

Practical No 1

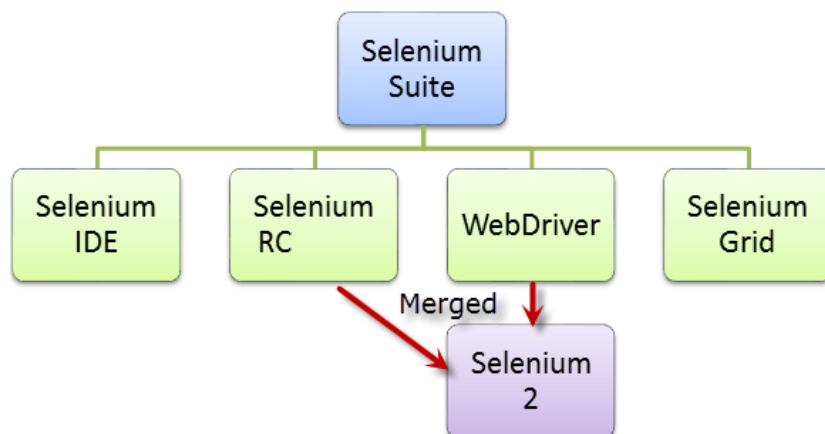
Title: Test Automation using Selenium IDE

Problem Statement:

Automate web test using Selenium IDE and Firebug.

Theory:

- **Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.**
- Selenium is not just a single tool but a suite of software, each catering to different testing needs of an organization. **It has four components.**
 - Selenium Integrated Development Environment (IDE)
 - Selenium Remote Control (RC)
 - WebDriver
 - Selenium Grid



- Selenium Integrated Development Environment (IDE) is the **simplest framework** in the Selenium suite and is **the easiest one to learn**. It is a **Firefox plugin** that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a **prototyping tool**.
- If you want to create more advanced test cases, you will need to use either Selenium RC or WebDriver.



1. Selenium IDE Demo Installation

Version to be used

(Selenium IDE Version 3.1.1 works with Firefox 56.0 and later)

You need

- Mozilla Firefox
- Active Internet Connection
- If you do not have Mozilla Firefox yet, you can download it **from** <http://www.mozilla.org/en-US/firefox/new>.

STEPS:

1. Launch **Mozilla Firefox** Browser.
2. Type URL <https://www.seleniumhq.org/download/>
OR
<https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>
3. Selenium IDE Add-ons page will get open then Click on Add to Firefox button
4. Firefox will show one popup saying do you want to allow Mozilla Firefox to install Selenium IDE Add-ons or not. **Click on Install** button.
5. Firefox will automatically install Selenium IDE software. After the installation is completed, a pop up window appears asking to re-start the Firefox. Click on the “Restart Now” button to reflect the Selenium IDE installation. Click on Restart Now button.
6. On clicking on the Restart Now button, Firefox will restart automatically. In case you missed the pop-up, simply close the Firefox and launch again.
7. Once the Firefox is booted and started again, we can see selenium IDE under the tools menu list. **Selenium IDE icon** will be displayed in the Firefox toolbar.
8. Click on **Selenium IDE icon to open Selenium IDE.**
 - Selenium IDE Works with all major versions, but we recommend to use 47.0.1 & above as they have better stability.
 - Selenium IDE is no longer compatible with Firefox 55 and above.

4 test cases for 4 websites

- i. <http://store.demoqa.com>
- ii. <https://www.seleniumhq.org>

- iii. <http://www.google.com>
- iv. <http://www.yahoo.com>
- v. <http://demo.guru99.com/test/newtours/>

Step 1

- Launch Firefox and Selenium IDE.
- Type the value for our Base URL: <http://demo.guru99.com/test/newtours/>.
- Toggle the Record button on (if it is not yet toggled on by default).

Step 2

- In Firefox, navigate to <http://demo.guru99.com/test/newtours/>. Firefox should take you to the page similar to the one shown below.

