will be in Python
GETTING UP AND RUNNING
Assumptions

Python is installed – version 2.6, 2.7, 3.2 or 3.3
Selenium Webdriver bindings are installed
(available as a Python package)
# import webdriver
from selenium import webdriver

# start a browser session
browser = webdriver.Chrome()

# Load a web page
browser.get("http://w3.abcd.harvard.edu")
LOCATING ELEMENTS
Elements

Interactions are with specific elements on a page. An element is an individual DOM node.
<table>
<thead>
<tr>
<th>Id</th>
<th>Partial link text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Class name</td>
</tr>
<tr>
<td>Tag name</td>
<td>Css selector</td>
</tr>
<tr>
<td>Link text</td>
<td>Xpath</td>
</tr>
</tbody>
</table>
# Webdriver location strategies

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td><code>browser.find_element_by_id(&quot;textInput&quot;)</code></td>
</tr>
<tr>
<td>name</td>
<td><code>browser.find_element_by_name(&quot;input Name&quot;)</code></td>
</tr>
<tr>
<td>tag name</td>
<td><code>browser.find_element_by_tag_name(&quot;input&quot;)</code></td>
</tr>
<tr>
<td>class name</td>
<td><code>browser.find_element_by_class_name(&quot;input&quot;)</code></td>
</tr>
<tr>
<td>css</td>
<td><code>browser.find_element_by_css_selector(&quot;input.input&quot;)</code></td>
</tr>
<tr>
<td>xpath</td>
<td><code>browser.find_element_by_xpath(&quot;/input[@name=textInput]&quot;)</code></td>
</tr>
<tr>
<td>link text</td>
<td><code>browser.find_element_by_link_text(&quot;Web page&quot;)</code></td>
</tr>
<tr>
<td>partial link text</td>
<td><code>browser.find_element_by_partial_link_text(&quot;page&quot;)</code></td>
</tr>
</tbody>
</table>
Singular and Plural

Singular

```python
element = browser.find_element_by_css_selector("a.link")
element.click()
```

Finds the first element that matches the selector
Returns the element
Throws a Webdriver exception if no match is found
Singular and Plural

Plural

```python
elements = browser.find_elements_by_css_selector("a.link")
elements[0].click()
```

Finds all elements that match the selector
Returns a list of elements
Returns an empty list if no match is found
Alternate locator syntax

These two statements are equivalent

```python
cheese = browser.find_element_by_name("cheese")
```

```python
from selenium.webdriver.common.by import By
cheese = browser.find_element(By.NAME, "cheese")
```
Choosing a location strategy

Use id or name whenever possible

CSS attribute selectors and pseudoclasses can be extremely powerful

Use xpath as a last resort
CSS attribute selectors

Attribute contains:

```html
a[href*="registerForm"]
```

Attribute ends with:

```html
a[href?="pdf"]
```
CSS pseudoclasses

```html
<table id="datatable">
  <tr>
    <th>1</th>
    <td>Num 1</td>
    <td>Num 2</td>
  </tr>
  <tr>
    <th>2</th>
    <td>Num 3</td>
    <td>Num 4</td>
  </tr>
</table>

#datatable tr:nth-child(2) td:nth-of-type(1)
```
XPATH alternatives

```xml
<table id="datatable">
  <tr>
    <th>1</th>
    <td>Num 1</td>
    <td>Num 2</td>
  </tr>
  <tr>
    <th>2</th>
    <td>Num 3</td>
    <td>Num 4</td>
  </tr>
</table>
```

// table[@id='datatable']/tr[2]/td[1]
XPATH alternatives

```xml
<table id="datatable">
  <tr>
    <th>1</th>
    <td>Num 1</td>
    <td>Num 2</td>
  </tr>
  <tr>
    <th>2</th>
    <td>Num 3</td>
    <td>Num 4</td>
  </tr>
</table>

//th[text()='2']/following-sibling::td[1]
```
BUILDING A SIMPLE SCRIPT
Interaction methods

