Selenium Testing Group

Lamar Sims

David Chapman

Joshuah Cook

Ryan Ramey

Austin May

Team Organization

Austin May - Team Lead

Lamar Sims - Test Analyst

David Chapman - Test Analyst

Joshuah Cook - Tester

Ryan Ramey - Tester

Team Lead

Divvied up roles for test analysts and testers. Communicated milestones to group members and followed up on their execution based on the set timetable. Explored the depths of Selenium and partitioned out roles for who will take what. Worked alongside test analysts to help gather data on the product at hand. Assembled report.

Test Analysts

Test analysts worked vehemently on the Selenium Grid (David) and Selenium WebDriver (Lamar) to understand its power and configure its ability to run. Realizing how limited Selenium was just with the Firefox plugin IDE, getting these two components jumped off were major breakthroughs that the test analysts brought to the table.

Testers

Testers were in the trenches running test cases throughout the IDE. They were understanding the components and uses of the Selenium IDE and were our eyes and ears to how smoothly the software actually runs in real time against web applications.

Table of Contents

Overview of Selenium Components……...…………………………………………………………………………………4

Terms of Reference………………………………………………………………………………………………………………….9

Test Plan………………………………………………………………………………………………………………………..………..8

Selenium IDE…………………………………………………………………………………………………………………………...8

Selenium WebDriver……………………………………………………………………………………………………………..14

Selenium Grid……………………………………………………………………………………………………………….……….15

Post Mortem……………………………………………………………………………………………………………….…………16

References……………………………………………………………………………………………………………….……………17

Appendix……………………………………………………………………………………………………………….………………18

Overview

Selenium and Selenium IDE

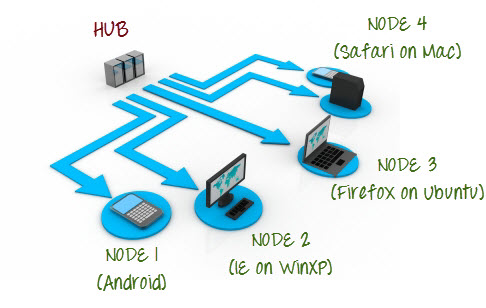
Selenium is a software testing tool used to formulate automatic test scripts for web applications. At just over a decade old, it uses a Javascript library that drives interaction with the tested website and allows users to generate test cases with the click of a button. This becomes very handy when most testers want to rerun their test cases without manually stepping through every node. This open source tool provides a test suite through an IDE that has a record feature that can generate, edit, and debug test cases on its easy-to-use interface. This can be done on any given web application. Selenium underscores the value of testing due to its ability to test across multiple platforms (such as multiple web browsers and multiple operating systems. Unfortunately, the Selenium IDE is only based in Firefox. A stifling discovery to those that build web apps for other browsers. But the solution to this is that Selenium offers a test language called Selenese that allows testers to use a number of programming languages to generate tests if they are testing outside of Firefox and the IDE. Meaning, you can write similar Java, C#, Python, PHP programs to what you see in the IDE. This eliminates having to learn new techniques for testing outside of Firefox.

Selenium had many shortcomings in its early stages. The fact that it is open source has propelled Selenium to uphold a multitude of features it didn’t have before. And that’s the beauty of Selenium. Its strongest asset is the flexibility it has to create unique automated test scripts. Through its IDE, Selenium WebDriver that drives the browser using its native support for automation. This technique has replaced the Selenium Remote Control server that is bundled with Selenium Core where it injects Javascript functions into whatever browser you’re using to drive the AUT (Application under Test). The Selenium Grid, which allows for distributed test execution. It surely provides a robust testing climate. In an era where two billion people use the internet and its traffic is constantly being overloaded, Selenium provides a great way to be sure that deliverables are functioning as expected.