Selenium IDE:

Menu bar

Base URL bar

Toolbar

Test Case Pane

Log/Reference/UI-Element/Rollup Pane

Editor

Command	Target	Value
open	/	
type	userName	tutorial
type	password	cause_of_failure
clickAndWait	login	
verifyTitle	Find a Flight: Mercu...	
clickAndWait	link=SIGN-OFF	
verifyTitle	Sign-on: Mercury T...	
clickAndWait	link=Home	
verifyTitle	Welcome: Mercury ...	

Runs: 2
Failures: 1

Log

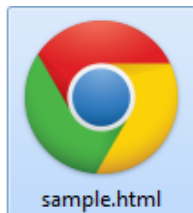
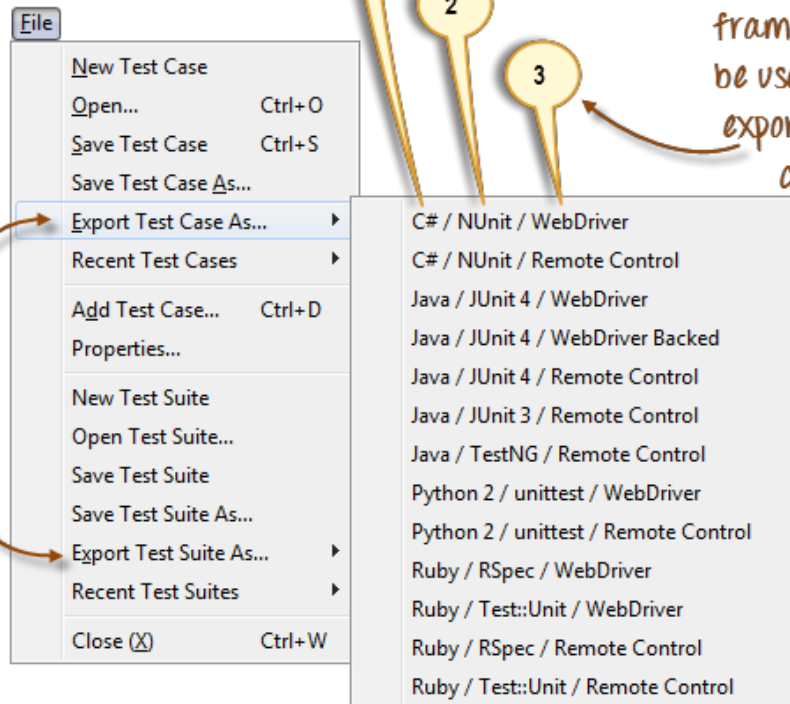
```
[info] Changed test case
[info] Executing: |open| / | |
[info] Executing: |type| userName | tutorial |
[info] Executing: |type| password | cause_of_failure |
[info] Executing: |clickAndWait| login | |
[info] Executing: |verifyTitle| Find a Flight: Mercury Tours: | |
[error] Actual value 'Sign-on: Mercury Tours' did not match 'Find a Flight: Mercury Tours:'
[info] Executing: |clickAndWait| link=SIGN-OFF | |
[error] Element link=SIGN-OFF not found
```

The file format to which your Selenium IDE test case will be exported

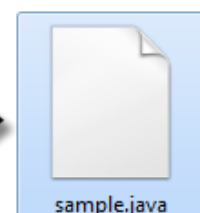
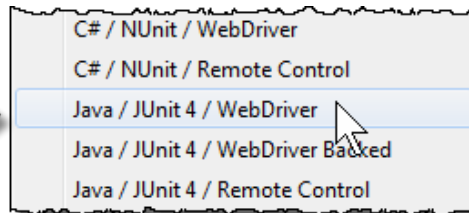
The unit testing framework to be used

The Selenium framework to be used on the exported test case.