# click the element
```python
element.click()
```

# type into the element
```python
element.send_keys("some text")
```

# is the element visible to a user
```python
element.is_displayed()
```

# get an attribute of the element
```python
element.get_attribute("name")
```

# get the text of the element
```python
element.text
```
```python
from selenium import webdriver

browser = webdriver.Chrome()
browser.get("https://www.directory.harvard.edu/")
browser.find_element_by_name("lastName").send_keys("Malone")
browser.find_element_by_id("search").click()
browser.find_element_by_link_text("Andrew P. Malone").click()
xpath = "//font[text()='Appointment Title:']/../following-sibling::td"
title = browser.find_element_by_xpath(xpath).text

if title == "Senior Systems Operations Analyst":
    print "Title is correct"
else:
    print "Title is not correct"
```
Search the Public Harvard Directory

To look up Harvard students, faculty, and staff members, fill in one or more of the fields and click Search.

Last Name:
First Name:
Email:
Phone:

Search  Reset

Search the Internal Directory
Authenticate and view additional information.

Privacy
The compilation or redistribution of information from Harvard University directories is forbidden.
Privacy Statement

Help
Need Help with your Search?
Log in to mobile site

© Copyright 2013 President and Fellows of Harvard College
Search the Internal Directory
Authenticate and view additional information.
browser.find_element_by_name("lastName").send_keys("Malone")
browser.find_element_by_id("search").click()
Online Phonebook Search Results
The following person(s) matched your search criteria.

Name
- Andrew P. Malone

Department/School
- Harvard Medical School
- Harvard Business School
- Office of Vice President for Finance
- Faculty of Arts and Sciences

browser.find_element_by_link_text("Andrew P. Malone").click()
Listing Detail

Name: Andrew P. Malone
Appointment Title: Senior Systems Operations Analyst
Department: CADM/VPP/FSS Resrch Admin Syst
Office Phone: 617-824-5915

xpath = "//font[text()='Appointment Title:']/../following-sibling::td"
title = browser.find_element_by_xpath(xpath).text
from selenium import webdriver

browser = webdriver.Chrome()
browser.get("https://www.directory.harvard.edu/")
browser.find_element_by_name("lastName").send_keys("Malone")
browser.find_element_by_id("search").click()
browser.find_element_by_link_text("Andrew P. Malone").click()
xpath = "//font[text()='Appointment Title:']/../following-sibling::td"
title = browser.find_element_by_xpath(xpath).text

if title == "Senior Systems Operations Analyst":
    print "Title is correct"
else:
    print "Title is not correct"
Tips

Learn and use the web inspector and browser console

\$("css selector")

\$x("xpath selector")
PAGE OBJECTS
Page object pattern

Keep all the webdriver interactions in one place
Adds a layer of abstraction between webdriver implementation and website/application functionality
Search the Public Harvard Directory

To look up Harvard students, faculty, and staff members, fill in one or more of the fields and click Search.

Last Name: 
First Name: 
Email: 
Phone: 

Search Reset

Search the Internal Directory

Authenticate and view additional information.

Tips
Search other directory and networking sites:
Harvard Medical School White Pages,
Department Listings.

Search Help
Need help with your search?

Privacy
The collection or redistribution of information from
Harvard University directories is forbidden.
Privacy Statement

Help
Need Help with your Search?
Log in to mobile site

© Copyright 2013 President and Fellows of Harvard
College
class Search_Page(Page):
    locators = {
        "last_name": "name=lastName",
        "search": "id=search"
    }

    def go(self):
        self.browser.get("https://www.directory.harvard.edu/"
        return self

    def set_last_name(self, last_name):
        self.find(self.locators["last_name"]).send_keys(last_name)

    def search(self):
        self.find(self.locators["search"]).click()
        return Results_List(self.browser)
class Page(object):
    def __init__(self, browser):
        self.browser = browser

    def find(self, locator):
        # convenience method for writing simpler locators
        from selenium.webdriver.common.by import By
        locator_map = {
            "css": By.CSS_SELECTOR,
            "id": By.ID,
            "name": By.NAME,
            "link": By.LINK_TEXT,
            "xpath": By.XPATH
        }
        locator_type = locator[:locator.find("=")]
        locator_value = locator[locator.find("=") + 1:]
        return self.browser.find_element(locator_map[locator_type], locator_value)
class Search_Page(Page):
    locators = {
        "last_name": "name=lastName",
        "search": "id=search"
    }

def go(self):
    self.browser.get("https://www.directory.harvard.edu/")
    return self

def set_last_name(self, last_name):
    self.find(self.locators["last_name"]).send_keys(last_name)

def search(self):
    self.find(self.locators["search"]).click()
    return Results_List(self.browser)
Online Phonebook Search Results

The following person(s) matched your search criteria.

