Team 7: Selenium

CSCI 4534

Instructor: Dr. Vladan Jovanovic

Team Members:

# David Chapman

# Josh Cook

# Austin May

# Ryan Ramey

# Lamar Sims

# Contents

Team Organization...……………………………………………………………………………...3

SQA Plan………...………………………………………………………………………………..5

Software Requirement Specifications……………………………………………………………..7

Test Plan…………………….…………………………………………………………………....9

Testing Approach………………………………………………………………………...11

Comparison of Features………………………………………………………………….12

Incident Report…………………………………………………………………………..13

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

References……………………………………………………………………………………….18

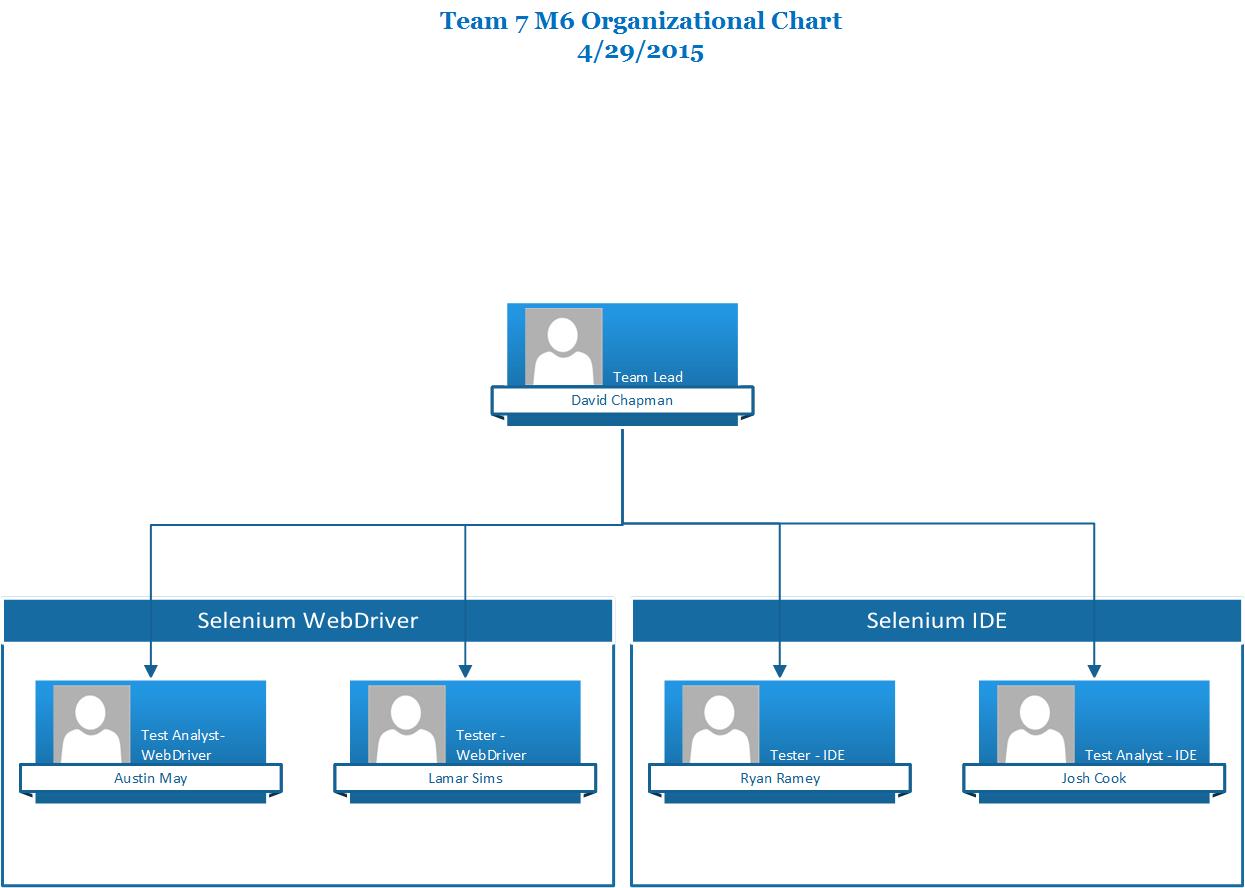
Appendix………………………………………………………………………………………...19

Selenium IDE……………………………………………………………………………19

Selenium WebDriver…………………………………………………………………….26

## 

# Team Organization



## Team Lead - David Chapman

1. Responsible for organizing/coordinating team members
2. Responsible for generating the test plans
3. Responsible for assigning tasks to Test Analysts and providing support when needed
4. Responsible for assigning tasks to Testers and providing support when needed
5. Organization/compilation of testing documents
6. Aids in the development and execution of WebDriver tests

## Test Analyst - Austin May

1. Responsible for organizing/coordinating WebDriver testing efforts
2. Responsible for overseeing test cases of the Webdriver
3. Aid WebDriver Tester(s) when needed.
4. Assist the Team Leader with organization/compilation of testing documents

## 

## Test Analyst - Josh Cook

1. Responsible for organizing/coordinating IDE testing efforts
2. Responsible for overseeing test cases of the IDE
3. Aid IDE Tester(s) when needed.
4. Assist the Team Leader with organization/compilation of testing documents

## Tester - Lamar Sims

1. Performing WebDriver testing tasks as specified by the Team Leader
2. Execute WebDriver test scripts
3. Observation and execution of WebDriver test results
4. Maintenance and retention of WebDriver test results

## Tester - Ryan Ramey

1. Performing IDE testing tasks as specified by the Team Leader
2. Execute IDE test scripts
3. Observation and execution of IDE test results
4. Maintenance and retention of IDE test results

# 

# 

# SQA Plan

1. Introduction

This Software Quality Assurance Plan will usher in the process and procedures for performing Software Quality Assurance for the Selenium Testing project.

1.1 Purpose and Scope

This plan provides the skeleton for which quality assurance activities will be handled and managed and is based on the two main components of Selenium, our work products. It maps out the requirements of the testing team, defines the milestones of the testing plan and how they will be conducted, and identifies the SQA work products.

1.2 Reference Materials

All relevant information that will progress our quality assurance will be the Selenium documentation found on Selenium’s website [www.seleniumhq.org](http://www.seleniumhq.org)

1.3 Project Checkpoints

Because our product has already been deployed, we have checkpoints atypical of a regular ongoing development process. The checkpoints will consist of testing the components of Selenium, the IDE and the WebDriver. At the end of each we will consider a checkpoint completed. Each component will be thoroughly diagnosed to determine it’s functionality. The testing team will then fulfill the items to be tested, which are mapped out by the test analysts, to assure quality functionality in the software. All outstanding issues will be documented in the Incident Report. Because this plan seeks to find the bugs and not necessarily resolve them, outstanding issues will remain dormant in the Incident Report.

2. Activities

All tasks will be completed by members of Group 7 with each person’s role being defined uniquely and deliberately. The tasks will be conducted within the scope of the SQA plan by personnel working under the domain of a specific work product. The SQA will assert the Selenium IDE can perform necessary tests on simplistic and complicated elements of a web page that are defined in the Items to be Tested table (see table). The SQA will also certify that the WebDriver is able to mimic the commands of the IDE (using Selenium’s built-in Selenese language) using different web browser’s through its different built-in automation support.

3. SQA Program Requirements

This section assures that Selenium is in accordance with our quality assurance plan. Because the quality plan is devoid of contractors and SQA personnel that can sign-off on the tasks, the test cases will be tested by our testers and examined by our test analysts to achieve the purpose of our SQA plan defined in the scope.

3.1 SQA Program Audits

Because we have no compliance with a specific bureau of testing, we deemed auditing unnecessary.

3.2 SQA Status Reports

Team Lead will assign roles for each tester. Testers will test using the tools defined in section 4 and report any nuances with the actual and expected result in the Incident Report. Testers will document what the result of each case manifests, and Test Analysts will compile the data in the testing plan.

3.3. Software Development Process

Because Selenium is an open-source platform, development is in constant motion. Each life-cycle phase is thus not audited. Our plan is to certify that the current, most up-to-date version of the software is tested against its core functionality.

3.4 Project Reviews

In the first iteration of using the software, the team’s familiarity of Selenium was pronounced. Recognizing some pitfalls such as wait times for dynamic AJAX requests, we were able to become familiar with Selenium’s drawbacks. This enabled some of our test cases which we were then able to report and document for potential resolutions. This is the scope of our project review. Using the product initially, then drilling down into the system to see what all of its defects are.

3.5 Tools and Techniques of Tests

Junit tests to run through our test cases for the WebDriver component. Instead of testing a particular website to show Selenium’s power, elements of the Selenium’s testing IDE were used on a website to highlight its varying functionality. As a result, these tests showed whether it matches a pass or fail criteria.

Black box testing against the documented functionality of the Selenium IDE. Built-in test cases were supplied with the source code for the WebDriver. Running those proved various elements of the WebDriver were running as reported.

3.6 Problem Reporting

All AUT will be tested with their results being matched with the expected result of the test. Any discrepancies go into the Incident Report log that will remain dormant for the time being since we are not looking to deploy this software, only test it.

# 

# Software Requirement Specifications

Selenium is a portable software testing framework for web applications. The software is open source and available to download for free. Used by many industry leaders, Selenium provides an easy way to automate websites and web applications. The Selenium suite of testing tools is comprised of two main elements: Selenium IDE and Selenium WebDriver.

The Selenium IDE is a plug in for the Firefox web browser. It allows the user to record events that happen in the browser, play them back, edit and debug issues. Using the IDE, the user can create quick bug reproduction scripts and create scripts to aid in automation-aided exploratory testing.One of the best features of Selenium is how simple it is to install and use. The IDE can be downloaded from the selenium website or the firefox add-ons page. The IDE is easy to use and requires no coding experience. The record function allows users to use the web in a normal manner while the IDE records their actions in steps. The user can then replay the test case. The IDE allows the user to create multiple test cases and run the test suite, the entirety of the test cases, to check multiple cases at one time. Selenium has an export function that generates the code for the test cases in multiple languages including Java, C# and Ruby. One shortcoming of the IDE is that it does not allow the user to loop test cases. Another large shortcoming of the Selenium IDE is that it is only available with firefox web browser which limits the scope of testing to the firefox web browser.

The Selenium Webdriver is an Application Program Interface that allows the user to drive a browser natively either locally or on a remote machine using the Selenium Server. The Webdriver is an upgrade from the existing Selenium Remote Control (RC). The RC server bundles Selenium Core and automatically injects it into the browser. The thing to note is that Selenium Core is just a set of Javascript functions that interprets Selenese commands - Selenium’s own testing language - from your testing program. It does this by leveraging the browser’s built-in javascript interpreter. And as a result you can execute test cases in languages such as Java, C#, and Python allowing you to test outside of Firefox. 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. To execute, it uses the designated drivers (ChromeDriver, InternetExplorerDriver, FirefoxDriver, etc.) that must be downloaded first.

In order to use Selenium Webdriver it requires a very lengthy setup. Selenium Webdriver can be setup for many different languages, operating systems, and IDEs, but this will focus on how to setup Selenium for use with Java, Windows, and Eclipse. In order to use Selenium Webdriver on a machine it must have these downloads:

* A download of Java, specifically the Java JDK
* A version of Apache Maven compatible with your Java JDK
* A version of Eclipse
* m2eclipse, Eclipse plug-in, compatible with your Java JDK and Eclipse version
* The respective drivers for the browsers that are to be used.

After all downloads have been made and installed, the machine’s system path must be edited to include the \bin of the Java JDK and Apache Maven. To use Maven and install all of Selenium Webdriver’s dependencies there will need to be the creation of a Maven pom.xml configuration file. The dependencies for using Selenium Webdriver on Eclipse can now be downloaded from Maven servers. These dependencies can now be imported into Eclipse as a Java Maven project, from there Selenium Webdriver test can now be created and used to test websites on multiple web browsers. Writing out Selenium Webdriver test cases in Java, can be a fairly simple process once the basics of the Webdriver’s interface are understood. And most complications in creating test cases come from the HTML used to create the website, if it is poorly written, making it difficult to write out easy and efficient Webdriver test cases against that HTML.

