

TÀI LIỆU HƯỚNG DẪN SELENIUM

Chương 1. Giới thiệu

Đây là công cụ tốt nhất cho kiểm tra web-application. Kiểm tra tự động hóa có nghĩa là sử dụng một công cụ để chạy thử nghiệm lặp lại đối với các ứng dụng nhằm mục tiêu bất cứ khi nào cần thiết.

Chương 2. Introducing Selenium

2.1. Tự động hóa hay không tự động hóa?

Selenium không phải luôn gặp thuận lợi khi để tự động hóa các trường hợp kiểm thử. Có khi kiểm thử bằng tay lại thích hợp hơn. Ví dụ như giao diện của người dùng trình ứng dụng sẽ thay đổi trong tương lai gần, như vậy thì bất kỳ tự động hóa nào cũng được viết lại. Đôi khi đơn giản là không đủ thời gian để viết tự động hóa. Đối với kỳ hạn ngắn, kiểm tra bằng tay có thể có hiệu quả hơn. Nếu một ứng dụng có một thời hạn rất chặt chẽ, hiện tại không có tự động kiểm tra có sẵn, và nó bắt buộc là việc thử nghiệm được thực hiện trong khoảng thời gian, kiểm tra bằng tay là giải pháp tốt nhất.

Tuy nhiên, tự động hóa có lợi thế cụ thể để nâng cao hiệu quả lâu dài của một nhóm phần mềm kiểm tra quy trình. Kiểm tra tự động hỗ trợ:

- Thường xuyên kiểm tra hồi quy
- Phản hồi nhanh chóng để phát triển trong quá trình phát triển
- Hầu như không giới hạn sự thực hiện lặp đi lặp lại trường hợp thử nghiệm
- Tùy chỉnh báo cáo của các sai sót ứng dụng
- Hỗ trợ cho Agile và phương pháp phát triển eXtreme
- Tài liệu có tính kỷ luật của các trường hợp thử nghiệm
- Tìm các khuyết tật bị mất bằng cách kiểm tra bằng tay

2.2. Test Automation for Web Applications

2.3. Introducing Selenium

Selenium là một tập hợp mạnh mẽ của các công cụ hỗ trợ phát triển nhanh chóng của các thử nghiệm tự động hóa cho các ứng dụng dựa trên web. Selenium cung cấp một tập phong phú của các thử nghiệm chức năng đặc biệt hướng đến các nhu cầu của các thử nghiệm của một ứng dụng web. Các hoạt động này là rất linh hoạt, cho phép nhiều tùy chọn cho vị trí các thành phần UI và so sánh kết quả thử nghiệm dự kiến sẽ chống lại hành vi ứng dụng thực tế

2.4. Selenium Components

Selenium gồm có ba công cụ chính. Mỗi có một vai trò cụ thể trong việc hỗ trợ sự phát triển của tự động hóa kiểm tra ứng dụng web

2.4.1. Selenium-IDE

Selen-IDE là môi trường phát triển tích hợp cho việc xây dựng trường hợp thử nghiệm Selenium. Nó hoạt động như một trình duyệt Firefox add-on và cung cấp một giao diện dễ sử dụng để phát triển và chạy trường hợp kiểm thử cá nhân, bộ kiểm tra toàn bộ. Selenium-IDE có một tính năng ghi lại, sẽ giữ tài khoản của người sử dụng khi chúng được thực hiện và lưu trữ chúng như là một kịch bản tái sử dụng để phát sử dụng. Nó cũng có một menu ngữ cảnh (nhấn chuột phải) tích hợp với trình duyệt Firefox, cho phép người dùng chọn từ một danh sách xác nhận và xác minh cho các vị trí đã chọn. Selenium-IDE cũng cung cấp chỉnh sửa đầy đủ các trường hợp thử nghiệm cho chính xác hơn và kiểm soát.

Mặc dù Selen-IDE chỉ là một Firefox add-on, các kiểm thử tạo ra trong nó cũng có thể được chạy cho các trình duyệt khác bằng cách sử dụng Selenium-RC và chỉ định tên của bộ ứng dụng thử nghiệm trên dòng lệnh.