Selenium Webdriver

WebDriver is an API introduced into Selenium 2.0, replacing the outdated RC (Remote Control). Key differences: the Selenium RC uses its bundled together javascript program called Selenium Core to control the browser. The Selenium WebDriver does exactly what its name refers to, it uses the browser's own engine, its own built in automation support, to drive it. WebDriver is used to write automated tests, attempting to mimic a user using a website by interacting directly with the HTML of the specified website. Automated tests for WebDriver can be written in many different languages such as Java, C#, Python, Ruby, PHP, Perl, and Javascript. The browsers supported by WebDriver are Chrome, Internet Explorer, Firefox, Opera, and HTMLUnit. WebDriver also has support for mobile, supporting Android, OperaMobile, and IPhone. The browser are executed through their own drivers written against WebDriver. The drivers are ChromeDriver, InternetExplorerDriver, FirefoxDriver, OperaDriver, HTMLUnitDriver, AndroidDriver, OperaMobileDriver, IPhoneDriver. Multiple instances of each browser can be ran at the same time as each instance is independent from other instances, by giving them their own profiles. A con of WebDriver is that it is not thread-safe, allowing multiple threads to share a reference to an instance. A limitation of WebDriver is that it has an imperfect blocking API, it cannot wait for all conditions to be met before the test proceeds because it does not know them. Selenium by default tries to wait for those (and a little bit more) on page loads via the driver.get() and element.click() methods. They are already blocking, they wait for the page to fully load and those should be working ok. The problem is with the redirects via AJAX requests and running scripts - those can't be caught by Selenium, it doesn't wait for them to finish. Also, you can't reliably catch them via readyState - it waits for a bit, which can be useful, but it will signal complete long before all the AJAX content is downloaded. This is mainly due to how JavaScript works, because it’s almost impossible to know when JavaScript is finished executing. It also cannot follow OS level operations such as a mouse click. This is only a limitation because WebDriver does provide some methods and a Wait class that can solve this problem, but having to use these can slow down the performance of the tests.

Selenium Grid

Selenium grid is a feature of the Selenium test suite that allows for parallel testing on different machines against different browsers. It allows the tester to run multiple tests at the same time against different machines running different browsers and operating systems. By utilizing the capabilities of Selenium Grid you can speed up the time it takes to run tests while expanding the reach of tested browsers and operating systems. 

There are many different ways that Grid can be useful. It can speed up the execution of a test plan by simultaneously running different test cases at the same time. It can also be used to test the same test cases simultaneously in different environments. For example, if you set up a grid with 4 nodes (see figure 1) you could run the same test on four different operating systems using four different browsers simultaneously. Incorporating and combining this functionality makes for flexible, speedy, wide-coverage testing.

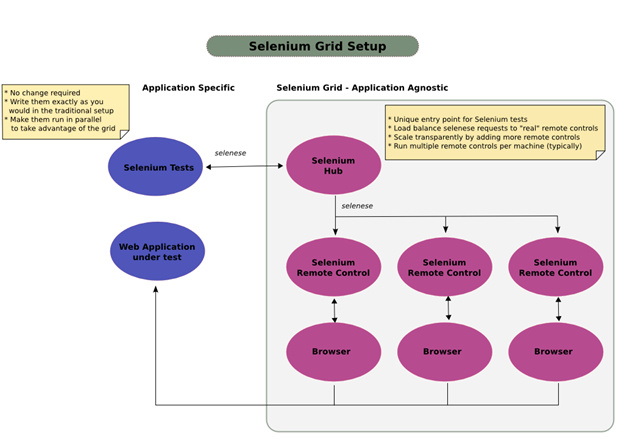


Figure 1

Selenium Grid – How it Works

First, machines that will comprise the grid must be connected to the same network. The grid is comprised of a hub connected to one or more nodes. Both of these must be started using the Selenium Standalone Server .jar file (free download). First, one machine must be registered as the hub (see Figure 2). The IP address and port number that the signal is being broadcast on must be noted so that nodes can know where to connect (see Figure 3).

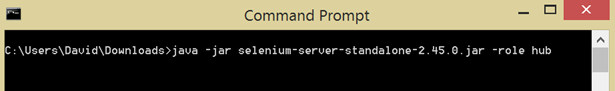


Figure 2



Figure 3

Next, a machine can be registered as a node to the hub using the hubs IP address and port number. Each node specifies what browser it will use. Test cases are generated (in whatever language is preferred) and once all nodes are registered, the hub will push test scripts out to be run on the nodes.

Selenium – Lessons Learned

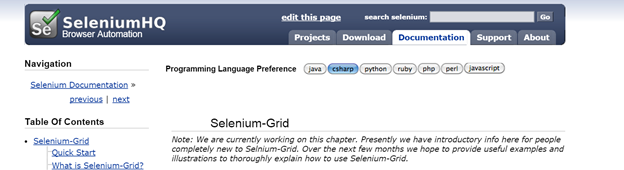


Figure 4

Selenium Grid is a powerful feature that can be of great benefit to software testing. However, like any other software, it has its drawbacks. While tutorials and videos exist that can walk you through setting up the Selenium Grid, documentation is sparse. Even the Selenium website has not updated their Grid page with very relevant information (see Figure 4). It also requires disabling of firewalls which can leave a machine susceptible to security threats. While major companies like Google, Facebook, and Salesforce utilize it, they also employ individuals with a lot of experience.

One of the biggest downsides to utilizing Selenium Grid to its fullest capabilities is the amount of time it takes to learn how to use it. After a lot of time scouring the web for information and putting bits and pieces together from here and there, we were finally able to run test cases using a grid. This trial and error research cost us a lot of time and frustration. However, given the benefits of such a powerful feature, we believe it is worth the investment.