Selenium WebDriver supports the following browsers along with the operating systems these browser a compatible with:

* Google Chrome
* Internet Explorer 6, 7, 8, 9, 10 -32 and -64 bit where applicable
* Firefox: latest ESR, previous ESR, current release, one previous release
* Safari
* Opera
* HtmlUnit
* phantomjs
* Android (with Selendroid or appium)
* iOS (with ios-driver or appium)

# Test Plan

Items to be Tested

|  |  |
| --- | --- |
| Item to be Tested | Version Number |
| Selenium IDE | Version 2.9 |
| Selenium WebDriver | Version 2.45.0 |

Features to be Tested

|  |  |
| --- | --- |
| Item being Tested | Business Scenarios being Tested |
| Selenium IDE | Run any test suite on fast AND slow |
| Selenium IDE | Use ‘Assert’ for a failed test case |
| Selenium IDE | Use ‘Verify’ for a failed test case |
| Selenium IDE | Find element on page using Xpath locator |
| Selenium IDE | Test login of a website (function test) |
|  | Use a start point and place after log in feature so you only have to log in one time |
| Selenium IDE | Use an insert command in the middle of test suite |
| Selenium IDE | Input data into appropriate fields (textboxes, status updates, etc.) |
| Selenium IDE | Run test suite from the middle |
| Selenium IDE | Wait for a page load using clickAndWait |
| Selenium IDE | Make sure specific element is present in specific location |
| Selenium IDE | Glob pattern with specific piece of text |
| Selenium IDE | Regular Expression pattern with specific text |
| Selenium IDE | Use WaitFor command to find an element dynamically |
| Selenium IDE | Use an echo statement to print something to test output |
| Selenium IDE | Assert some type of pop up window using assertAlert, assertConfirmation, or assertPrompt |
| Selenium WebDriver | Locating and verifying an Ajax element |
| Selenium WebDriver | Testing multiple instances of browsers |
| Selenium WebDriver | Cycling through a combo box on web page |
| Selenium WebDriver | Test pop up window |
| Selenium WebDriver | Return current URL |
| Selenium WebDriver | Return title of page |
| Selenium WebDriver | Return source code of page |

Features NOT to be tested

|  |  |
| --- | --- |
| Item being Tested | Business Scenarios NOT being Tested |
| Selenium Grid | Selenium Grid component |
| Selenium IDE | Exact Pattern |
| Selenium IDE/Selenium WebDriver | Condition statements |
| Selenium IDE | Locating an element by name/ID/DOM/CSS |
| Selenium IDE | Storing commands and Selenium variables |
| Selenium IDE | Javascript usage with script/non-script parameters |
| Selenium WebDriver | Locating UI elements on a page |
| Selenium WebDriver | Output cookies for specific URL |

# Testing Approach

The project is using a waterfall approach since the software has already been created and deployed. The waterfall methodology allows for little flexibility during the software’s life cycle. However, since we are only focusing on one component (testing the software) the waterfall model works best because we can statically test the software without worrying about constant updates or changes to the software like, say, an agile process would.

Exploratory testing is our main approach to testing the Selenium software. Because we are having to simultaneously learn, design and execute the necessary tests we feel this approach is best. This approach depends on the ability to formulate quality test cases during easy and difficult situations all while learning how the software actually works. Test cases for planned functionality will be created and added to the Test Items tables which we will later run and review.

While we run the tests we will have the input and expected results for each test case and comparing them to the actual output of the test script. Then we will record each iteration of our test case and put them in our Test Log. Any discrepancies between actual and expected results will be put into the Incident Report.

After we have run the tests we will review the discrepancies and determine what effect they will have on scenarios and what it could mean for a business using this software.

# Comparison of Features

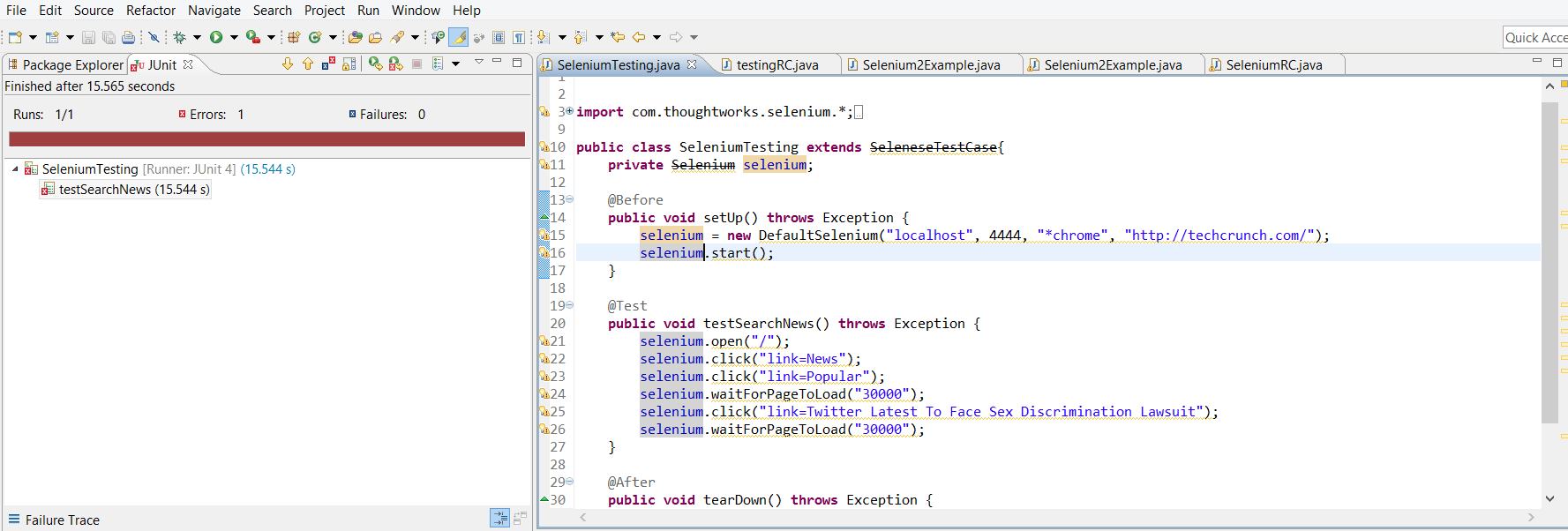
|  |  |
| --- | --- |
| Selenium IDE | Selenium Webdriver |
| Support FireFox browser | Supports FireFox, IE, Chrome, Safari, Opera, HTMLUnit browsers |
| Record and Playback Supported | Record and Playback Not Supported |
| Does not require Selenium server to run tests scripts | Does not require Selenium server to run tests |
| GUI Plugin for Firefox | Core API with bindings in multiple languages |
| Core engine is Javascript based | Interacts natively with browser application |
| Not object oriented | API’s are entirely object oriented |
| Does not support moving of mouse cursors | Supports moving of mouse cursors |
| Need to append full xpath syntax | Does not need to append full xpath syntax |
| Does not support listeners | Supports implementation of listeners |
| Does not support test of mobile applications | Supports test of mobile applications |

**Incident Reporting**

**Test Case 1: Test for Ajax**

A bug we found in the Webdriver is when it waits for a page load. Sometimes when you launch your testing suite Junit will throw an error because it couldn’t find the elements within the necessary page (see Figure 1). Because when you’re trying to load a page the built-in command ‘readyState’ will wait for a bit, but will signal complete long before all AJAX content is loaded. Selenium can’t catch the redirects via AJAX requests and running scripts because it doesn’t wait for them to finish. So the tester must tinker around with how he or she wants to go about working around this issue. They can create an implicit wait. This specifies the amount of time the driver should wait when searching for an element if it is not immediately present. So essentially they can customize the wait time to wait for 10 seconds which (hopefully) is plenty of time for the content to be loaded. If the page loading exceeds 10 seconds it will throw an exception. Using these can slow down the performance of tests, however, there is no general solution that can work for everyone. So, in essence, there is no universal means to loading a page optimally.

(Figure 1)



**Test Case 2: Test Pop Up/Alerts**

There was a bug when interacting with pop-ups/alerts. We expected when we clicked on an object it would bring up the alert and then we would be able to interact with the alert to accept or cancel it. What we actually got was when we brought the alert up we could not interact with it, so we could not accept or cancel it.

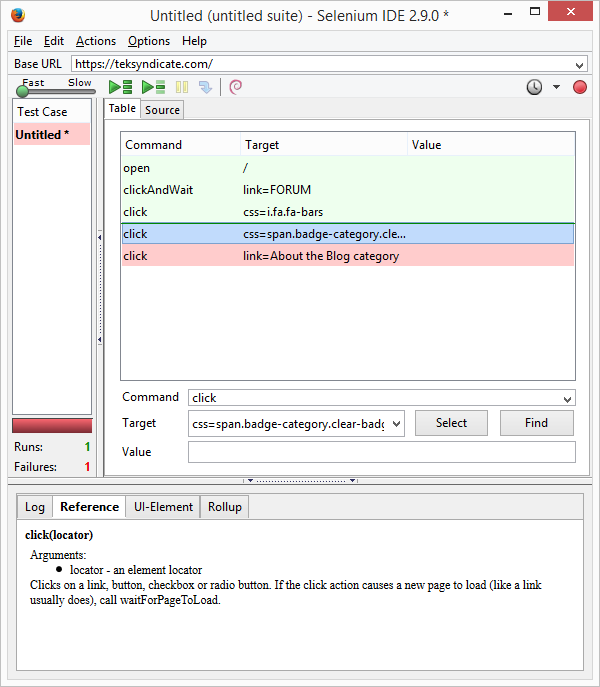
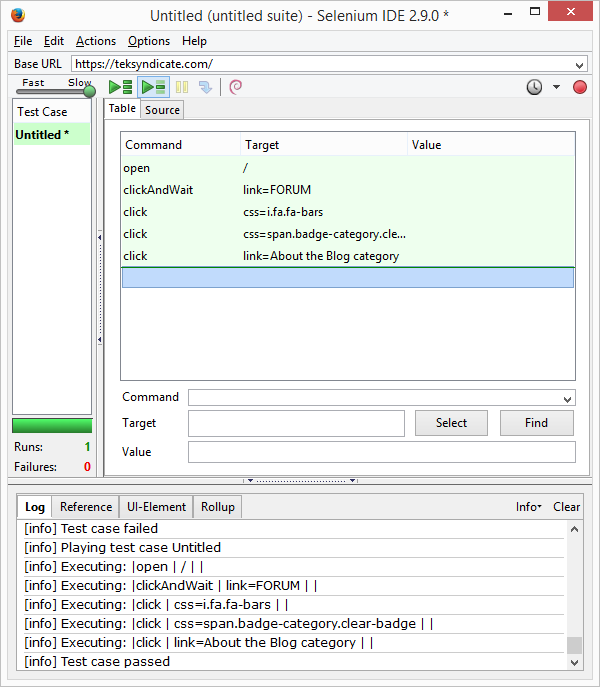
**Test Case 4: Test Looping through Select tags**

While this test case displays doing what we intended it to do, initially it did not. This test case is expected to interact with the ‘select tag’ on the website and then loop through each of the objects in the ‘select tag’. What we initially got was a thread racing error, where the internal blocking mechanism for the Webdriver could not fully wait until the webpage was done loading because of the Javascript. So it would attempt to grab an object that was not there yet. We were able to fix this by explicitly telling the program to wait 10 seconds, before trying to grab the next object in the ‘select tag’.

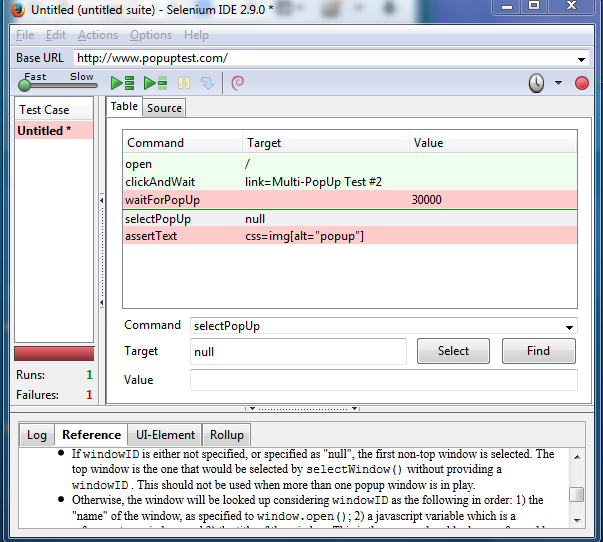
**Test Case IDE-1: IDE on Fast and Slow**

When running tests with Selenium, we noticed that some tests could run on any speed without issue, but others required more time for things to run. Things like selecting boxes or typing in data fields can be done almost instantaneously, but using commands like click, clickAndWait, or Element checkers required waiting until things had loaded. An example failed case is provided.

fast: slow:

**Test Case IDE-16- Manipultion of Windows**



Selenium IDE encounters trouble when trying to select a pop up window to perform actions on. Selenium will allow you to assert and verify items in a pop up window but will not close the window. This problem seems to be rooted in the select window function which only seems to select other tabs or instances of firefox that the user has initiated themselves. Selenium defaults the first non-top window when dealing with pop ups that have no windowID. the Selenium IDE notes “(don't rely on this if you are working with multiple popups simultaneously)”.