2.4.2. Selenium-RC (Remote Control)

Selen-RC cho phép các nhà phát triển tự động hóa kiểm tra sử dụng một ngôn ngữ lập trình cho tính linh hoạt tối đa và mở rộng trong việc phát triển logic thử nghiệm. Ví dụ, nếu trình ứng dụng trả về một tập kết quả ảo việc kiểm tra, và nếu chương trình thử nghiệm tự động cần chạy thử nghiệm trên mỗi phần tử trong tập hợp kết quả, hỗ trợ lặp đi lặp lại các ngôn ngữ lập trình có thể được sử dụng để chuyển đổi thông qua việc tập hợp kết quả, kêu gọi Selenium lệnh chạy thử nghiệm trên mỗi mục.

Selen-RC cung cấp một API (Application Programming Interface) và thư viện cho mỗi ngôn ngữ được hỗ trợ: HTML, Java, C #, Perl, PHP, Python, và Ruby. Khả năng sử dụng Selen-RC với một ngôn ngữ lập trình bậc cao để phát triển các trường hợp thử nghiệm cũng cho phép thử nghiệm tự động được tích hợp với một dự án xây dựng môi trường tự động.

Chương 3. SELENIUM BASICS

Selenium Commands có 3 “flavors” : Actions, Accessors and Assertions.

- Actions là các lệnh thao tác thông thường của ứng dụng như “Click this link” và “select that option” . nếu có 1 Action thất bại hoặc có lỗi, việc thực hiện các thực thi hiện tại dừng lại.
Nhiều Action có thể được gọi với các hậu tố “AndWait” e.g như là “ClickAndWait” nghĩa là hành động này sẽ làm cho browser gọi server và Selenium đợi để tải 1 trang mới.
- Accessors kiểm tra trạng thái của ứng dụng và lưu trữ các kết quả trong các biến e.g “storeTitle”. Chúng được gọi để tự động tạo ra Assertions.

- Assertions như là Accessors, nhưng cũng kiểm tra trạng thái của ứng dụng có phù hợp với những gì được mong đợi không. Ex: như “đảm bảo các tiêu đề trang là X” và “xác nhận checkbox này được kiểm tra”.
Tất cả Selenium Assertions có thể dùng trong 3 hình thức: “assert”, “verify”, “waitFor”. Ex: “assertText”, “verifyText” và “waitForText”. Khi 1 “assert” thất bại, test được hủy bỏ. Khi “verify” thất bại, thì test sẽ tiếp tục thực thi, login thất bại. Nó cho phép “assert” đảm bảo ứng dụng trên trang là chính xác,

“waitFor” lệnh chờ đợi đối với một số điều kiện để trở thành hiện thực (mà có thể hữu ích khi test các ứng dụng Ajax). Sẽ thành công ngay lập tức nếu điều kiện này luôn luôn đúng. Tuy nhiên, sẽ thất bại và ngăn chặn sự kiểm tra nếu điều kiện không đúng trong các thiết lập thời gian hiện tại (xem các hành động setTimeout dưới đây).

3.1. Script Syntax

Lệnh Selenium rất đơn giản, chúng bao gồm các lệnh và 2 tham số. Ex:

verifyText	//div//a[2]	Login
------------	-------------	-------

Các tham số không phải luôn luôn cần thiết, nó phụ thuộc vào lệnh. Trong một số trường hợp yêu cầu cả hai, một số chỉ yêu cầu tham số , và một số khác không yêu cầu cả cả hai

goBackAndWait verifyTextPresent type type	id = phone id = address1	Welcome To My Home Page (555) 666-7066 \${myVariableAddress}
--	-----------------------------	--

Các lệnh tham chiếu mô tả các yêu cầu tham số cho mỗi lệnh.