Terms of Reference

Selenium IDE

* A Firefox-only GUI part of Selenium that allows for recording of test cases to be automated.
* It can act as a standalone component to the rest of Selenium.
* Selenium offers a test language called Selenese that can write tests in popular programming languages such as C#, Java, PHP, and Perl.
* It only functions on Firefox but has the ability to transcend to Chrome, IE, Opera, and Safari through third-party drivers.
* Features
  + Records user activity on any given base URL
  + Plays back that activity
  + Walks through test
  + Can debug and edit tests using breakpoints
  + Save tests in Selenese, Ruby scripts, or other formats

Selenium WebDriver

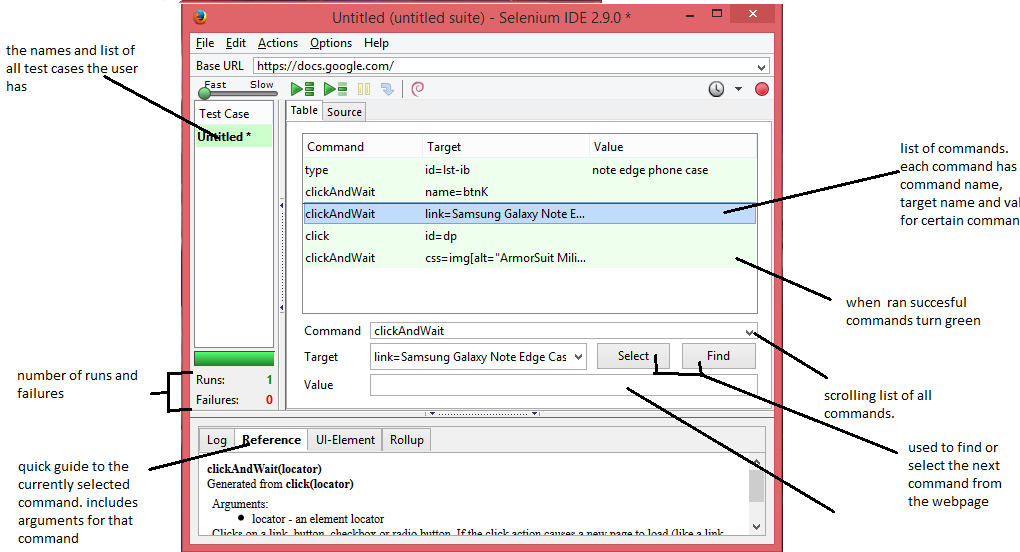
* Drives interactivity using the specific browser’s support for automation
* Interacts directly with the HTML
* Uses third-party drivers to power tests on other browsers
* Compatible with Chrome, Internet Explorer, Firefox, Opera, and HTMLUnit as well as mobile systems such as Android, OperaMobile, and iPhone.
* Problems with fully waiting on page loads because it can’t detect the time needed for the AJAX scripts being fed in.

Selenium Grid

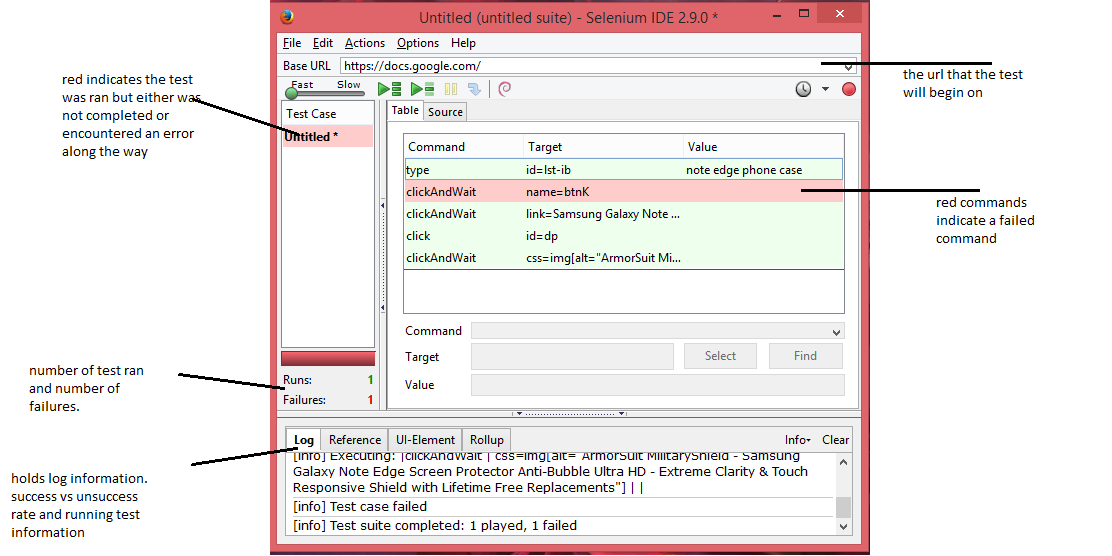
* Distributed testing execution in parallel.
* Contains one hub attached to any number of registered nodes to that hub
* The hub and nodes shake hands through the specified IP and port number
* Nodes can contain different OS and/or browsers
* Faster test cycles, reducing in less overall test times
* Must turn off firewalls