# 

# Post Mortem

Selenium is a powerful suite of automated web testing tools that allows for fast and simple web testing. While most functionality of web sites and web-based applications are handled, Selenium still has some issues with more modern functionality. Through this test plan, we have demonstrated some of the differences in capability between Selenium IDE and Selenium WebDriver. While the WebDriver feature allows the tester much more flexibility and in-depth testing ability, they also share common problems. For example both the IDE and WebDriver seem to struggle dealing with pop up windows. Ultimately, the WebDriver provides much more customization and a more thorough testing experience.

## Lessons Learned

## 

**David Chapman**

This project has taught me a great deal about the importance of implementing a strategy for software testing and quality assurance. I have gained much more insight into the differences between black box and white box testing. Creating scripts that are designed to test the specified areas was more of a challenge than I had initially anticipated. Some test were run and failed, although the failure was on the part of the team. We had to figure out a way to make it work and that experience has been invaluable.

Through working on this project, I have learned the value of organization. While tasks independently may cut down on confusion, teamwork is necessary to accomplish large tasks. This project was a large undertaking and took many hours to complete with everyone involved. Trying to complete this individually would have most definitely been impossible in the time frame given. Overall, I was very pleased with our ability to pull together and overcome the constraints to put this together. Working as part of a team has been the most valuable lesson I have learned.

**Austin May**

There’s an incredible difference between thinking something is a quality deliverable and actually jumping through the hoops to prove it’s a quality deliverable. Seeing how a good software testing report is modeled, manifesting the report has taught me that software testing is not about tinkering around with what developers have built. It’s about keeping pace with them at every turn through the development process and auditing what goes wrong so that they may fix it before they make conglomerate changes in the next iteration. It’s about having requirements and standards that the software must meet. The software testing practices we have learned in class can and should all be used in a highly complex system. Especially when that system has lives depending on it. Reliability is not something you can deem a success in one batch of tests. You must test it in many different scenarios. Quality assurance has taught me that you can’t expect to ship a deliverable without backing it up with merit of countless hours of testing. Testing is what gives software its integrity.

**Josh Cook**

Taking on a project like this was a really interesting experience for me, it allowed me to really expand my knowledge about web applications and how they run. In order to test something, you have to understand it after all. I was exposed to many browsers and operating systems I had little experience in, and got a lot of good practice in using them. Working with a real application that is used out in the private sector was really fun as well, and let me see how groups work together on projects just like this out there.

**Ryan Ramey**

**Lamar Sims**

This whole testing process has been a completely new and important experience. It has completely changed what I thought I knew about testing a project, throughout my college career. It has really shown me that you really need to put effort and thinking into your testing, instead of just randomly coming at the project from all sides. I always thought testing was just a boring task and mindless task, but this has just completely proved me wrong, if done right in a constructive and technical manner, it can be really fun. Working with professional grade applications was also a huge eye opener, and the need for using different applications for different situations. Overall it was all really interesting and a very important learning experience, one that I am very thankful for and won’t be forgetting any time soon.

# References

1. <http://www.seleniumhq.org/about/history.jsp>

2. <http://www.seleniumhq.org/docs/03_webdriver.jsp>

3. <https://code.google.com/p/selenium/>

4. <https://sites.google.com/a/chromium.org/chromedriver/>

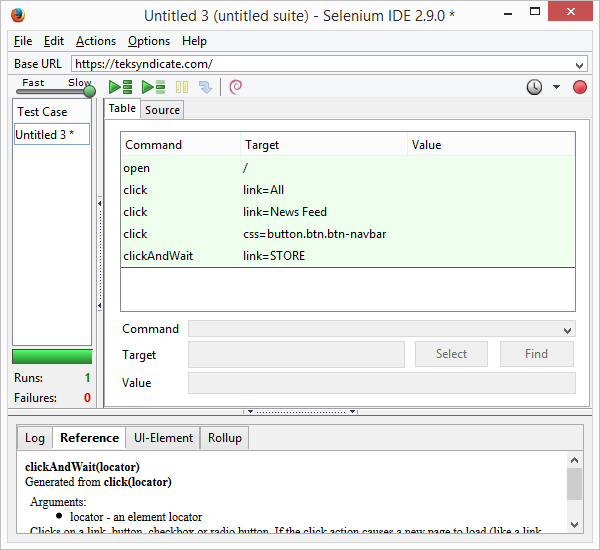
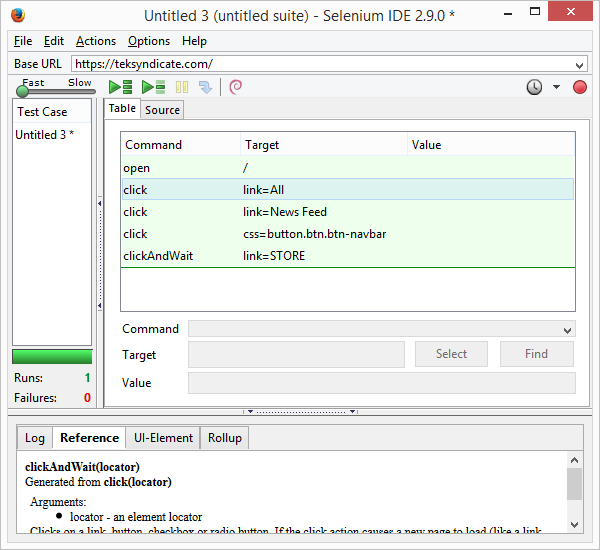
5. <https://code.google.com/p/selenium/wiki/InternetExplorerDriver>

# 

# Appendix

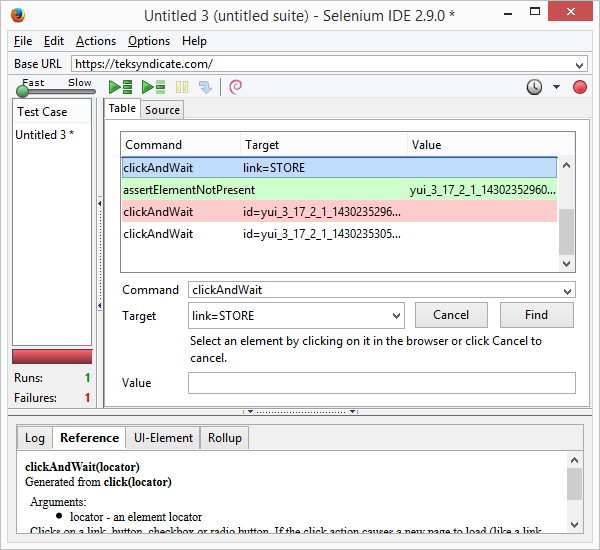
**Selenium IDE Test Cases**

**IDE-1: Run any test suite on fast AND slow**



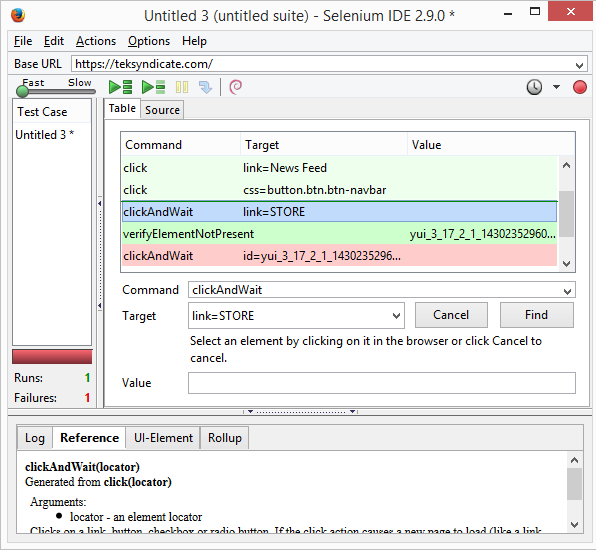
This was a case that could run on either speed, slow or fast

**IDE-2: Use ‘Assert’ for a failed test case**



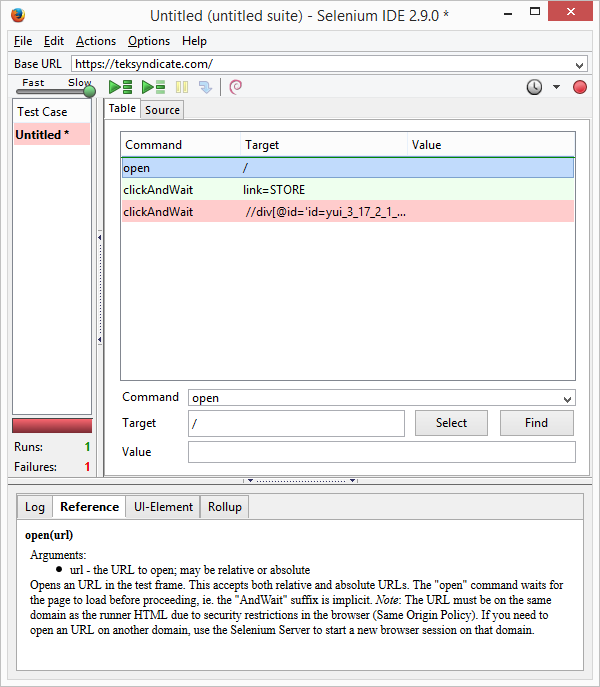
assertElementNotPresent shows the element didnt load before the test tried to click it

**IDE-3: Use ‘Verify’ for a failed test case**



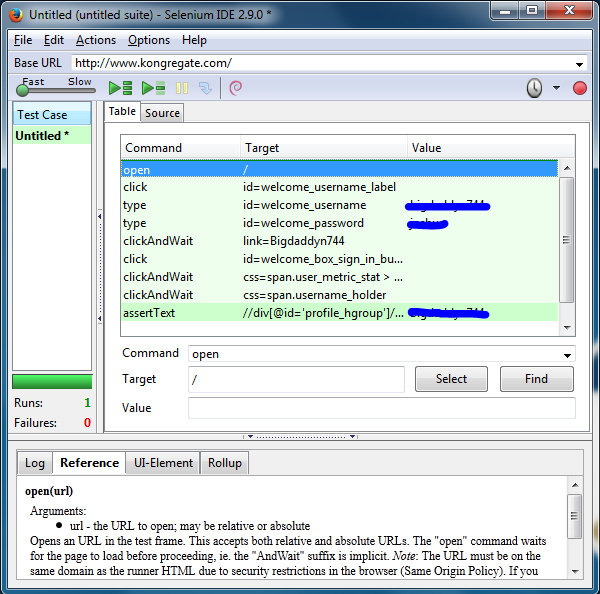
same as assert

**IDE-4: Find element on page using Xpath locator**



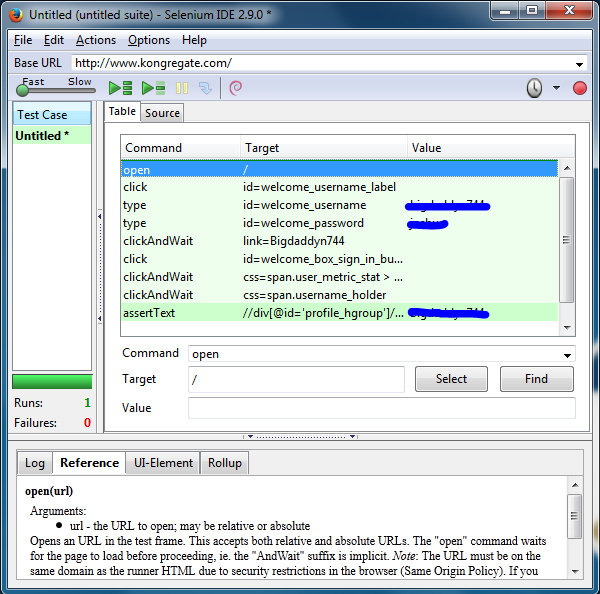
I believe it was my xpath that was incorrect, but I was unable to get the xpath to work.