<table>
<thead>
<tr>
<th>Name</th>
<th>Department/School</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam Malone</td>
<td>Harvard Medical School</td>
</tr>
<tr>
<td>Al-Ling Jamila Malone</td>
<td>Harvard Business School</td>
</tr>
<tr>
<td>Andrew P. Malone</td>
<td>Office of Vice President for Finance</td>
</tr>
<tr>
<td>Brad D. Malone</td>
<td>Faculty of Arts and Sciences</td>
</tr>
<tr>
<td>Christopher Maurice Malone</td>
<td>Grad School of Business Admin</td>
</tr>
<tr>
<td>Clare Felicity Malone</td>
<td>Grad School of Arts and Sci</td>
</tr>
<tr>
<td>Daniel James Malone</td>
<td>Graduate School of Design</td>
</tr>
<tr>
<td>David Malone</td>
<td>Campus Services</td>
</tr>
<tr>
<td>Johanna Catherine Malone</td>
<td>Harvard Medical School</td>
</tr>
<tr>
<td>Patricia Ann Malone</td>
<td>Faculty of Arts and Sciences</td>
</tr>
<tr>
<td>Roberta Malone</td>
<td>Harvard University Division of Continuing Education</td>
</tr>
<tr>
<td>Ryan Malone</td>
<td>DCE - Harvard Extension School</td>
</tr>
<tr>
<td>Sean Ryne Malone</td>
<td>Harvard College</td>
</tr>
</tbody>
</table>

Search Again
class Results_List(Page):
    locators = {
        "name": "link=REPLACE"
    }

    def click_name(self, name):
        self.find(self.locators["name"]
            .replace("REPLACE", name)).click()

    return Results_Page(self.browser)
Listing Detail

Name: Andrew P. Malone
Appointment Title: Senior Systems Operations Analyst
Department: CADM^VPF^FSS Rsrch Admin Syst
Office Phone: +1 617 384 5915
8 Story St. 209
Office Fax: +1 617 384 8324
University Mailing Address:
Harvard Univ
Suite 209
8 Story St
Cambridge MA 02138
Email: andrew_malone@harvard.edu
Unit: Office of Vice President for Finance
class Results_Page(Page):
    locators = {
        "title": "xpath=//font[text() = 'Appointment Title']/../following-sibling::td"
    }

def title(self):
    return self.find(self.locators["title"]).text
from selenium import webdriver
browser = webdriver.Chrome()
search_page = Search_Page(browser).go()
search_page.set_last_name("Malone")
results_list = search_page.search()
results_page = results_list.click_name("Andrew P. Malone")
title = results_page.title()
Why use page objects?

When underlying source of pages change, updates only need to be made in one place.

Allows script writers to focus on tests, not on details of webdriver interactions.
WHAT CAN YOU USE THIS FOR?
Streamline testing efficiency

Data staging
Creating specific application state for testing
Other uses

- Automated test execution
- Data driven test
- Cross browser testing
- Running Selenium on a server
- Integration with a build process
RISKS
“Tools can extend or enhance or enable fabulous testing. But by accelerating some action, tools can enable us to do bad testing faster than ever...”

http://www.developsense.com/blog/2013/02/manual-and-automated-testing/
Automation is software development

Does your team have the right skills to create maintainable, extendable, reliable code?
Scope creep

Aiming for 100% test coverage is not the right goal. Focus instead on reducing risk or increasing efficiency.
Don’t be like this...
Be like this!
THANK YOU!

Andrew Malone
Financial Systems Solutions (FSS)
andrew_malone@harvard.edu
External resources

Selenium project homepage
http://www.seleniumhq.org/

Python webdriver bindings

GMAS automation project
https://git.huit.harvard.edu/gmas_webdriver