Test Plan

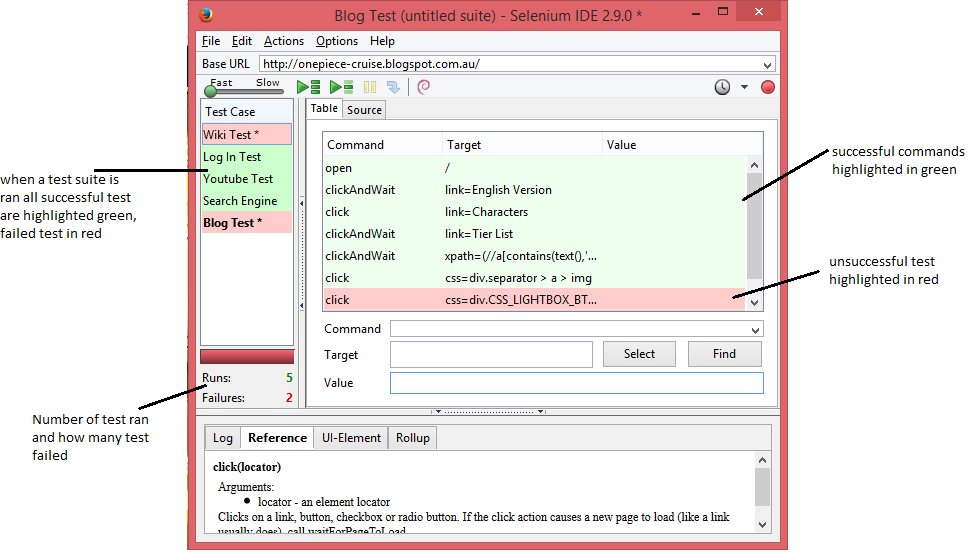
Selenium IDE

Successful test:

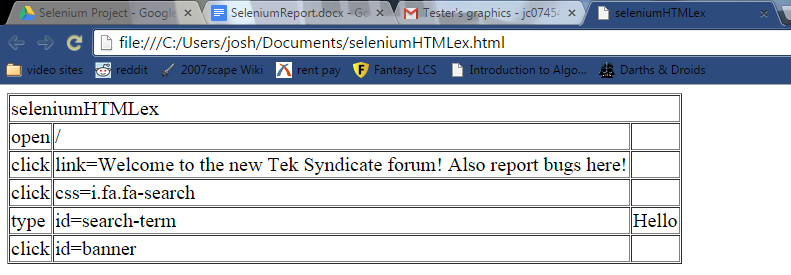
unsuccessful test:



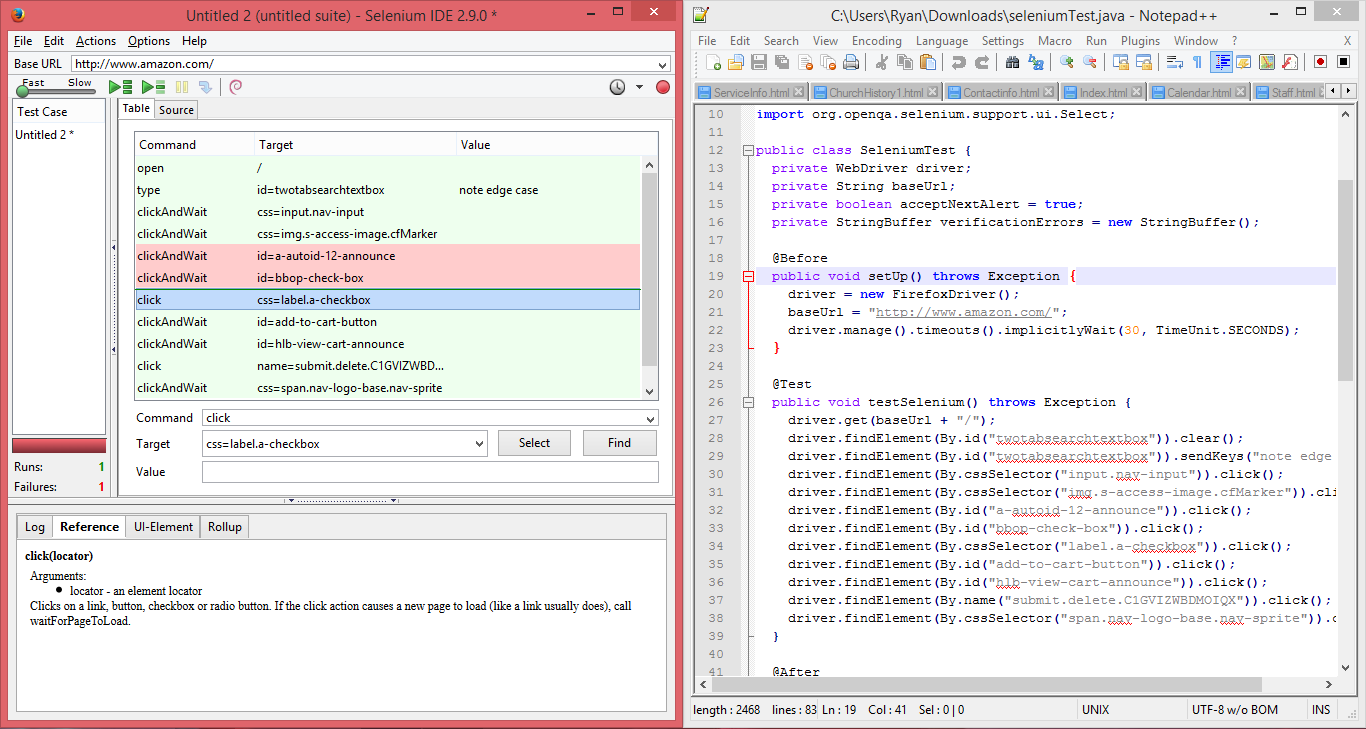
Test Suite Visual:



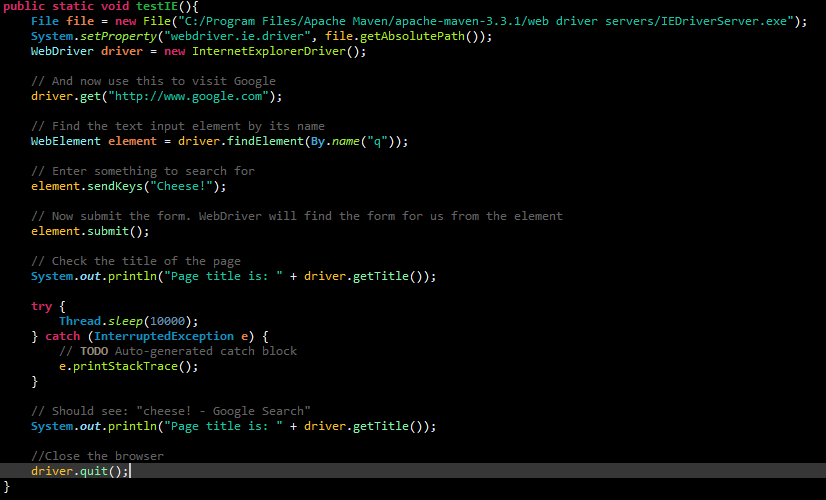
HTML File:



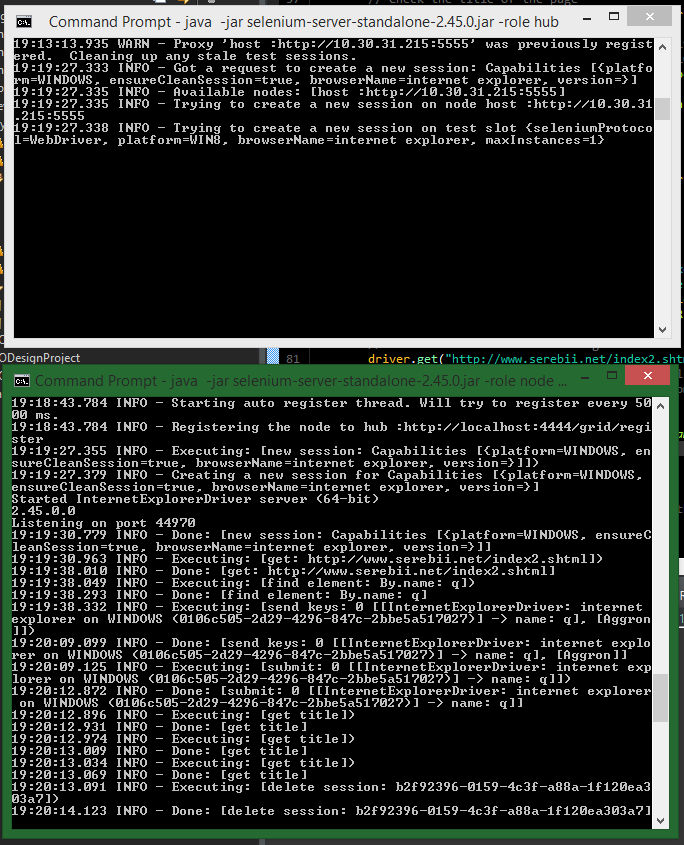
IDE vs Exported Code



Selenium WebDriver (test case)