**IDE-5: Test login of a website (function test)**



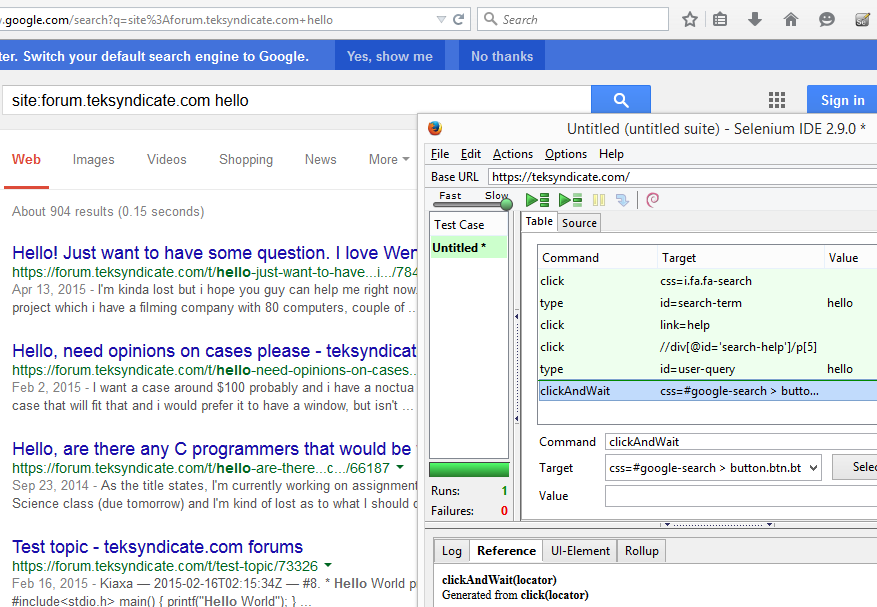
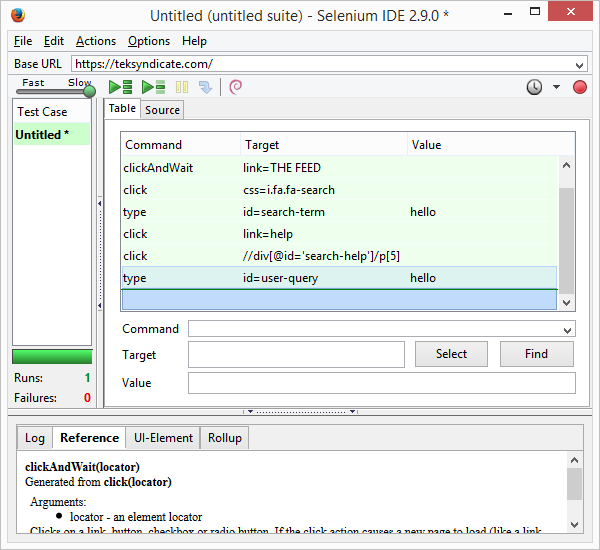
Selenium IDE encounters no problem with logging in to a website. There is one big downside to selenium IDEs login ability. It does not encode password and username information. This leaves the user in a very vulnerable position. I took the precaution of marking over the username and password in my screenshot but that is not possible in the IDE. If the user wishes to run or show off the test case for another person with the IDE then the other person viewing will be able to see that information.

**IDE-6: Use a start point and place after login feature so you only have to login one time**



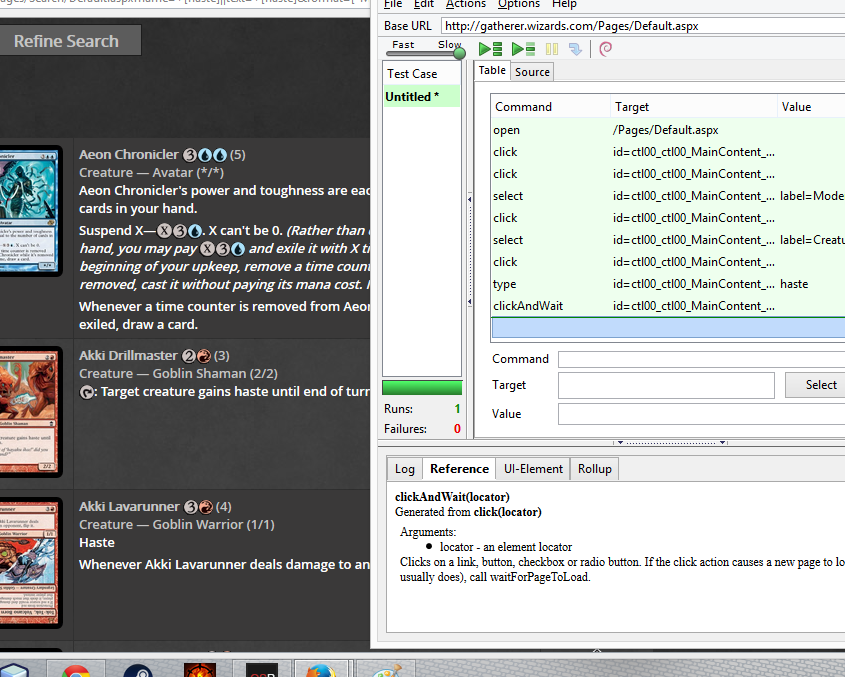
In this case i was testing to see if Selenium IDE would be able to deal with having login instructions but the browser is already logged in to an account. Selenium handled the situation perfectly. When logged in the webpage will not generate the boxes that accept the user’s login information. I believed that this would cause a hiccup in the Selenium IDEs process. The IDE however simply steps pass the login steps and continues on with the other given tasks.

**IDE-7: Use an insert command in the middle of test suite**

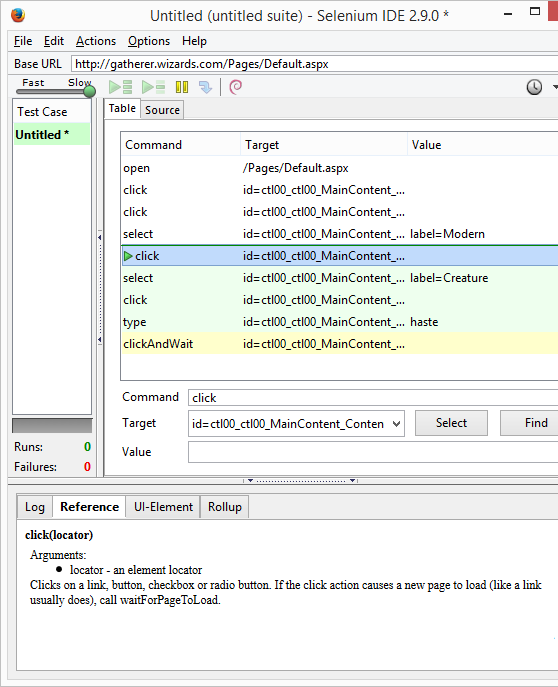


a clickAndWait was inserted onto the end of this case, extending the case

**IDE-8: Input data into appropriate fields (textboxes, status updates, etc.)**



**IDE-9: Run test suite from the middle**



by inserting a new start point you can change where the case starts

**IDE-10: Wait for a page load using clickAndWait**



the clickAndWait is used when you have to wait for a page load after the click

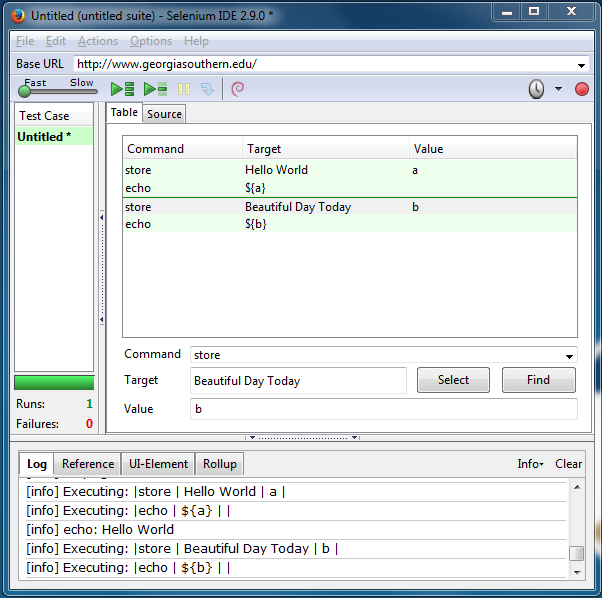
**IDE-11: Make sure specific element is present in specific location**

**IDE-12: Glob pattern with specific piece of text**

**IDE-13: Regular Expression pattern with specific text**

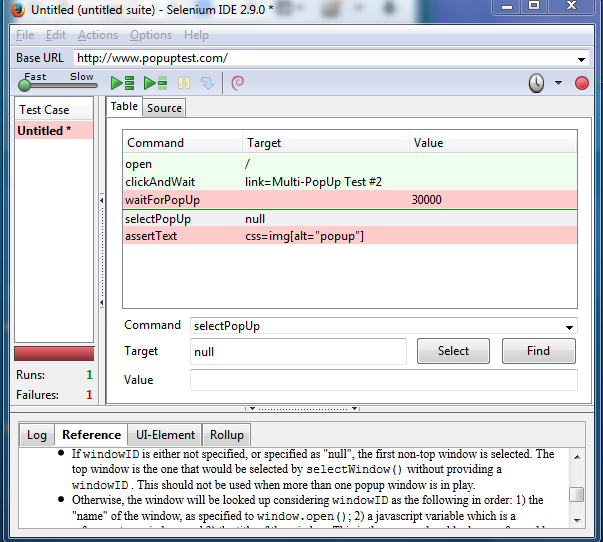
**IDE-14: Use WaitFor command to find an element dynamically**

**IDE-15: Use an echo statement to print something to test output**



Selenium behaves properly when using the store and echo command together. The echo command simply prints out the information in a variable to the log area of the selenium IDE. There is little room for error as the invocations of store and echo are specific. The syntax of the store function is simple and intuitive but the syntax for echo requires the use of curly brackets that encapsulate the variable name and a ‘$’ which is simple but not very intuitive to new users because the echo reference table does not specify that the user needs to use this specific syntax.

**IDE-16: Assert some type of pop up window using assertAlert, assertConfirmation, or assertPrompt**



Selenium IDE encounters trouble when trying to select a pop up window to perform actions on. Selenium will allow you to assert and verify items in a pop up window but will not close the window. This problem seems to be rooted in the select window function which only seems to select other tabs or instances of firefox that the user has initiated themselves. Selenium defaults the first non-top window when dealing with pop ups that have no windowID. the Selenium IDE notes “(don't rely on this if you are working with multiple popups simultaneously)”.

**Selenium Webdriver Test Cases**

**Test Case 1: Test for Ajax.**

public static void testAjax(){

System.setProperty("webdriver.chrome.driver", "C:/Users/David/Downloads/Software/apache-maven-3.3.1/webdrivers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("http://www.degraeve.com/reference/simple-ajax-example.php");

driver.findElement(By.name("word")).sendKeys("Lamar is the man!");;

driver.findElement(By.cssSelector("input[type='button']")).click();

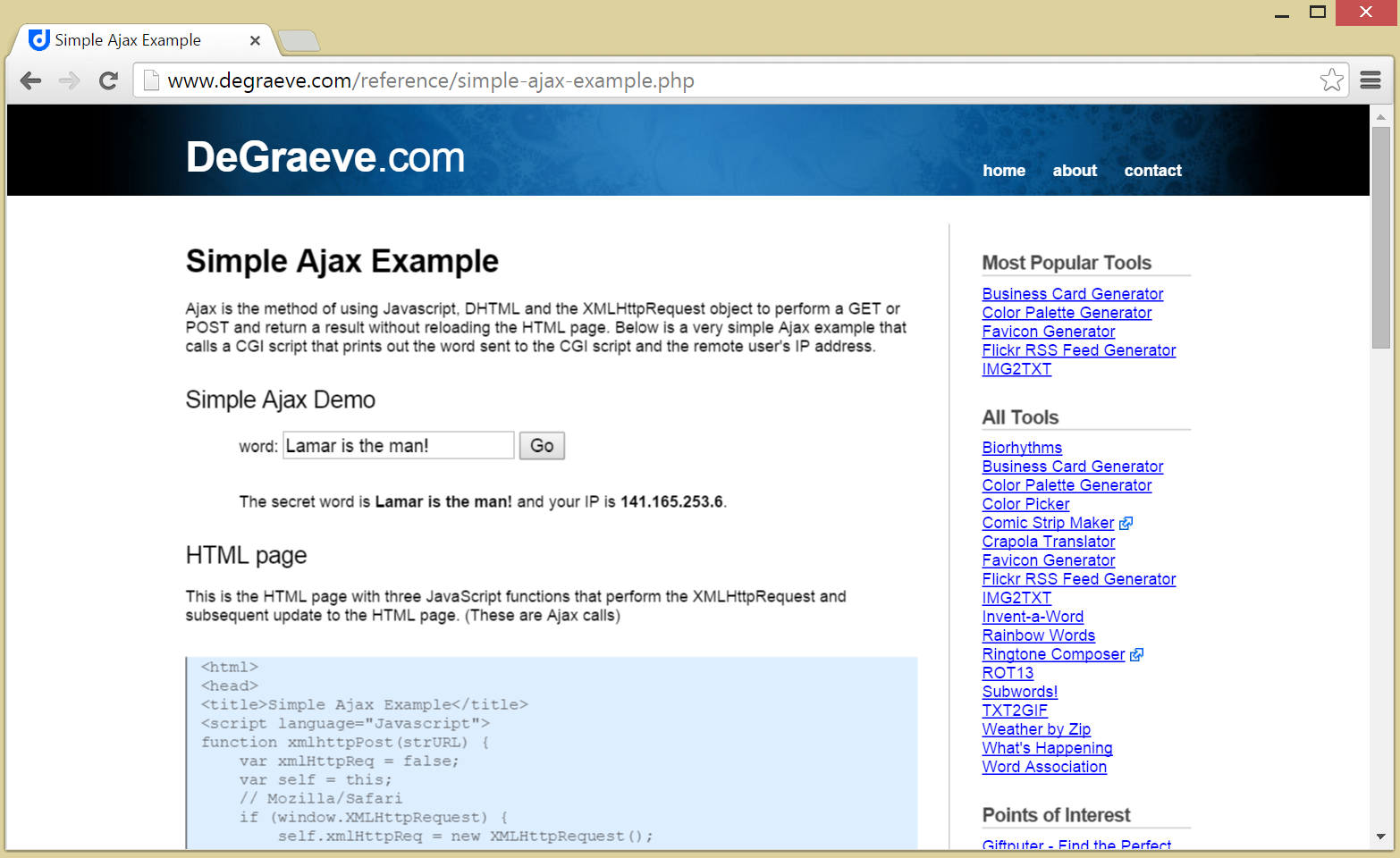
WebDriverWait driverWait = new WebDriverWait(driver, 10);