Thông số khác nhau, tuy nhiên chúng thường:

- một *location* để xác định một yếu tố giao diện người dùng trong trang.
- một *text pattern* thăm tra hoặc khẳng định trang nội dung dự kiến
- một *text pattern* hoặc một biến selenium để nhập văn bản vào một trường đầu vào hoặc để chọn một tùy chọn từ một danh sách tùy chọn.

Selenium Accessors

Chương 4. SELENIUM-IDE

4.1. Introduction

Các Selen-IDE (Môi trường phát triển tích hợp) là công cụ bạn sử dụng để phát triển các trường hợp kiểm thử Selenium của bạn. Nó dễ sử dụng cho Firefox plug-in và nói chung là cách hiệu quả nhất để phát triển các trường hợp kiểm thử. Nó cũng chứa

một menu ngữ cảnh cho phép bạn lựa chọn đầu tiên một yếu tố giao diện người dùng từ các trình duyệt hiện đang hiển thị trang và sau đó chọn từ một danh sách các lệnh Selenium với các thông số được xác định trước theo ngữ cảnh của phần tử giao diện người dùng lựa chọn. Điều này không chỉ tiết kiệm thời gian, mà còn là cách tuyệt vời để học cú pháp lệnh Selenium.

4.2. Installing the IDE

Dùng với firefox, trước tiên download IDE từ trang [downloads page](#)

4.3. Log/Reference/UI-Element/Rollup Pane

4.3.1. Log

Khi bạn chạy trường hợp thử nghiệm của bạn, thông báo lỗi và thông báo thông tin cho thấy sự tiến bộ được hiển thị trong cửa sổ này sẽ tự động, ngay cả khi bạn không đầu tiên chọn tab Log. Các thông điệp này thường có ích để gỡ lỗi trường hợp thử nghiệm. Chú ý nút Clear để xóa Đăng nhập. Ngoài ra thông báo vào nút Info là một lựa chọn thả xuống cho phép của các cấp độ khác nhau của thông tin đăng nhập.

4.3.2. Reference

Thẻ tham chiếu là lựa chọn mặc định bất cứ khi nào bạn nhập hoặc thay đổi các lệnh Selenese và các thông số trong bảng chế độ. Trong Bảng chế độ, các cửa sổ sẽ hiển thị tài liệu tham khảo trên hiện lệnh. Khi nhập hoặc sửa đổi lệnh, đã từ Bảng hoặc chế độ nguồn, đó là quan trọng để đảm bảo rằng các thông số quy định tại các mục tiêu và giá trị phù hợp với các lĩnh vực quy định trong danh sách tham số trong khung Reference. Số lượng các thông số được cung cấp phải phù hợp số quy định, thứ tự các tham số được cung cấp phải phù hợp với trình tự quy định, và loại các thông số được cung cấp phải phù hợp với loại quy định. Nếu có sự không phù hợp trong bất kỳ của ba khu vực, lệnh sẽ không chạy đúng.

4.3.3. UI-Element and Rollup

Thông tin chi tiết trên hai tấm (trong đó bao gồm tính năng nâng cao) có thể được tìm thấy trong các UIElement Tài liệu trên menu Help của Selen-IDE.

4.4. Building Test Cases

Có ba phương pháp chính cho các trường hợp thử nghiệm phát triển. Thông thường, một nhà phát triển thử nghiệm sẽ yêu cầu tất cả ba kỹ thuật.

4.4.1. Recording

Nhiều người sử dụng lần đầu bắt đầu bằng cách ghi lại một trường hợp thử nghiệm từ sự tương tác của họ với một trang web. Khi Selen-IDE là lần đầu tiên mở ra, nút ghi lại được vẽ theo mặc định.

- Lưu ý: Điều này có thể được thiết lập để tắt như là một mặc định với một phần mở rộng người sử dụng có sẵn.