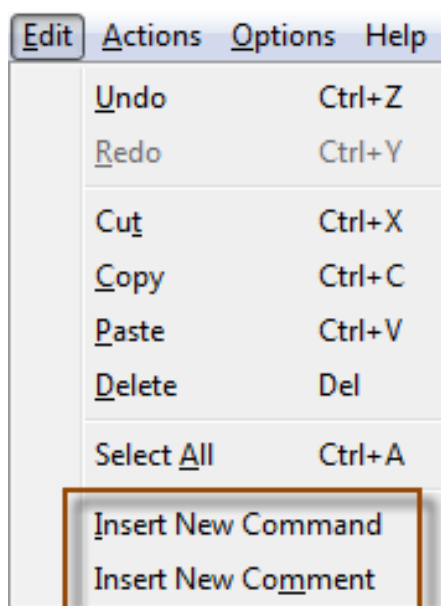
the 2 Export options provided by the File menu



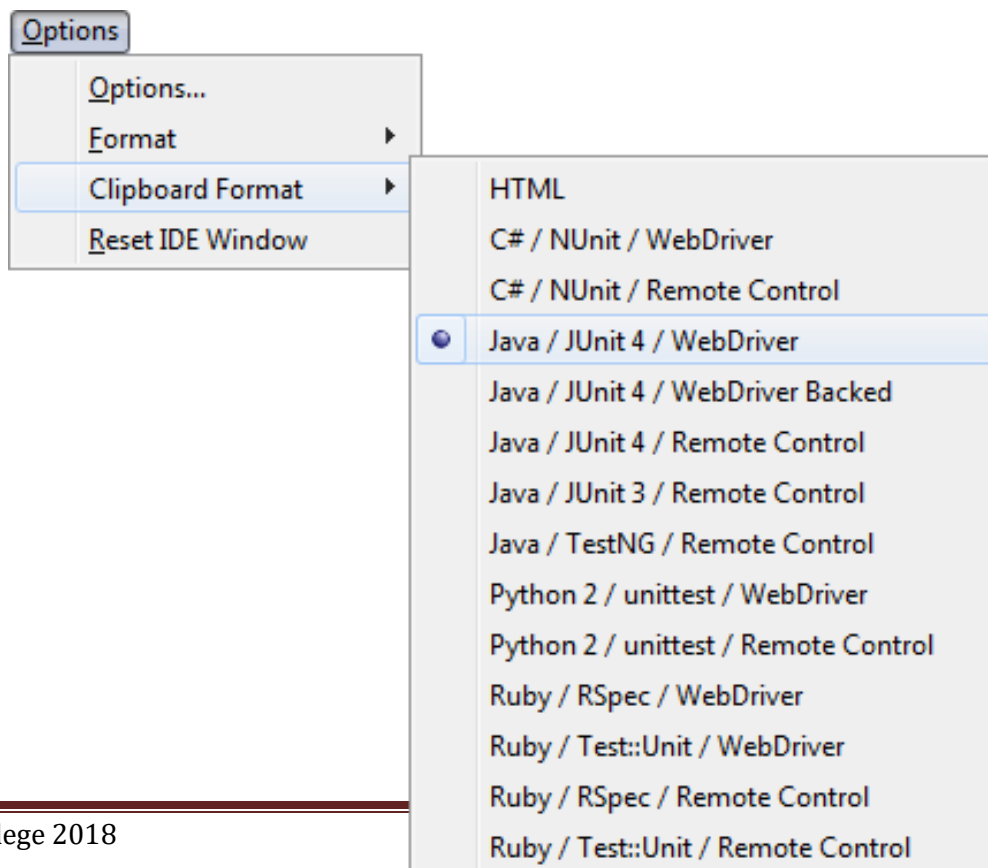
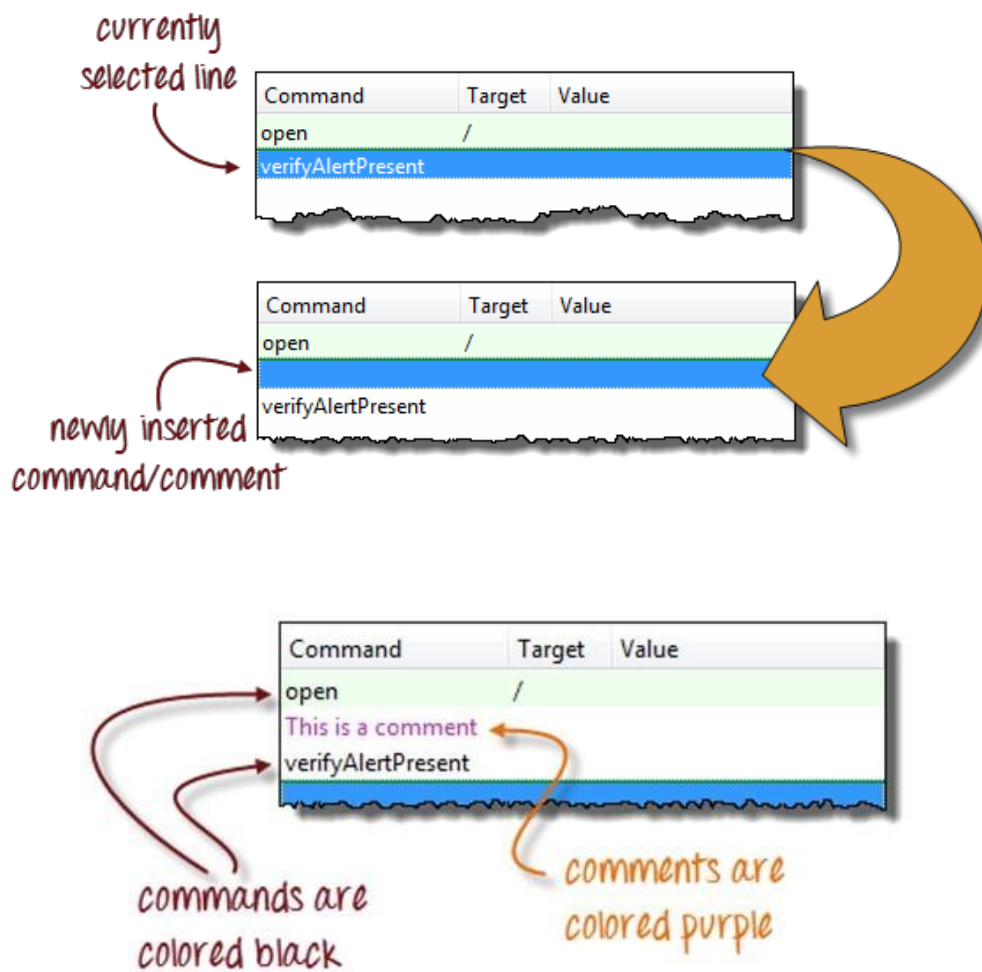
Selenium IDE test case