Selenium Grid (registering nodes)



Conclusion

We have found that Selenium is a powerhouse testing tool in automating web applications. The ability to cycle through the same or different test cases on interactivity of a web page with the click of a mouse is easy and straightforward. Its provides real time feedback on whether the test case failed or passed and allowing you to breakpoint at any given test step to resolve the issue. It becomes a remarkable tool when you harness the ability to use its two other main features: the WebDriver and the Grid. For companies that scale increasingly large web applications and deliver them in high-traffic arenas, the ability to test thoroughly and robustly is important. Selenium offers the ability to test any browser with the appropriate driver, Linux, Windows, or iOS operating systems, and also create any number of test cases to assert reliability. You can also reduce test time by having the specific people from the team test with different browsers and different operating systems. This allows a grand picture of what can bugs are detected from a universal platform so that nothing slips through the cracks. Selenium is powerful, and with its open-source nature it will only continue to grow in strength.

Post Mortem

The first part of understanding Selenium was quite simple. The IDE is incredibly user friendly and takes just a little time to learn the syntax. But the most useful parts of Selenium (the WebDriver and the Grid) were by far the toughest parts to comprehend and boot. Selenium has a host of drivers that it must communicate with for it to implement its cross-platform uses. The Grid requires disabling of firewalls which can leave a machine susceptible to security threats.

Overall, when going about using testing software for the first time it is important to understand what the aim of the software is. Jumping right into the IDE without any background on what Selenium could do for us was a mistake. After scouring Selenium’s documentation it has on its website we began to understand further how to leverage Selenium. This allowed us not only to facilitate our own automated test suites, but gave us the ability to share this experience with other teams that might need it.

References

docs.seleniumhq.org/docs

Selenium Testing Tools (Beginner’s Guide)

stackoverflow.com

http://groups.google.com/groupselenium/developers

http://www.te52.com/wp-content/themes/wptellurium/tellurium/files/WhySeleniumIsntReallyFree.pdf

http://docs.seleniumhq.org/docs/07\_selenium\_grid.jsp

http://www.guru99.com/introduction-to-selenium-grid.html

http://grid.selenium.googlecode.com/git-history/22ed3ff910401af083bf06a4d13514f4c6a623ca/src/main/webapp/how\_it\_works.html

Appendix

WebDriver code for test case used by Selenium Grid

import java.io.File;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.openqa.selenium.support.ui.ExpectedCondition;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

public class Selenium2Example {

public static void main(String[] args) {

//testIE();

//testChrome();

//testIESerebiiTextInput("http://www.serebii.net/index2.shtml", "Aggron");

//testIESerebiiSelect();

/\*try {

testRE();

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}\*/

//testMatrixCalc();

//testChromeRegistration();

//Thread t = new Thread();

Thread t1 = new Thread(new Runnable(){public void run(){

try {

testSerebiiQueryGridCH();

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}});

Thread t2 = new Thread(new Runnable(){public void run(){

try {

testMatrixCalcGridFF();

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}});

Thread t3 = new Thread(new Runnable(){public void run(){

try {

testFilmClubRegGridOP();

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}});

Thread t4 = new Thread(new Runnable(){public void run(){

try {

testSerebiiSelectGridCH1();

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}});

Thread t5 = new Thread(new Runnable(){public void run(){

try {

testMatrixCalcGridIE();

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}});

t1.start();

t2.start();

t3.start();

t4.start();

t5.start();

try {

t1.join();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

t2.join();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

t3.join();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

t4.join();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

t5.join();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void testIE(){

File file = new File("C:/Program Files/Apache Maven/apache-maven-3.3.1/web driver servers/IEDriverServer.exe");

System.setProperty("webdriver.ie.driver", file.getAbsolutePath());

WebDriver driver = new InternetExplorerDriver();

// And now use this to visit Google

driver.get("http://www.google.com");

// Find the text input element by its name

WebElement element = driver.findElement(By.name("q"));

// Enter something to search for

element.sendKeys("Cheese!");

// Now submit the form. WebDriver will find the form for us from the element

element.submit();

// Check the title of the page

System.out.println("Page title is: " + driver.getTitle());

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// Should see: "cheese! - Google Search"

System.out.println("Page title is: " + driver.getTitle());

//Close the browser

driver.quit();

}

public static void testChrome(){

System.setProperty("webdriver.chrome.driver", "C:/Program Files/Apache Maven/apache-maven-3.3.1/web driver servers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

//driver.get("http://www.google.com/xhtml");