Trong quá trình ghi, Selenium-IDE sẽ tự động chèn thêm các lệnh vào trường hợp thử nghiệm của bạn dựa trên hành động của bạn. Thông thường, điều này sẽ bao gồm:

- Click a link- *click* or *clickAndWait* commands
- Nhập các giá trị- *type* command
- Selecting options from a drop-down listbox - *select* command
- Clicking checkboxes or radio buttons - *click* command

Sau đây là một số “gotchas” phải biết:

- Lệnh loại có thể yêu cầu cách bấm vào một số khu vực khác của trang web cho nó để ghi lại.
- Sau một liên kết thường ghi một lệnh nhấp chuột. Bạn thường sẽ cần phải thay đổi điều này để *clickAndWait* để đảm bảo trường hợp thử nghiệm của bạn tạm dừng cho đến khi trang mới được tải xong. Nếu không, trường hợp thử nghiệm của bạn sẽ tiếp tục chạy các lệnh trước khi các trang đã được nạp tất cả các yếu tố giao diện người dùng của nó. Điều này sẽ gây ra trường hợp bất ngờ thất bại thử nghiệm

4.4.2. Adding Verifications and Asserts With the Context Menu

Trường hợp kiểm tra của bạn cũng sẽ cần phải kiểm tra các thuộc tính của một trang web. Điều này đòi hỏi khẳng định và xác nhận lệnh. Chúng tôi sẽ không mô tả chi tiết cụ thể của các lệnh này ở đây, đó là trong chương về "Selenese" Selenium lệnh. Ở đây chúng tôi chỉ đơn giản là sẽ mô tả làm thế nào để thêm chúng vào trường hợp thử nghiệm của bạn.

Với Selen-IDE ghi âm, đi vào các trình duyệt hiển thị các ứng dụng thử nghiệm của bạn và kích chuột phải vào bất cứ nơi nào trên trang. Bạn sẽ thấy một hiển thị menu ngữ cảnh xác minh và / hoặc khẳng định lệnh.

Lần đầu tiên bạn sử dụng Selenium, có thể chỉ là một trong Selenium lệnh được liệt kê. Khi bạn sử dụng các IDE tuy nhiên, bạn sẽ tìm thấy lệnh bổ sung sẽ nhanh chóng được bổ sung vào trình đơn này. Selen-IDE sẽ cố gắng để dự đoán những gì lệnh, cùng với các thông số, bạn sẽ cần phải có một yếu tố giao diện người dùng được lựa chọn trên trang web hiện nay.

Hãy xem cách làm việc này. Mở trang web lựa chọn của bạn và chọn một khối văn bản trên trang web. Một đoạn văn hoặc một nhóm sẽ làm việc tốt. Bây giờ, kích chuột phải vào văn bản đã chọn. Các menu ngữ cảnh sẽ cho bạn một lệnh *verifyTextPresent* và tham số được đề xuất phải được văn bản chính nó.

Ngoài ra, thông báo Show All Available Commands menu option. Điều này cho thấy nhiều, rất nhiều lệnh hơn, một lần nữa, cùng với các thông số đề xuất, để thử nghiệm các phần tử hiện đang được chọn giao diện người dùng của bạn.

Hãy thử một vài yếu tố giao diện người dùng hơn. Hãy thử kích chuột phải vào một hình ảnh, hoặc kiểm soát một người sử dụng như button hoặc checkbox. Bạn có thể cần phải sử dụng Show All Available Commands có sẵn để xem các tùy chọn khác hơn verifyTextPresent. Một khi bạn chọn các tùy chọn khác, những người thường được sử dụng sẽ hiển thị trên menu ngữ cảnh chính. Ví dụ, chọn verifyElementPresent cho hình ảnh sau đó sẽ gây ra lệnh có sẵn trên menu ngữ cảnh chính thời gian tiếp theo bạn chọn một hình ảnh và kích chuột phải.

Một lần nữa, các lệnh này sẽ được giải thích chi tiết trong chương lệnh Selenium. Để bây giờ, mặc dù cảm thấy miễn phí để sử dụng IDE để ghi lại và chọn các lệnh vào một trường hợp thử nghiệm và sau đó chạy nó. Bạn có thể tìm hiểu rất nhiều về Selenium các lệnh đơn giản bằng cách thử nghiệm mặc dù các IDE.