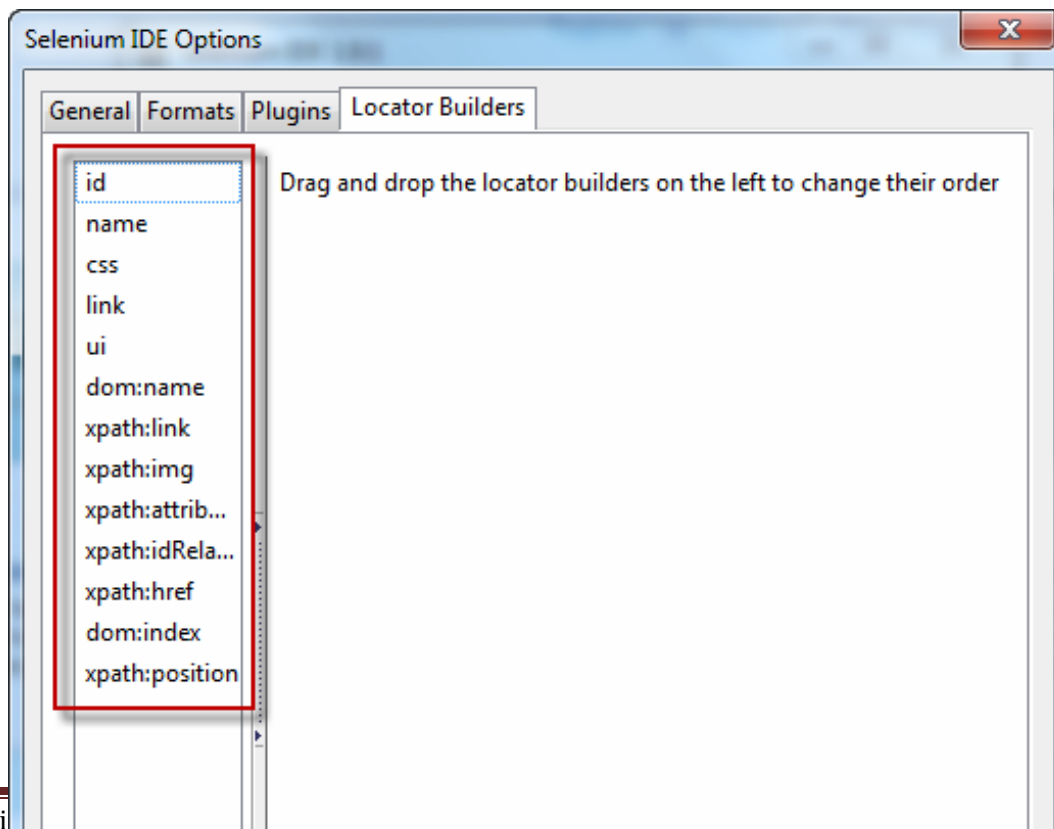
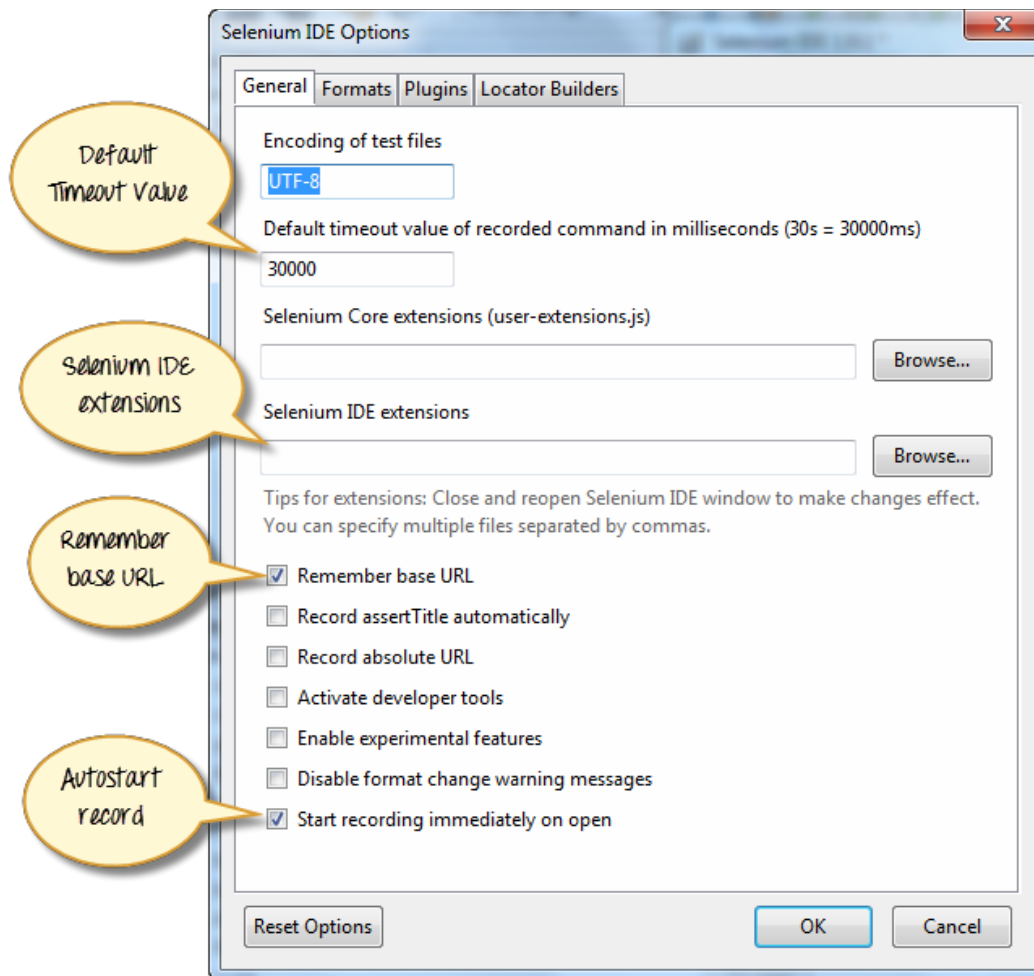