WebElement dynamicElement = driverWait.until(ExpectedConditions.presenceOfElementLocated(By.cssSelector("#result p")));

System.out.println(dynamicElement.getText());

}

**Screenshot Test for Ajax:**



**Test Case 2: Test Pop Up/Alerts**

Notes:

Selenium has a difficult time dealing with pop up windows in javascript. We are unable to find a workaround for this.

public static void testPopUp(){

System.setProperty("webdriver.chrome.driver", "C:/Users/David/Downloads/Software/apache-maven-3.3.1/webdrivers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Go to the Google Suggest home page

driver.get("http://www.mediacollege.com/internet/javascript/basic/alert.html");

WebElement alertPop = driver.findElement(By.cssSelector("input[value='Click Here']"));

alertPop.click();

alertPop.click();

try {

Thread.sleep(5000);

} catch (InterruptedException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

Alert alert = driver.switchTo().alert();

alert.accept();

try {

Thread.sleep(10000);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

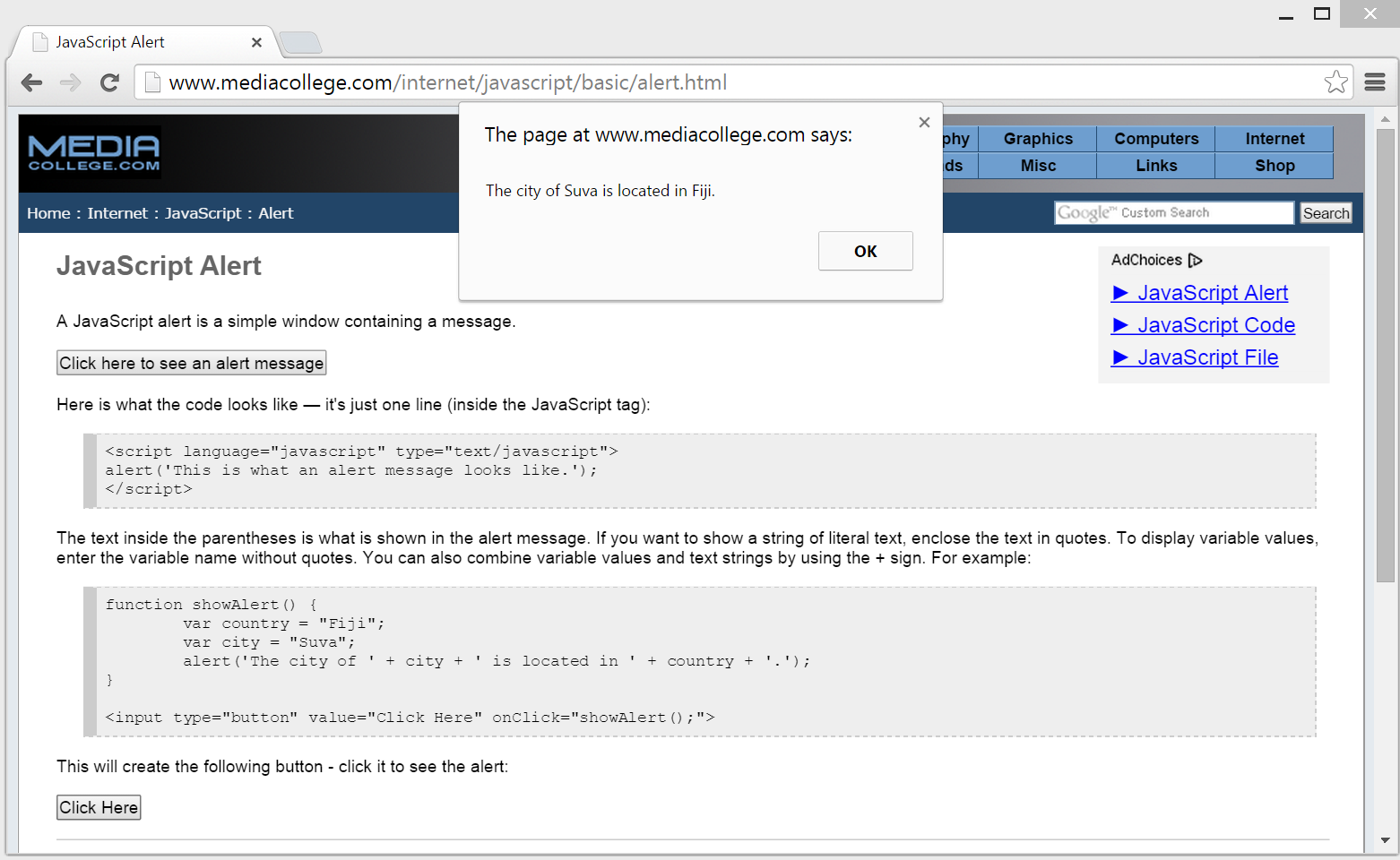
e.printStackTrace();

}

driver.quit();

}

Screenshot:



**Test Case 3: Test Multiple Browsers**

import static org.junit.Assert.\*;

import java.io.File;

import org.junit.Test;

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.ie.InternetExplorerDriver;

public class multInstanceBrowserTest {

@Test

public void test() {

fail("Not yet implemented");

}

@Test

public void testIE(){

File file = new File("C:/Users/David/Downloads/Software/apache-maven-3.3.1/webdrivers/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());

assertEquals(driver.getTitle(), "Cheese! - Google Search");

//Close the browser

driver.quit();

}

@Test

public void testChrome(){

System.setProperty("webdriver.chrome.driver", "C:/Users/David/Downloads/Software/apache-maven-3.3.1/webdrivers/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());

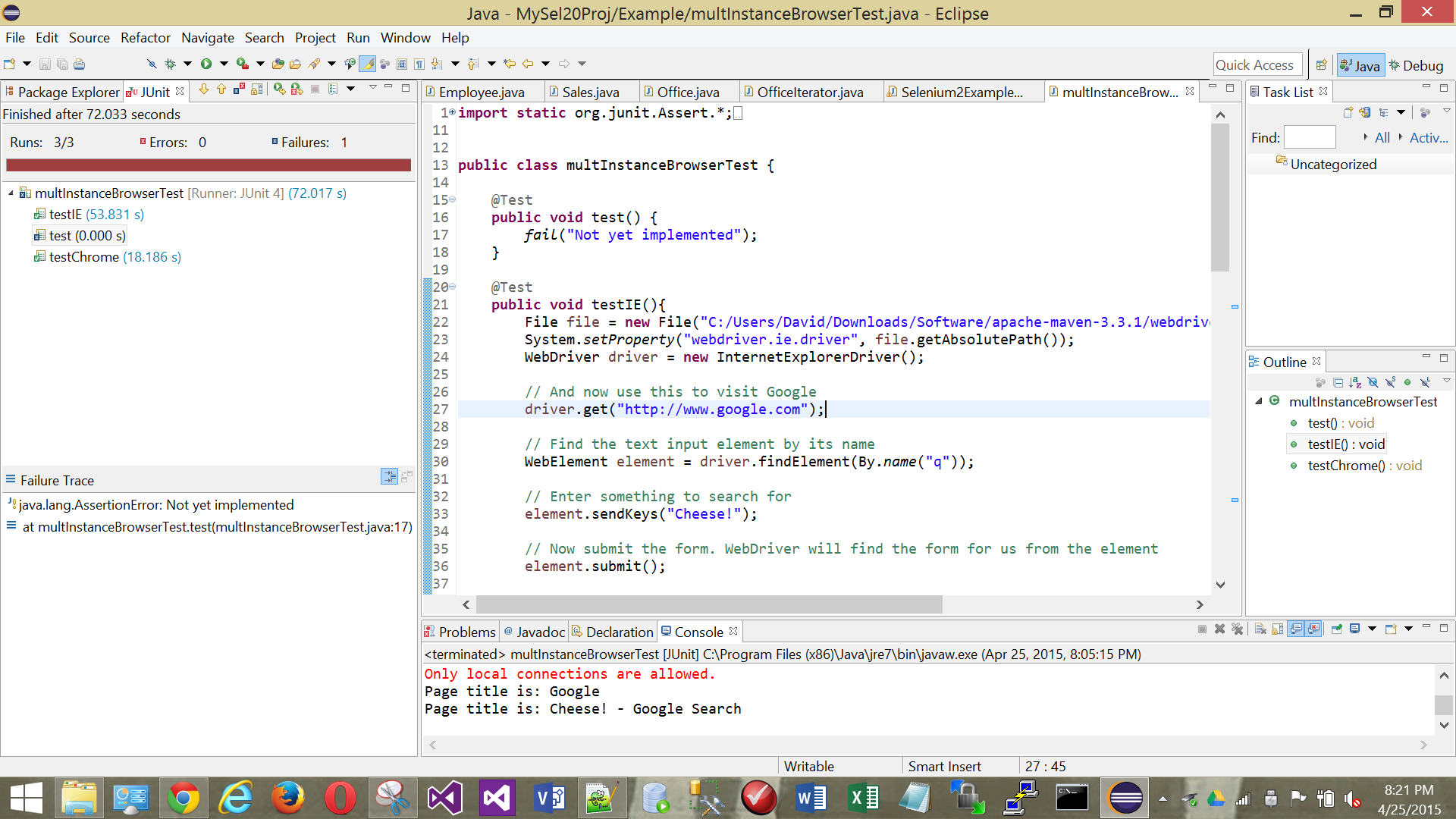
assertEquals(driver.getTitle(), "Cheese! - Google Search");

//Close the browser

driver.quit();

}

}



**Test Case 4: Test Looping through Select tags**

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");

}

});