4.4.3. Editing

4.5. Running Test Cases

4.6. Writing a Test Suite

Một bộ thử nghiệm là tập hợp các trường hợp thử nghiệm đó được hiển thị trong khung bên ngoài cùng bên trái trong IDE

Chương 5. SELENIUM-RC

5.1. Introduction

Selen-RC là giải pháp cho các kiểm thử mà cần nhiều hơn các hành động trình duyệt đơn giản và dòng thực hiện. Selen-RC sử dụng toàn bộ sức mạnh của ngôn ngữ lập trình để tạo thêm các xét nghiệm phức tạp như đọc và viết các tập tin, truy vấn một cơ sở dữ liệu, và gửi email các kết quả thử nghiệm.

Bạn sẽ muốn sử dụng bất cứ khi nào Selen-RC thử nghiệm của bạn đòi hỏi logic không hỗ trợ Selen-IDE. Điều gì logic này có thể được? Ví dụ, Selenium-IDE không trực tiếp hỗ trợ:

- Điều kiện báo cáo
- Bước lặp
- Đăng nhập và kết quả báo cáo kiểm thử
- Error handling, particularly unexpected errors(lỗi đặc biệt bất ngờ)
- Kiểm tra database
- Nhóm test case
- Thực hiện lại các kiểm thử không thành công
- Những test case phụ thuộc
- Chụp ảnh của những kiểm thử có lỗi

5.2. How Selenium-RC Works

5.1. RC Components

- Selenium Server cái mà khởi động và ngắt bộ trình duyệt, biên dịch và chạy lệnh Selenium từ chương trình kiểm tra và hoạt động như một proxy HTTP, ngăn chặn và kiểm tra HTTP tin nhắn thông qua giữa trình duyệt và AUT này.
- Client libraries which provide the interface between each programming language and the

5.2.1.1. Selenium Server

Selenium Server nhận được lệnh từ Selenium chương trình thử nghiệm của bạn, biên dịch nó, và báo cáo lại cho chương trình của bạn kết quả của việc chạy những kiểm thử.

Các máy chủ RC có Selenium Core và tự động đưa nó vào trình duyệt

Điều này xảy ra khi chương trình thử nghiệm của bạn mở trình duyệt (sử dụng một thư viện hàm API của khách hàng). Selenium-Core là một chương trình JavaScript, thực sự là một tập hợp các chức năng JavaScript mà diễn giải và thực thi các lệnh Selenese sử dụng của trình duyệt được xây dựng trong JavaScript thông dịch viên

The Server receives the Selenese commands from your test program using simple HTTP GET/POST requests. This means you can use any programming language that can send HTTP requests to automate Selenium tests on the browser.

5.2.1.2. Client Libraries

Client libraries cung cấp sự hỗ trợ lập trình cho phép bạn chạy lệnh Selenium từ chương trình của chính bạn thiết kế. Có các client library khác nhau hỗ trợ cho mỗi ngôn ngữ. Một thư viện Selenium cung cấp giao diện lập trình (API) nghĩa là một tập hợp các chức năng, mà chạy lệnh Selenium từ ctr của riêng bạn. Trong mỗi giao diện có một chức năng lập trình có hỗ trợ mỗi lệnh Selenium.

Các client library có một lệnh Selenese và chuyển nó vào máy chủ Selenium để một hành động cụ thể hoặc thử nghiệm đối với các ứng dụng được thử nghiệm (AUT). Các thư viện khách hàng cũng nhận được kết quả của lệnh và chuyển nó trở lại chương trình của bạn. Chương trình của bạn có thể nhận được kết quả và lưu nó vào một biến chương trình và báo cáo nó như là một thành công hay thất bại, hoặc có thể có hành động khắc phục nếu nó là một lỗi không mong muốn.