WebDriver test case



The two most important options in the Edit menu

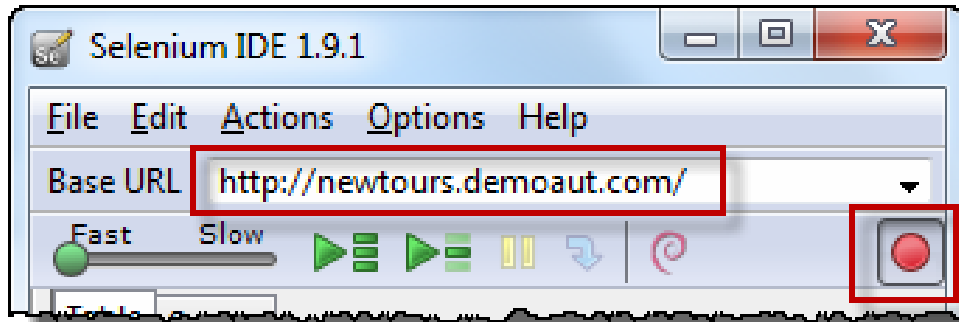




Create a Script by Recording:

Step 1:

- Launch Firefox and Selenium IDE.
- Type the value for our Base URL: <http://newtours.demoaut.com/>.
- Toggle the Record button on (if it is not yet toggled on by default).



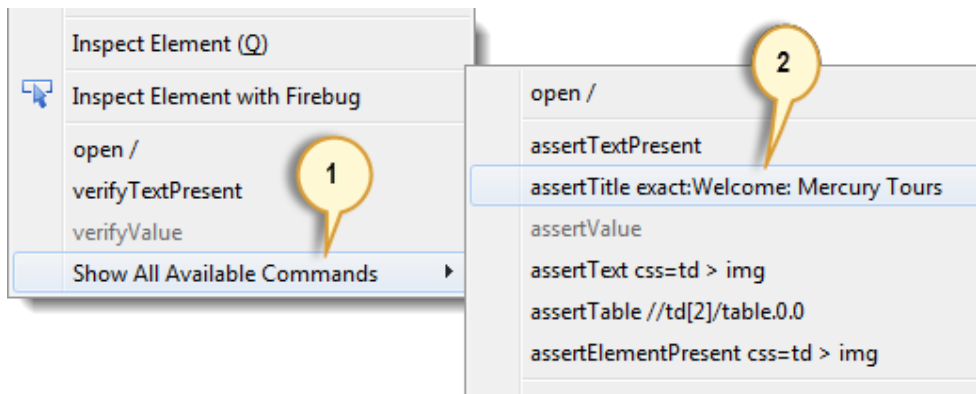
Step 2:

- In Firefox, navigate to <http://newtours.demoaut.com/>. Firefox should take you to the page similar to the one shown below.