// And now use this to visit Google

driver.get("http://www.google.com");

// Find the text input element by its name

WebElement element = driver.findElement(By.name("q"));

// Enter something to search for

element.sendKeys("Cheese!");

// Now submit the form. WebDriver will find the form for us from the element

element.submit();

// Check the title of the page

System.out.println("Page title is: " + driver.getTitle());

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// Should see: "cheese! - Google Search"

System.out.println("Page title is: " + driver.getTitle());

//Close the browser

driver.quit();

}

public static void testIESerebiiTextInput(String website, String pokemon){

File file = new File("C:/Program Files/Apache Maven/apache-maven-3.3.1/web driver servers/IEDriverServer.exe");

System.setProperty("webdriver.ie.driver", file.getAbsolutePath());

WebDriver driver = new InternetExplorerDriver();

driver.get(website);

// Find the text input element by its name

WebElement element = driver.findElement(By.name("q"));

// Enter something to search for

element.sendKeys(pokemon);

// Now submit the form. WebDriver will find the form for us from the element

element.submit();

// Check the title of the page

System.out.println("Page title is: " + driver.getTitle());

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// Should see: "serebii - Google Search"

System.out.println("Page title is: " + driver.getTitle());

//Close the browser

driver.quit();

}

public static void testIESerebiiSelect(){

File file = new File("C:/Program Files/Apache Maven/apache-maven-3.3.1/web driver servers/IEDriverServer.exe");

System.setProperty("webdriver.ie.driver", file.getAbsolutePath());

WebDriver driver = new InternetExplorerDriver();

driver.get("http://www.serebii.net/pokedex-xy/");

//Get the first select element on the page and then, deselect all options first, and then select the specified option

WebElement select = driver.findElement(By.tagName("select"));

int optionsize = select.findElements(By.tagName("option")).size();

System.out.println(optionsize);

for (int i = 0; i < optionsize; i++){

boolean ex = false;

try{

select.findElements(By.tagName("option")).get(i).click(); System.out.println(-1);

}catch(org.openqa.selenium.StaleElementReferenceException e){

select = driver.findElement(By.tagName("select")); System.out.println(-2);

ex = true;

}

if (ex) {

select.findElements(By.tagName("option")).get(i).click();

System.out.println(-3);

}

try {

Thread.sleep(10000); System.out.println(-4);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println(i);

}

// Check the title of the page

System.out.println("Page title is: " + driver.getTitle());

// Google's search is rendered dynamically with JavaScript.

// Wait for the page to load, timeout after 10 seconds

(new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {

public Boolean apply(WebDriver d) {

return d.getTitle().toLowerCase().startsWith("serebii");

}

});

System.out.println("Page title is: " + driver.getTitle());

//Close the browser

driver.quit();

}

public static void testRE() throws MalformedURLException{

File file = new File("C:/Program Files/Apache Maven/apache-maven-3.3.1/web driver servers/IEDriverServer.exe");

System.setProperty("webdriver.ie.driver", file.getAbsolutePath());

DesiredCapabilities capability = new DesiredCapabilities();

capability.setBrowserName("internetexplorer");

WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

driver.get("http://www.serebii.net/index2.shtml");

// Find the text input element by its name

WebElement element = driver.findElement(By.name("q"));

// Enter something to search for

element.sendKeys("Aggron");

// Now submit the form. WebDriver will find the form for us from the element

element.submit();

// Check the title of the page

System.out.println("Page title is: " + driver.getTitle());

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Page title is: " + driver.getTitle());

//Close the browser

driver.quit();

}

public static void testMatrixCalc(){

System.setProperty("webdriver.chrome.driver", "C:/Program Files/Apache Maven/apache-maven-3.3.1/web driver servers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Go to the Google Suggest home page

driver.get("http://matrixcalc.org/en/slu.html");

WebElement matrix = driver.findElement(By.className("matrix"));

List<WebElement> entries = matrix.findElements(By.tagName("input"));

int i = 0;

for (WebElement entry: entries){

entry.sendKeys("" + i++ + "");

}

// WebElement operations = driver.findElement(By.className("bbo xslu-buttons"));

List<WebElement> ops = driver.findElements(By.className("expression"));

for (WebElement op: ops){

op.click();;

}

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

driver.quit();

}

public static void testChromeRegistration(){

System.setProperty("webdriver.chrome.driver", "C:/Program Files/Apache Maven/apache-maven-3.3.1/web driver servers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Go to the Google Suggest home page

driver.get("http://www.chadgolden.com/se/test/register");

WebElement form = driver.findElement(By.className("form-horizontal"));