Vì vậy, để tạo ra một chương trình thử nghiệm, bạn chỉ cần viết một chương trình chạy một tập hợp các Selenium lệnh bằng cách sử dụng thư viện API của khách hàng. Và, tùy chọn, nếu bạn đã có một kịch bản thử nghiệm Selenese tạo ra trong Selenium IDE, bạn có thể tạo ra các mã Selen-RC. Các Selen-IDE có thể dịch (sử dụng mục trình đơn xuất khẩu của nó) lệnh của nó Selenium vào các cuộc gọi chức

năng điều khiển một máy khách của API. Xem các chương Selen-IDE cho chi tiết cụ thể về xuất khẩu mã từ Selenium RC-IDE.

Selenium element

Name	Description	Default Value	Required
server (attribute)	Server name of the Selenium RC host	localhost	false
port (attribute)	Port number of the Selenium RC host	4444	false
defaultBrowser (attribute)	Selenium browser profile to use	*iexplore	false
BrowserUrl (attribute)	Base url for all selenium tests	null	true- if not supplied by <i>WebFixture</i> or <i>WebTest</i> attribute
factoryType	Type of factory object used to create custom ISelenium instances - if not supplied, a default factory is used (added 0.81)	null	true
addin (element)	JUnit addin settings	default	false
runtime (element)	Configuration settings for Selenium-RC process	default	false
browsers (element)	Custom browser aliases	default	false
userExtensions (element)	Custom javascript extensions	null	false

5.3. Installation

5.3.1. Installing Selenium Server

Selenium-RC server đơn giản là một file Java (selen-server.jar), mà không cần cài đặt đặc biệt. Chỉ cần tải file zip và giải nén các máy chủ trong thư mục mong muốn là đủ

5.3.2. Running Selenium Server

Kiểm tra bạn đã cài đặt java bằng dòng lệnh sau

```
java -version
```

Trước khi test bạn phải kiểm tra server, chỉ đường dẫn tới thư mục chứa Selenium-RC's server sau đó chạy dòng lệnh sau:

```
java -jar selenium-server.jar
```

```
java -jar selenium-server.jar -log selenium.log
```

5.3.3. Using the Java Client Driver

Hướng dẫn sau

5.3.4. Using the .NET Client Driver

Download Selenium-RC from the SeleniumHQ downloads page

- Extract the folder
- Download and install **NUnit** (Note: You can use NUnit as your test engine. If you're not familiar yet with NUnit, you can also write a simple main() function to run your tests; however NUnit is very useful as a test engine.)
- Open your desired .Net IDE (Visual Studio, SharpDevelop, MonoDevelop)
- Create a class library (.dll)
- Add references to the following DLLs: nmock.dll, nunit.core.dll, nunit.framework.dll, ThoughtWorks.Selenium.Core.dll, ThoughtWorks.Selenium.IntegrationTests.dll and Thought-Works.Selenium.UnitTests.dll
- Write your Selenium test in a .Net language (C#, VB.Net), or export a script from Selenium-IDE to a C# file and copy this code into the class file you just created.
- Write your own simple main() program or you can include NUnit in your project for running your test. These concepts are explained later in this chapter.
- Run Selenium server from console
- Run your test either from the IDE, from the NUnit GUI or from the command line

5.4. From Selenese to a Program

5.4.1. Sample Test Script

Ví dụ như bắt đầu với test với sự ghi lại của Selenium-IDE

open type clickAndWait assertTextPresent	/ q btnG Results * for selenium rc	selenium rc
---	---	-------------

5.4.2. Selenese as Programming Code

Sau đây là test script được export từ Selenium-IDE

In C#

```
using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using Selenium;

namespace SeleniumTests
{
    [TestFixture]
    public class Untitled 3
    {
        private ISelenium selenium;

        private StringBuilder verificationErrors;

        [SetUp]
        public void SetupTest()
        {
            selenium = new DefaultSelenium("localhost", 4444, "*chrome",
"http://www.google.com.vn/");

            selenium.Start();
```

```

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

selenium.Stop();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void TheUntitled 3Test()

{

        selenium.Open("/");

        selenium.Type("q", "");

        selenium.Click("btnG");

        selenium.WaitForPageToLoad("30000");

        Assert.IsTrue(selenium.IsTextPresent("Results
* for selenium rc"));

        selenium.Type("q", "ao da nu");

        selenium.Click("link=Áo Dạ thời trang 2010 -
Hàng đặt/ order - Thời trang Nữ | ĂnBạc.com");

        selenium.WaitForPageToLoad("30000");