Step 3:

- Right-click on any blank space within the page, like on the Mercury Tours logo on the upper left corner. This will bring up the Selenium IDE context menu. Note: Do not click on any hyperlinked objects or images
- Select the “Show Available Commands” option.
- Then, select “assertTitle exact:Welcome: Mercury Tours”. This is a command that makes sure that the page title is correct.

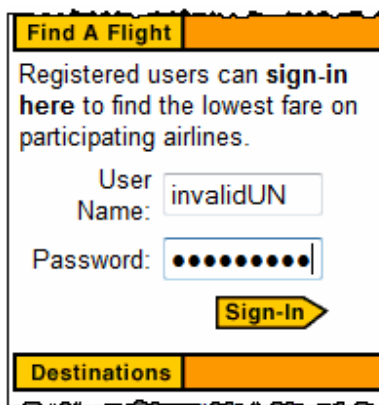


Comm...	Target	Value
open	/	
assertTitle	exact:Welcome: Mercury Tours	

After clicking on the assertTitle context menu option, your Selenium IDE Editor pane should now contain the following commands

Step 4:

- In the “User Name” text box of Mercury Tours, type an invalid username, “invalidUN”.
- In the “Password” text box, type an invalid password, “invalidPW”.



Comm...	Target	Value
open	/	
assertTitle	exact:Welcome: Mercury Tours	
type	name=username	invalidUN
type	name=password	invalidPW

Your Editor should now look like this

Step 5:

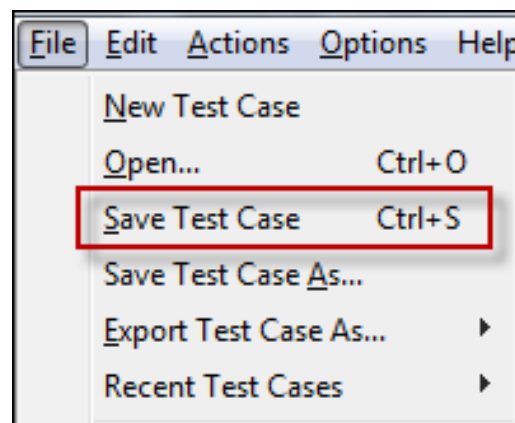
Step 6:

- Toggle the record button off to stop recording. Your script should now look like the one shown below.

Command	Target	Value
open		
assertTitle	exact:Welcome: Mercury Tours	
type	name=username	invalidUN
type	name=password	invalidPW
clickAndWait	name=login	

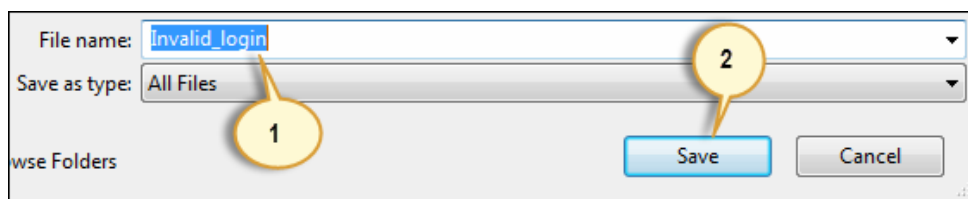
Step 7:

- Now that we are done with our test script, we shall save it in a test case. In the File menu, select "Save Test Case". Alternatively, you can simply press Ctrl+S.