List<WebElement> entries = form.findElements(By.className("form-control"));

int i = 0;

for (WebElement entry: entries){

switch(i){

case 0: entry.sendKeys("Grim Adventures");

break;

case 1: entry.sendKeys("words@email.com");

break;

case 2: entry.sendKeys("Biily");

break;

case 3: entry.sendKeys("Mandy");

break;

case 4: entry.sendKeys("CN");

break;

case 5: entry.sendKeys("CN");

}

System.out.println(entry.getAttribute("value"));

i++;

}

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

driver.quit();

}

public static void testSerebiiQueryGridCH() throws MalformedURLException{

DesiredCapabilities capability = new DesiredCapabilities();

capability.setBrowserName("chrome");

WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

// And now use this to visit Google

driver.get("http://www.serebii.net/index2.shtml");

// Find the text input element by its name

WebElement element = driver.findElement(By.name("q"));

// Enter something to search for

element.sendKeys("Aggron");

// Now submit the form. WebDriver will find the form for us from the element

element.submit();

// Check the title of the page

System.out.println("Page title is: " + driver.getTitle());

// Google's search is rendered dynamically with JavaScript.

// Wait for the page to load, timeout after 10 seconds

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// Should see: "serebii - Google Search"

System.out.println("Page title is: " + driver.getTitle());

//Close the browser

driver.quit();

}

public static void testSerebiiSelectGridCH1() throws MalformedURLException{

DesiredCapabilities capability = new DesiredCapabilities();

capability.setBrowserName("chrome");

WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

// And now use this to visit Google

driver.get("http://www.serebii.net/pokedex-xy/");

//Get the first select element on the page and then, deselect all options first, and then select the specified option

WebElement select = driver.findElement(By.tagName("select"));

int optionsize = select.findElements(By.tagName("option")).size();

System.out.println(optionsize);

for (int i = 0; i < optionsize; i++){

boolean ex = false;

try{

select.findElements(By.tagName("option")).get(i).click();

}catch(org.openqa.selenium.StaleElementReferenceException e){

select = driver.findElement(By.tagName("select"));

ex = true;

}

if (ex) {

select.findElements(By.tagName("option")).get(i).click();

}

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//System.out.println(i);

}

// Check the title of the page

System.out.println("Page title is: " + driver.getTitle());

// Google's search is rendered dynamically with JavaScript.

// Wait for the page to load, timeout after 10 seconds

(new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>() {

public Boolean apply(WebDriver d) {

return d.getTitle().toLowerCase().startsWith("serebii");

}

});

// Should see: "Serebii - Google Search"

System.out.println("Page title is: " + driver.getTitle());

//Close the browser

driver.quit();

}

public static void testMatrixCalcGridFF() throws MalformedURLException{

DesiredCapabilities capability = new DesiredCapabilities();

capability.setBrowserName("firefox");

WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

driver.get("http://matrixcalc.org/en/slu.html");

WebElement matrix = driver.findElement(By.className("matrix"));

List<WebElement> entries = matrix.findElements(By.tagName("input"));

int i = 0;

for (WebElement entry: entries){

entry.sendKeys("" + i++ + "");

}

// WebElement operations = driver.findElement(By.className("bbo xslu-buttons"));

List<WebElement> ops = driver.findElements(By.className("expression"));

for (WebElement op: ops){

op.click();;

}

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

driver.quit();

}

public static void testMatrixCalcGridIE() throws MalformedURLException{

DesiredCapabilities capability = new DesiredCapabilities();

capability.setBrowserName("internetexplorer");

WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

driver.get("http://matrixcalc.org/en/slu.html");

WebElement matrix = driver.findElement(By.className("matrix"));

List<WebElement> entries = matrix.findElements(By.tagName("input"));

int i = 0;

for (WebElement entry: entries){

entry.sendKeys("" + i++ + "");

}

// WebElement operations = driver.findElement(By.className("bbo xslu-buttons"));

List<WebElement> ops = driver.findElements(By.className("expression"));

for (WebElement op: ops){

op.click();;

}

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

driver.quit();

}

public static void testFilmClubRegGridOP() throws MalformedURLException{

DesiredCapabilities capability = new DesiredCapabilities();

capability.setBrowserName("opera");

WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

// Go to the Google Suggest home page

driver.get("http://www.chadgolden.com/se/test/register");

WebElement form = driver.findElement(By.className("form-horizontal"));

List<WebElement> entries = form.findElements(By.className("form-control"));

int i = 0;

for (WebElement entry: entries){

switch(i){

case 0: entry.sendKeys("Grim Adventures");

break;

case 1: entry.sendKeys("words@email.com");

break;

case 2: entry.sendKeys("Biily");

break;

case 3: entry.sendKeys("Mandy");

break;

case 4: entry.sendKeys("CN");

break;

case 5: entry.sendKeys("CN");

}

System.out.println(entry.getAttribute("value"));

i++;

}

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

driver.quit();

}

}