```

```

        selenium.Click("//table[@id='table1633']/tbody/tr[2]/td[2]/font
/a/font");

        selenium.WaitForPageToLoad("30000");

        selenium.Click("//table[@id='table1815']/tbody/tr[2]/td[2]/font
/a/span/font");

        selenium.WaitForPageToLoad("30000");

        selenium.Click("//table[@id='table1815']/tbody/tr[6]/td[2]/font
/a/span/font");

        selenium.WaitForPageToLoad("30000");

        selenium.Click("//table[@id='table1815']/tbody/tr[3]/td[5]/font
/a/span/font");

        selenium.WaitForPageToLoad("30000");

        selenium.Click("//table[@id='table1820']/tbody[2]/tr[2]/td/div[
1]/font/a/img");

        selenium.WaitForPageToLoad("30000");

        selenium.Click("//table[@id='table1425']/tbody[2]/tr[4]/td/div[
1]/a/img");

        selenium.WaitForPageToLoad("30000");

        selenium.Click("//table[@id='table1420']/tbody/tr[9]/td[2]/stro
ng/span/a/font/span");

        selenium.WaitForPageToLoad("30000");

        selenium.Click("//table[@id='table2643']/tbody/tr[6]/td[2]/font
/b/font/a/span/font");

        selenium.WaitForPageToLoad("30000");

    }

}

}

```

5.5. Programming Your Test

5.5.1.C#

Selenium IDE giả định NUnit là test framework của bạn . bạn có thể thay đổi bộ trình duyệt trong câu lệnh:

```
selenium = new DefaultSelenium("localhost", 4444, "*chrome",  
"http://www.google.com.vn/");
```

Bạn có thể cho phép NUnit để quản lý thực hiện tests của bạn.

5.5.2. Learning the API

Các Selen-RC API sử dụng quy ước đặt tên đó.

5.5.2.1. Starting the Browser

In C#:

```
selenium = new DefaultSelenium( "localhost" , 4444, "*firefox" ,  
"http://www.google.com/" selenium.Start();
```

5.6. Reporting Results

Selen-RC không có cơ chế riêng của mình để báo cáo kết quả. Đúng hơn, nó cho phép bạn xây dựng các báo cáo của bạn tùy chỉnh theo nhu cầu của bạn bằng cách sử dụng các tính năng của ngôn ngữ lập trình được lựa chọn của bạn. Đó là tuyệt vời, nhưng những gì nếu bạn chỉ muốn một cái gì đó nhanh chóng đó đã làm cho bạn? Thường thì một thư viện hiện tại hoặc khung thử nghiệm có thể đáp ứng nhu cầu của bạn nhanh hơn so với phát triển thử nghiệm của riêng bạn báo cáo mã.

5.6.1. Test Framework Reporting Tools

Java có 2 test framework là JUnit và TestNG còn .NET có NUnit


5.7. Supporting Additional Browsers and Browser Configurations


Chương 6. NUnit


NUnit là một framework đơn vị kiểm nghiệm cho tất cả các ngôn ngữ Net.. Ban đầu được chuyển từ JUnit, phiên bản sản xuất hiện nay, phiên bản 2.5, là bản thứ sáu chính phát hành công cụ này dựa trên thử nghiệm xUnit đơn vị cho Microsoft. NET. Nó được viết hoàn toàn bằng C # và đã được hoàn toàn thiết kế lại để tận dụng nhiều tính năng ngôn ngữ NET,. cho các thuộc tính tùy chỉnh và khả năng phản xạ ví dụ khác liên quan. NUnit mang xUnit cho tất cả ngôn ngữ NET...