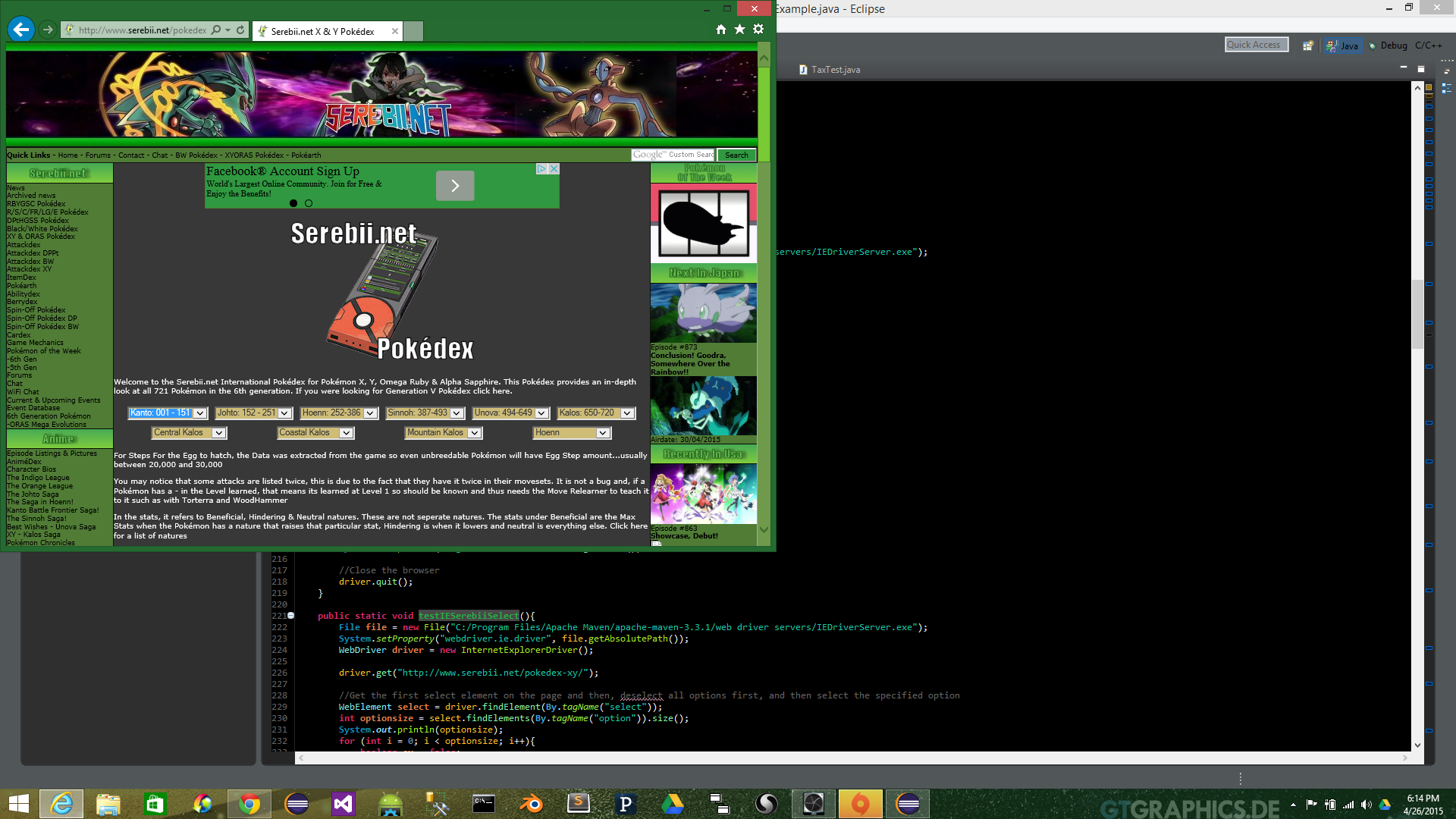
System.out.println("Page title is: " + driver.getTitle());

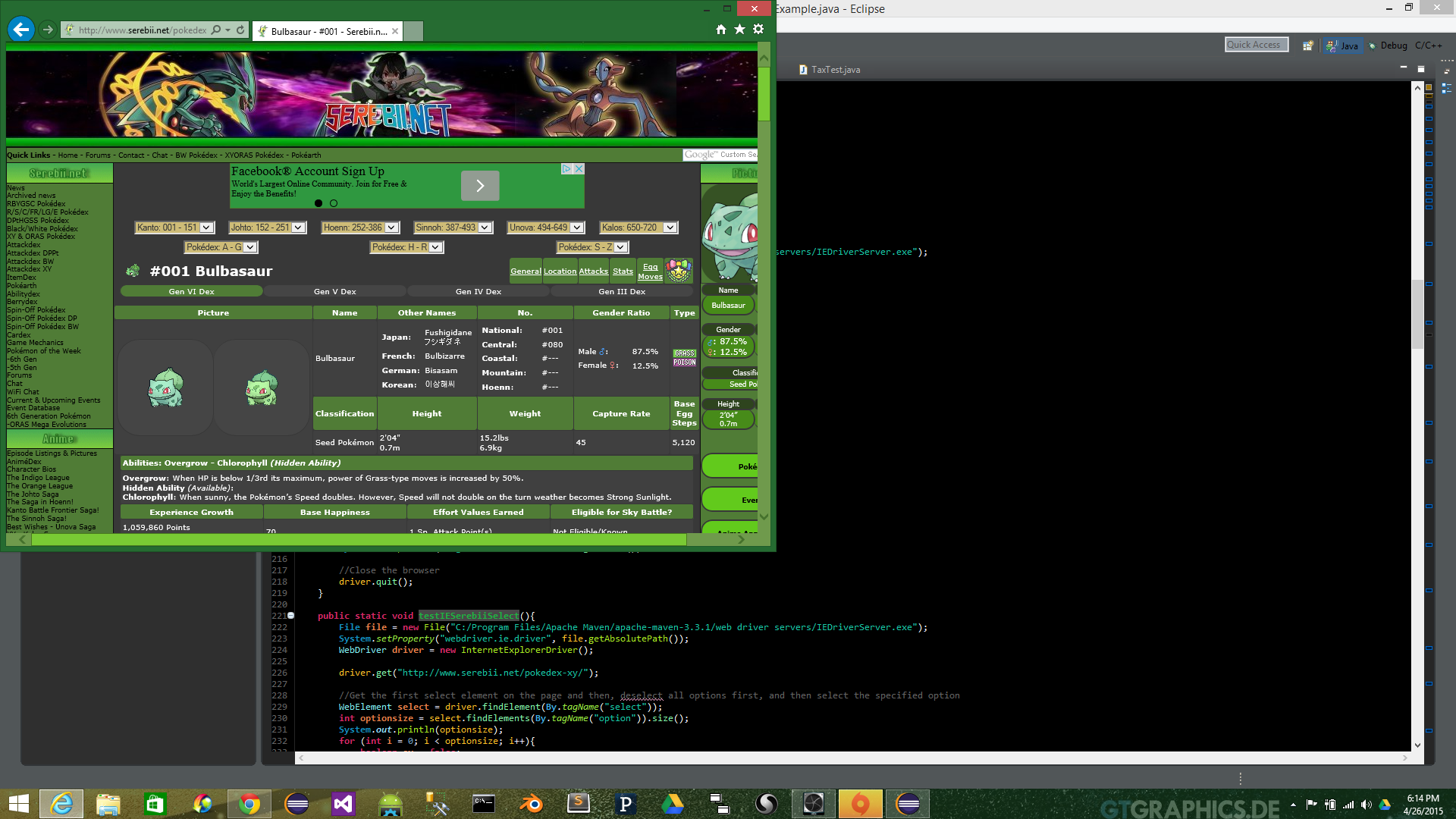
//Close the browser

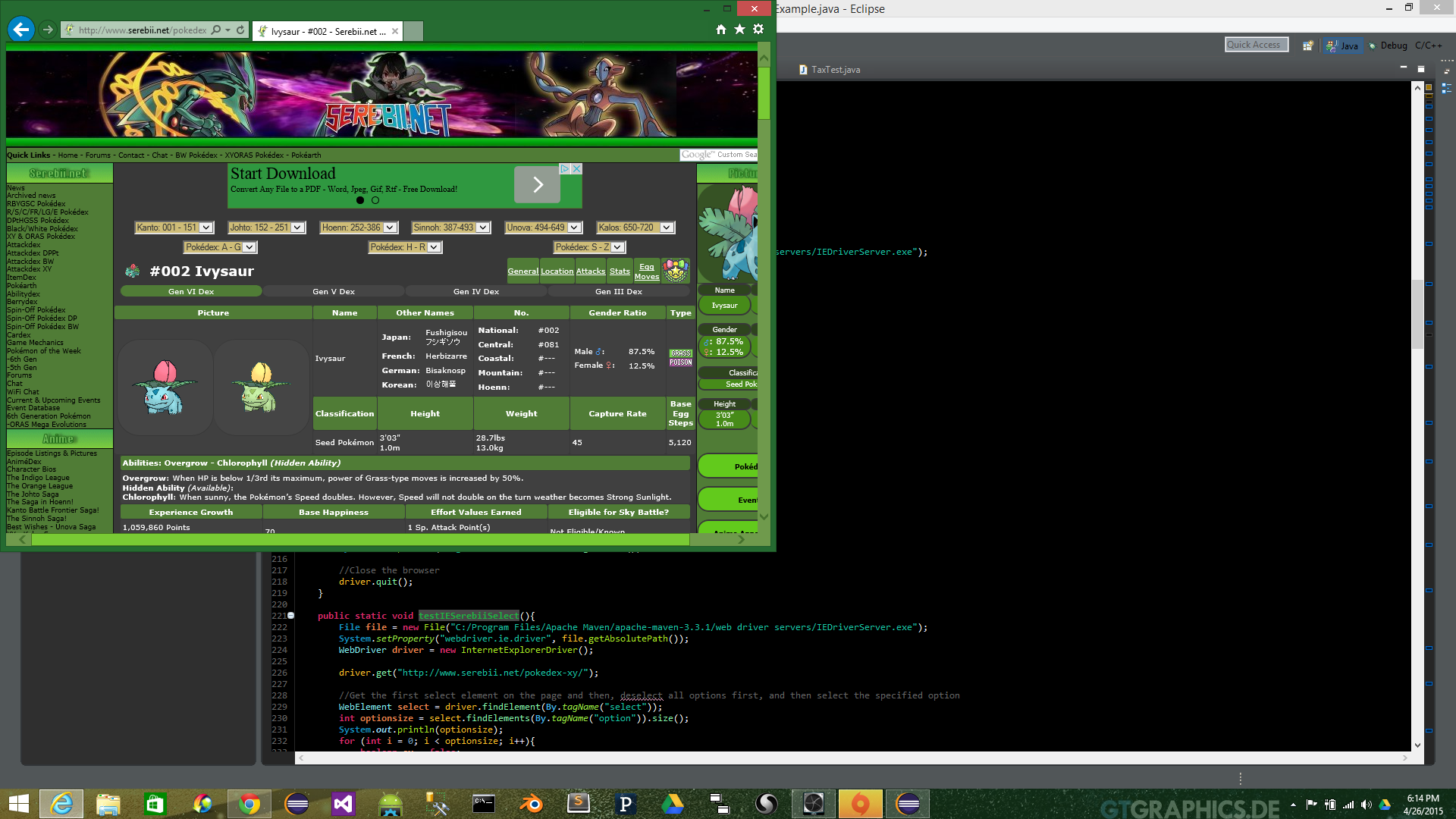
driver.quit();

}

**Screenshots:**







**Test Case 5: MiscTest**

Notes:

Unit Test that comes with the Selenium Source Code

Code:

// Licensed to the Software Freedom Conservancy (SFC) under one

// or more contributor license agreements. See the NOTICE file

// distributed with this work for additional information

// regarding copyright ownership. The SFC licenses this file

// to you under the Apache License, Version 2.0 (the

// "License"); you may not use this file except in compliance

// with the License. You may obtain a copy of the License at

//

// http://www.apache.org/licenses/LICENSE-2.0

//

// Unless required by applicable law or agreed to in writing,

// software distributed under the License is distributed on an

// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

// KIND, either express or implied. See the License for the

// specific language governing permissions and limitations

// under the License.

package org.openqa.selenium;

import org.junit.Test;

import org.openqa.selenium.testing.Ignore;

import org.openqa.selenium.testing.JUnit4TestBase;

import org.openqa.selenium.testing.JavascriptEnabled;

import static org.hamcrest.Matchers.equalTo;

import static org.hamcrest.Matchers.is;

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertThat;

import static org.junit.Assert.assertTrue;

import static org.openqa.selenium.testing.Ignore.Driver.CHROME;

import static org.openqa.selenium.testing.Ignore.Driver.IE;

import static org.openqa.selenium.testing.Ignore.Driver.MARIONETTE;

import static org.openqa.selenium.testing.Ignore.Driver.SAFARI;

public class MiscTest extends JUnit4TestBase {

@Test

public void testShouldReturnTitleOfPageIfSet() {

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

assertThat(driver.getTitle(), equalTo(("XHTML Test Page")));

driver.get(pages.simpleTestPage);

assertThat(driver.getTitle(), equalTo("Hello WebDriver"));

}

@Test

public void testShouldReportTheCurrentUrlCorrectly() {

driver.get(pages.simpleTestPage);

assertTrue(pages.simpleTestPage.equalsIgnoreCase(driver.getCurrentUrl()));

driver.get(pages.javascriptPage);

assertTrue(pages.javascriptPage.equalsIgnoreCase(driver.getCurrentUrl()));

}

@Test

public void shouldReturnTagName() {

driver.get(pages.formPage);

WebElement selectBox = driver.findElement(By.id("cheese"));

assertThat(selectBox.getTagName().toLowerCase(), is("input"));

}

@JavascriptEnabled

@Test

public void testShouldReturnTheSourceOfAPage() {

driver.get(pages.simpleTestPage);

String source = driver.getPageSource().toLowerCase();

assertThat(source.contains("<html"), is(true));

assertThat(source.contains("</html"), is(true));

assertThat(source.contains("an inline element"), is(true));

assertThat(source.contains("<p id="), is(true));

assertThat(source.contains("lotsofspaces"), is(true));

assertThat(source.contains("with document.write and with document.write again"), is(true));