Thử nghiệm thành công là màu xanh, với một dấu. Các xét nghiệm mà sẽ được bỏ qua được đánh dấu bằng một vòng tròn màu vàng, có chứa một dấu hỏi.


Nếu bất kỳ thử nghiệm đã thất bại, họ sẽ được đánh dấu màu đỏ, với một biểu tượng-X.

Success là dấu tích màu xanh 

Failure là dấu chéo màu đỏ 

Ignored(bỏ qua) là dấu hỏi màu xám 

Inconclusive(không đi đến kết luận) là dấu hỏi màu tím 

Skipped(nhảy,bỏ) là màu xám 

Tab **Test Not Run** cung cấp một danh sách tất cả các xét nghiệm đã được chọn để chạy nhưng không chạy, cùng với các lý do.

Chương 6. SELENIUM COMMANDS

6.1. Assertion or Verification?

- Assertion sẽ không kiểm tra và hủy bỏ các trường hợp thử nghiệm hiện tại
- Verify sẽ không kiểm tra và tiếp tục chạy các trường hợp thử nghiệm.

6.2. VerifyTextPresent

VerifyTextPresent được sử dụng để xác nhận văn bản cụ thể tồn tại một nơi nào đó trên trang. Sử dụng verifyTextPresent khi bạn quan tâm đến văn bản chỉ có bản thân có mặt trên trang. Không sử dụng khi bạn cũng cần phải kiểm tra văn bản, nơi xảy ra trên trang.

Tạo ra từ isTextPresent (pattern)

Đối số:

* pattern - một mô hình để phù hợp với các văn bản của trang

Trả về:

đúng nếu mô hình phù hợp với văn bản, false nếu không

Xác nhận rằng các mẫu văn bản quy định xuất hiện ở đâu đó trên các trang kết xuất được hiển thị cho người dùng.

6.3. verifyElementPresent

Tạo ra từ isElementPresent (locator)

Đối số:

* locator - một yếu tố locator

Trả về:

true nếu phần tử là hiện nay, false nếu khác
Xác nhận rằng các yếu tố quy định là một nơi nào đó trên trang.

Sử dụng lệnh này khi bạn phải kiểm tra sự hiện diện của một yếu tố giao diện người dùng cụ thể, hơn là nội dung của nó. Xác minh này không kiểm tra các văn bản, chỉ có các tag HTML. Một thông thường sử dụng là để kiểm tra sự hiện diện của một hình ảnh.

6.4. **verifyText**

Sử dụng verifyText khi cả văn bản và giao diện phần tử của nó phải được kiểm tra. verifyText phải sử dụng locator. Nếu bạn chọn một XPath hoặc DOM locator, bạn có thể verify rằng văn bản cụ thể xuất hiện tại một địa điểm cụ thể trên trang web liên quan đến các thành phần UI trên trang.

Command	target	Value
verifyText	//table/tr/td/div/p	This is my text and it occurs right after the div inside the table.

verifyText (locator, mô hình)

Tạo ra từ gettext (định vị)

Đối số:

* locator - một yếu tố locator

Trả về:

các văn bản của phần tử

Nhận được văn bản của một phần tử. Điều này làm việc cho bất kỳ yếu tố có chứa văn bản. Lệnh này sử dụng cả các textContent (Mozilla, giống như các trình duyệt) hoặc innerText (IE như trình duyệt) của phần tử, đó là văn bản trả lại được hiển thị cho người dùng.

6.5. **Locating Elements**

6.5.1. **Locating by XPath**

XPath là ngôn ngữ được sử dụng để định vị các nút trong một tài liệu XML. Như HTML có thể là một thực hiện của XML (XHTML), Selenium người dùng có thể tận dụng ngôn ngữ này mạnh mẽ để nhắm mục tiêu các thành phần trong các ứng dụng web của họ. XPath mở rộng hơn (cũng như hỗ trợ) các phương pháp đơn giản của các vị bởi thuộc tính id hoặc tên, và mở ra tất cả các loại khả năng mới như định vị các hộp kiểm thứ ba trên trang.