}

@JavascriptEnabled

@Ignore(value = {CHROME, IE, SAFARI},

reason = "Chrome, Safari: returns XML content formatted for display as HTML document" +

"Others: untested")

@Test

public void testShouldBeAbleToGetTheSourceOfAnXmlDocument() {

driver.get(pages.simpleXmlDocument);

String source = driver.getPageSource().toLowerCase();

assertThat(source.replaceAll("\\s", ""), equalTo("<xml><foo><bar>baz</bar></foo></xml>"));

}

@Ignore(issues = {2282})

@Test

public void testStimulatesStrangeOnloadInteractionInFirefox()

throws Exception {

driver.get(pages.documentWrite);

// If this command succeeds, then all is well.

driver.findElement(By.xpath("//body"));

driver.get(pages.simpleTestPage);

driver.findElement(By.id("links"));

}

@JavascriptEnabled

@Test

@Ignore(MARIONETTE)

public void testClickingShouldNotTrampleWOrHInGlobalScope() throws Throwable {

driver.get(appServer.whereIs("globalscope.html"));

String[] vars = new String[]{"w", "h"};

for (String var : vars) {

assertEquals(var, getGlobalVar(driver, var));

}

driver.findElement(By.id("toclick")).click();

for (String var : vars) {

assertEquals(var, getGlobalVar(driver, var));

}

}

private String getGlobalVar(WebDriver driver, String var) {

Object val = ((JavascriptExecutor) driver).executeScript("return window." + var + ";");

return val == null ? "null" : val.toString();

}

}

Results:

Apr 25, 2015 4:17:09 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 4:17:09 PM org.openqa.selenium.testing.JUnit4TestBase$1 starting

INFO: >>> Starting testShouldReturnTheSourceOfAPage(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.Server doStart

INFO: jetty-7.x.y-SNAPSHOT

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/common,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/common/src/web/}

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/javascript,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/javascript/}

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/third\_party/closure/goog,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/third\_party/closure/goog/}

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/third\_party/js,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/third\_party/js/}

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.AbstractConnector doStart

INFO: Started SelectChannelConnector@0.0.0.0:27943

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.util.ssl.SslContextFactory doStart

INFO: Enabled Protocols [SSLv2Hello, TLSv1, TLSv1.1, TLSv1.2] of [SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2]

Apr 25, 2015 4:17:09 PM org.seleniumhq.jetty7.server.AbstractConnector doStart

INFO: Started SslSocketConnector@0.0.0.0:42930

Apr 25, 2015 4:17:09 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 4:17:09 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Building [//javascript/firefox-driver:webdriver] ...

>>> (in C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540)

>>> Preparing: //javascript/firefox-driver:webdriver as build/javascript/firefox-driver/webdriver.xpi

>>> Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=1024m; support was removed in 8.0

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: get [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/simpleTest.html}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/simpleTest.html}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: getPageSource [a659d8f8-5f7d-4498-b571-7b76d4b10113, getPageSource {}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, getPageSource {}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.testing.JUnit4TestBase$1 finished

INFO: <<< Finished testShouldReturnTheSourceOfAPage(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:21 PM org.openqa.selenium.testing.JUnit4TestBase$1 starting

INFO: >>> Starting testClickingShouldNotTrampleWOrHInGlobalScope(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: get [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/globalscope.html}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/globalscope.html}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: executeScript [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.w;, args=[]}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.w;, args=[]}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: executeScript [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.h;, args=[]}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.h;, args=[]}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: findElement [a659d8f8-5f7d-4498-b571-7b76d4b10113, findElement {using=id, value=toclick}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, findElement {using=id, value=toclick}]

Apr 25, 2015 4:17:21 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: clickElement [a659d8f8-5f7d-4498-b571-7b76d4b10113, clickElement {id={3fa320d7-af09-445c-a42d-221e4ee84493}}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, clickElement {id={3fa320d7-af09-445c-a42d-221e4ee84493}}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: executeScript [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.w;, args=[]}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.w;, args=[]}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: executeScript [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.h;, args=[]}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, executeScript {script=return window.h;, args=[]}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 finished

INFO: <<< Finished testClickingShouldNotTrampleWOrHInGlobalScope(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 starting

INFO: >>> Starting testShouldBeAbleToGetTheSourceOfAnXmlDocument(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: get [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/simple.xml}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/simple.xml}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: getPageSource [a659d8f8-5f7d-4498-b571-7b76d4b10113, getPageSource {}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, getPageSource {}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 finished

INFO: <<< Finished testShouldBeAbleToGetTheSourceOfAnXmlDocument(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 starting

INFO: >>> Starting shouldReturnTagName(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: get [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/formPage.html}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/formPage.html}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: findElement [a659d8f8-5f7d-4498-b571-7b76d4b10113, findElement {using=id, value=cheese}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, findElement {using=id, value=cheese}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: getElementTagName [a659d8f8-5f7d-4498-b571-7b76d4b10113, getElementTagName {id={2aefa95a-efd4-4fc8-b958-00836922623c}}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, getElementTagName {id={2aefa95a-efd4-4fc8-b958-00836922623c}}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 finished

INFO: <<< Finished shouldReturnTagName(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 starting

INFO: >>> Starting testShouldReportTheCurrentUrlCorrectly(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: get [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/simpleTest.html}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/simpleTest.html}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: getCurrentUrl [a659d8f8-5f7d-4498-b571-7b76d4b10113, getCurrentUrl {}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, getCurrentUrl {}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: get [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/javascriptPage.html}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://localhost:27943/common/javascriptPage.html}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: getCurrentUrl [a659d8f8-5f7d-4498-b571-7b76d4b10113, getCurrentUrl {}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, getCurrentUrl {}]

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 finished

INFO: <<< Finished testShouldReportTheCurrentUrlCorrectly(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.testing.JUnit4TestBase$1 starting

INFO: >>> Starting testShouldReturnTitleOfPageIfSet(org.openqa.selenium.MiscTest)

Apr 25, 2015 4:17:22 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executing: get [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://www.chadgolden.com/se/test/pictures}]

Apr 25, 2015 4:17:27 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, get {url=http://www.chadgolden.com/se/test/pictures}]

Apr 25, 2015 4:17:27 PM org.openqa.selenium.remote.RemoteWebDriver log

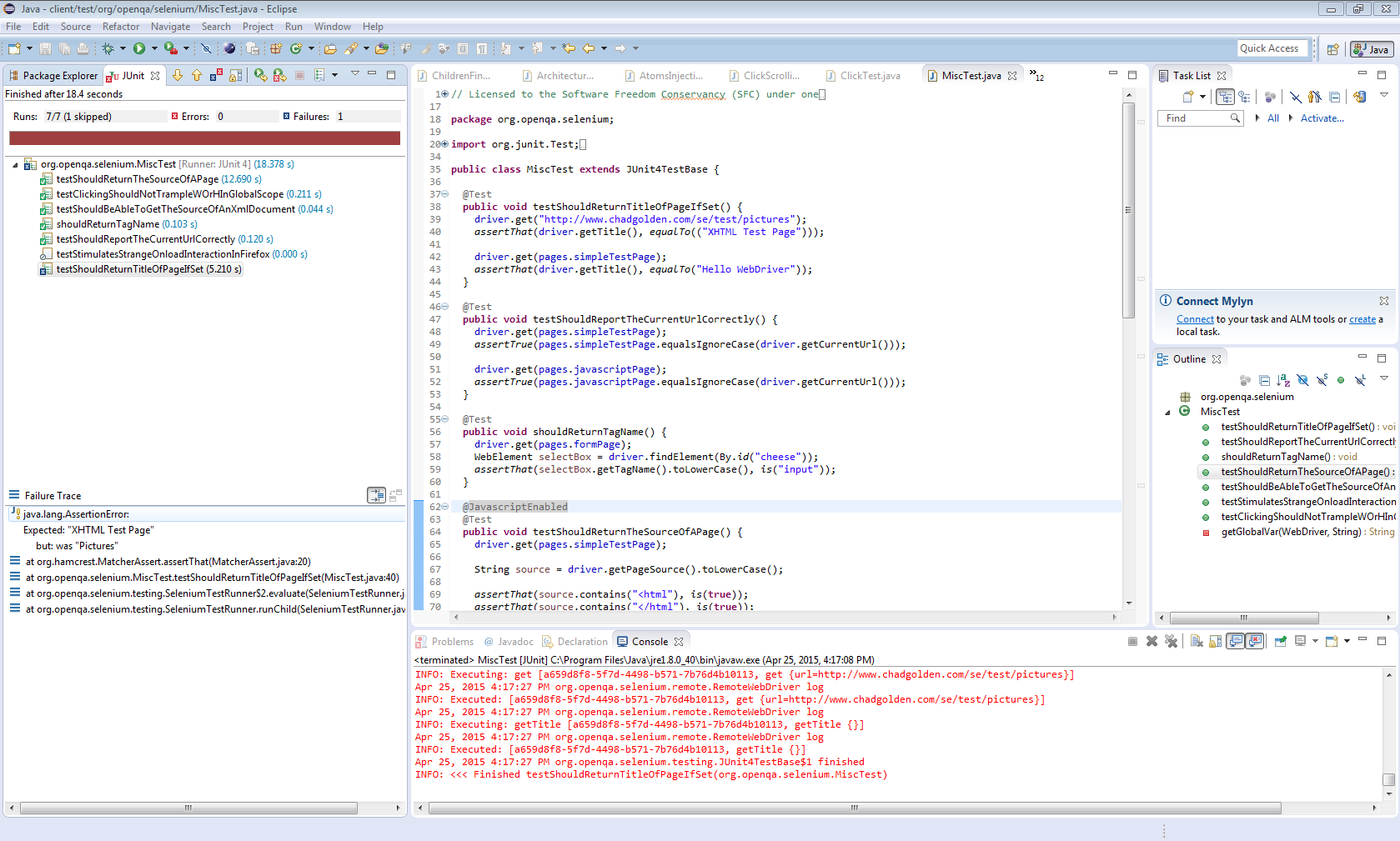
INFO: Executing: getTitle [a659d8f8-5f7d-4498-b571-7b76d4b10113, getTitle {}]

Apr 25, 2015 4:17:27 PM org.openqa.selenium.remote.RemoteWebDriver log

INFO: Executed: [a659d8f8-5f7d-4498-b571-7b76d4b10113, getTitle {}]

Apr 25, 2015 4:17:27 PM org.openqa.selenium.testing.JUnit4TestBase$1 finished

INFO: <<< Finished testShouldReturnTitleOfPageIfSet(org.openqa.selenium.MiscTest)



**Test Case 6: Webdriver Selenium Test Suite**

Notes:

Unit Test from Selenium source code

CODE:

// Licensed to the Software Freedom Conservancy (SFC) under one

// or more contributor license agreements. See the NOTICE file

// distributed with this work for additional information

// regarding copyright ownership. The SFC licenses this file

// to you under the Apache License, Version 2.0 (the

// "License"); you may not use this file except in compliance

// with the License. You may obtain a copy of the License at

//

// http://www.apache.org/licenses/LICENSE-2.0

//

// Unless required by applicable law or agreed to in writing,

// software distributed under the License is distributed on an

// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

// KIND, either express or implied. See the License for the

// specific language governing permissions and limitations

// under the License.

package com.thoughtworks.selenium;

import org.junit.runner.RunWith;

import org.junit.runners.Suite;

import com.thoughtworks.selenium.corebased.SeleniumMouseTest;

import com.thoughtworks.selenium.corebased.TestAddLocationStrategy;

import com.thoughtworks.selenium.corebased.TestAddSelection;

import com.thoughtworks.selenium.corebased.TestAlerts;

import com.thoughtworks.selenium.corebased.TestBasicAuth;

import com.thoughtworks.selenium.corebased.TestBrowserVersion;

import com.thoughtworks.selenium.corebased.TestCheckUncheck;

import com.thoughtworks.selenium.corebased.TestClick;

import com.thoughtworks.selenium.corebased.TestClickAt;

import com.thoughtworks.selenium.corebased.TestClickJavascriptHref;

import com.thoughtworks.selenium.corebased.TestClickJavascriptHrefChrome;

import com.thoughtworks.selenium.corebased.TestCommandError;

import com.thoughtworks.selenium.corebased.TestComments;

import com.thoughtworks.selenium.corebased.TestConfirmations;

import com.thoughtworks.selenium.corebased.TestCssLocators;

import com.thoughtworks.selenium.corebased.TestDojoDragAndDrop;

import com.thoughtworks.selenium.corebased.TestEditable;

import com.thoughtworks.selenium.corebased.TestElementIndex;

import com.thoughtworks.selenium.corebased.TestElementOrder;

import com.thoughtworks.selenium.corebased.TestElementPresent;

import com.thoughtworks.selenium.corebased.TestErrorChecking;

import com.thoughtworks.selenium.corebased.TestEval;

import com.thoughtworks.selenium.corebased.TestEvilClosingWindow;

import com.thoughtworks.selenium.corebased.TestFailingAssert;

import com.thoughtworks.selenium.corebased.TestFailingVerifications;

import com.thoughtworks.selenium.corebased.TestFocusOnBlur;

import com.thoughtworks.selenium.corebased.TestFramesClick;

import com.thoughtworks.selenium.corebased.TestFramesClickJavascriptHref;

import com.thoughtworks.selenium.corebased.TestFramesOpen;

import com.thoughtworks.selenium.corebased.TestFramesSpecialTargets;

import com.thoughtworks.selenium.corebased.TestFunkEventHandling;

import com.thoughtworks.selenium.corebased.TestGet;

import com.thoughtworks.selenium.corebased.TestGetTextContent;

import com.thoughtworks.selenium.corebased.TestGettingValueOfCheckbox;

import com.thoughtworks.selenium.corebased.TestGettingValueOfRadioButton;

import com.thoughtworks.selenium.corebased.TestGoBack;

import com.thoughtworks.selenium.corebased.TestHighlight;

import com.thoughtworks.selenium.corebased.TestHtmlSource;

import com.thoughtworks.selenium.corebased.TestImplicitLocators;

import com.thoughtworks.selenium.corebased.TestJavaScriptAttributes;

import com.thoughtworks.selenium.corebased.TestLocators;

import com.thoughtworks.selenium.corebased.TestMultiSelect;

import com.thoughtworks.selenium.corebased.TestOpen;

import com.thoughtworks.selenium.corebased.TestOpenInTargetFrame;

import com.thoughtworks.selenium.corebased.TestPatternMatching;

import com.thoughtworks.selenium.corebased.TestPause;

import com.thoughtworks.selenium.corebased.TestProxy;

import com.thoughtworks.selenium.corebased.TestQuickOpen;

import com.thoughtworks.selenium.corebased.TestSelect;

import com.thoughtworks.selenium.corebased.TestSelectMultiLevelFrame;

import com.thoughtworks.selenium.corebased.TestSelectPopUp;

import com.thoughtworks.selenium.corebased.TestSelectWindow;

import com.thoughtworks.selenium.corebased.TestSelectWindowTitle;

import com.thoughtworks.selenium.corebased.TestStore;

import com.thoughtworks.selenium.corebased.TestSubmit;

import com.thoughtworks.selenium.corebased.TestTextWhitespace;

import com.thoughtworks.selenium.corebased.TestType;

import com.thoughtworks.selenium.corebased.TestTypeRichText;

import com.thoughtworks.selenium.corebased.TestVerifications;

import com.thoughtworks.selenium.corebased.TestVisibility;

import com.thoughtworks.selenium.corebased.TestWait;

import com.thoughtworks.selenium.corebased.TestWaitFor;

import com.thoughtworks.selenium.corebased.TestWaitForNot;

import com.thoughtworks.selenium.corebased.TestWaitInPopupWindow;

import com.thoughtworks.selenium.corebased.TestXPathLocators;

@RunWith(Suite.class)

@Suite.SuiteClasses({

SeleniumMouseTest.class,

TestAddLocationStrategy.class,

TestAddSelection.class,

TestAlerts.class,

TestBasicAuth.class,

TestBrowserVersion.class,

TestCheckUncheck.class,

TestClick.class,

TestClickAt.class,

// TestClickBlankTarget.class,

TestClickJavascriptHref.class,

TestClickJavascriptHrefChrome.class, // alerts

// TestClickJavascriptHrefWithVoidChrome.class, // fails in IE

TestCommandError.class,

TestComments.class,

TestConfirmations.class, // fails in IE

// TestCookie.class,

TestCssLocators.class,

// TestCursorPosition.class,

TestDojoDragAndDrop.class, // fails in IE

// TestDomainCookie.class,

// TestDragAndDrop.class,

TestEditable.class,

TestElementIndex.class,

TestElementOrder.class,

TestElementPresent.class,

TestErrorChecking.class,

TestEval.class,

TestEvilClosingWindow.class,

TestFailingAssert.class,

TestFailingVerifications.class,

TestFocusOnBlur.class,

TestFramesClick.class,

TestFramesClickJavascriptHref.class,

// TestFramesNested.class,

TestFramesOpen.class,

TestFramesSpecialTargets.class,

TestFunkEventHandling.class,

TestGet.class,

TestGetTextContent.class,

TestGettingValueOfCheckbox.class,

TestGettingValueOfRadioButton.class,

TestGoBack.class,

TestHighlight.class,

TestHtmlSource.class,

TestImplicitLocators.class,

TestJavaScriptAttributes.class,

// TestJavascriptParameters.class,

TestLocators.class,

TestMultiSelect.class,

// TestModalDialog.class,

TestOpen.class,

TestOpenInTargetFrame.class,

TestPatternMatching.class,

TestPause.class,

// TestPrompt.class,

TestProxy.class,

TestQuickOpen.class,

// TestRefresh.class,

// TestRollup.class,

TestSelect.class,

TestSelectMultiLevelFrame.class,

TestSelectPopUp.class,

TestSelectWindow.class,

TestSelectWindowTitle.class,

// TestSetSpeed.class,

TestStore.class,

TestSubmit.class,

TestTextWhitespace.class,

TestType.class,

TestTypeRichText.class,

// TestUIElementLocators.class,

// TestUseXpathLibrary.class,

TestVerifications.class,

TestVisibility.class,

TestWait.class,

TestWaitFor.class,

TestWaitForNot.class,

TestWaitInPopupWindow.class,

// TestXPathLocatorInXHtml.class,

TestXPathLocators.class

})

public class WebDriverSeleniumTestSuite extends BaseSuite {

// Empty

}

RESULTS:

Apr 25, 2015 3:42:50 PM com.thoughtworks.selenium.BaseSuite$1 before

INFO: Preparing test environment

Building [//java/server/test/org/openqa/selenium:server-with-tests:uber] ...

>>> (in C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540)

>>> Preparing: //java/client/src/org/openqa/selenium/firefox:amd64 as build\java\client\src\org\openqa\selenium\firefox\amd64

>>> Preparing: //java/client/src/org/openqa/selenium/firefox:x86 as build\java\client\src\org\openqa\selenium\firefox\x86

>>> Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=1024m; support was removed in 8.0

15:42:59.818 INFO - Launching a standalone server

15:42:59.904 INFO - Java: Oracle Corporation 25.40-b25

15:42:59.904 INFO - OS: Windows 7 6.1 amd64

15:42:59.912 INFO - v2.45.0, with Core v2.45.0. Built from revision

15:42:59.955 INFO - Unable to register driver with className com.opera.core.systems.OperaDriver due to ClassNotFoundException

15:42:59.995 INFO - RemoteWebDriver instances should connect to: http://127.0.0.1:24706/wd/hub

15:42:59.995 INFO - Version Jetty/5.1.x

15:42:59.996 INFO - Started HttpContext[/selenium-server,/selenium-server]

15:43:00.067 INFO - Started org.openqa.jetty.jetty.servlet.ServletHandler@7907ec20

15:43:00.068 INFO - Started HttpContext[/wd,/wd]

15:43:00.068 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]

15:43:00.068 INFO - Started HttpContext[/,/]

15:43:00.071 INFO - Started SocketListener on 0.0.0.0:24706

15:43:00.071 INFO - Started org.openqa.jetty.jetty.Server@133314b

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.Server doStart

INFO: jetty-7.x.y-SNAPSHOT

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/common,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/common/src/web/}

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/javascript,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/javascript/}

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/third\_party/closure/goog,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/third\_party/closure/goog/}

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/third\_party/js,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/third\_party/js/}

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler startContext

INFO: started o.s.j.s.ServletContextHandler{/selenium-server,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/java/server/test/org/openqa/selenium/}

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.handler.ContextHandler$Context log

INFO: Aliases are enabled

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.AbstractConnector doStart

INFO: Started SelectChannelConnector@0.0.0.0:8982

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.util.ssl.SslContextFactory doStart

INFO: Enabled Protocols [SSLv2Hello, TLSv1, TLSv1.1, TLSv1.2] of [SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2]

Apr 25, 2015 3:43:00 PM org.seleniumhq.jetty7.server.AbstractConnector doStart

INFO: Started SslSocketConnector@0.0.0.0:13449

Apr 25, 2015 3:43:00 PM com.thoughtworks.selenium.InternalSelenseTestBase buildJavascriptLibraries

INFO: In dev mode. Copying required files in case we're using a WebDriver-backed Selenium

Building [//java/client/src/org/openqa/selenium/internal/seleniumemulation, //third\_party/js/sizzle] ...

>>> (in C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540)

>>> go aborted!

>>> Don't know how to build task '//java/client/src/org/openqa/selenium/internal/seleniumemulation'

>>>

>>> (See full trace by running task with --trace)

>>> Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=1024m; support was removed in 8.0

Apr 25, 2015 3:43:08 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:08 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Building [//javascript/firefox-driver:webdriver] ...

>>> (in C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540)

>>> Preparing: //javascript/firefox-driver:webdriver as build/javascript/firefox-driver/webdriver.xpi

>>> Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=1024m; support was removed in 8.0

Apr 25, 2015 3:43:20 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:20 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:24 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:24 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:28 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:28 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:33 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:33 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:37 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:37 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:41 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:41 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:46 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:46 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:51 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:51 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:55 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:55 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:43:59 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:43:59 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:03 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:03 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:07 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:07 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:12 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:12 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:16 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:16 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:21 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:21 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:26 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:26 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:30 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:30 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:34 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:34 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:39 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:39 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:43 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:43 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:47 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:47 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:52 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:52 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:44:57 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:44:57 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:01 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:01 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:06 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:06 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:10 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:10 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:15 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:15 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:19 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:19 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:24 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:24 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:29 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:29 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:33 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:33 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:38 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:38 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:43 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:43 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:48 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:48 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:53 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:53 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:45:58 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:45:58 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:02 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:02 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:09 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:09 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:13 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:13 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:17 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:17 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:22 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:22 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:27 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:27 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:31 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:31 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:35 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:35 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:40 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:40 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:45 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:45 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:49 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:49 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:54 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:54 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:46:58 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:46:58 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:03 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:03 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:07 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:07 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:12 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:12 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:16 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:16 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:21 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:21 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:25 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:25 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:30 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:30 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:35 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:35 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:39 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:39 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:43 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:43 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:47 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:47 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:51 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:51 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:47:55 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:47:55 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:48:01 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:48:01 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:48:05 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:48:05 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:48:10 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:48:10 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:48:14 PM org.openqa.selenium.testing.drivers.Browser detect

INFO: No browser detected, returning null

Apr 25, 2015 3:48:14 PM org.openqa.selenium.testing.drivers.DefaultDriverSupplier get

INFO: Providing default driver instance

Apr 25, 2015 3:48:18 PM com.thoughtworks.selenium.BaseSuite$2 after

INFO: Stopping browser

Apr 25, 2015 3:48:18 PM com.thoughtworks.selenium.BaseSuite$1 after

INFO: Cleaning test environment

Apr 25, 2015 3:48:18 PM org.seleniumhq.jetty7.server.handler.ContextHandler doStop

INFO: stopped o.s.j.s.ServletContextHandler{/selenium-server,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/java/server/test/org/openqa/selenium/}

Apr 25, 2015 3:48:18 PM org.seleniumhq.jetty7.server.handler.ContextHandler doStop

INFO: stopped o.s.j.s.ServletContextHandler{/third\_party/js,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/third\_party/js/}

Apr 25, 2015 3:48:18 PM org.seleniumhq.jetty7.server.handler.ContextHandler doStop

INFO: stopped o.s.j.s.ServletContextHandler{/third\_party/closure/goog,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/third\_party/closure/goog/}

Apr 25, 2015 3:48:18 PM org.seleniumhq.jetty7.server.handler.ContextHandler doStop

INFO: stopped o.s.j.s.ServletContextHandler{/javascript,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/javascript/}

Apr 25, 2015 3:48:18 PM org.seleniumhq.jetty7.server.handler.ContextHandler doStop

INFO: stopped o.s.j.s.ServletContextHandler{/common,file:/C:/Users/dc03693.AD/Downloads/selenium-4f86f013a6b189938d844d2a6dc5fb6a717ee540/common/src/web/}

Apr 25, 2015 3:48:18 PM org.openqa.selenium.os.UnixProcess$SeleniumWatchDog destroyHarder

INFO: Command failed to close cleanly. Destroying forcefully (v2). org.openqa.selenium.os.UnixProcess$SeleniumWatchDog@7cf6a5f9

**SCREENSHOT:**