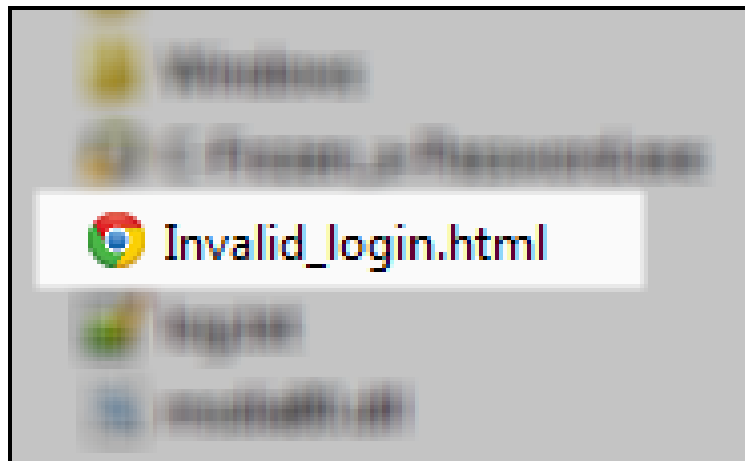


Step 8:

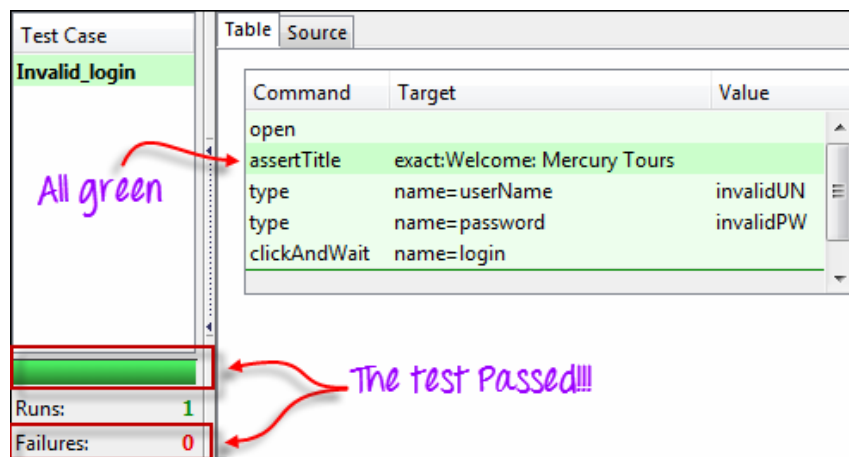
- Choose your desired location, and then name the test case as “Invalid_login”.
- Click the “Save” button.

**Step 9:**

- Notice that the file was saved as HTML.

**Step 10:**

- Go back to Selenium IDE and click the Playback button to execute the whole script. Selenium IDE should be able to replicate everything flawlessly.

**Common Commands:**

Command	No. of Parameters	Description
open	0 - 2	Opens a page using a URL.
click/clickAndWait	1	Clicks on a specified element.
type/typeKeys	2	Types a sequence of characters.

verifyTitle/assertTitle

1

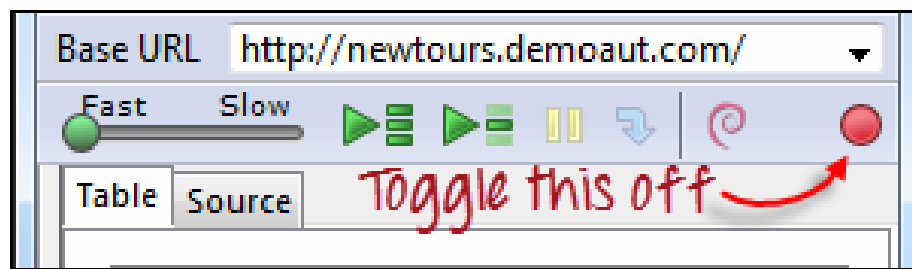
Compares the actual page title with an expected

verifyTextPresent	1	Checks if a certain text is found within the page.
verifyElementPresent	1	Checks the presence of a certain element.
verifyTable	2	Compares the contents of a table with expected
waitForPageToLoad	1	Pauses execution until the page is loaded
waitForElementPresent	1	Pauses execution until the specified element

Create a Script Manually with Firebug: Step

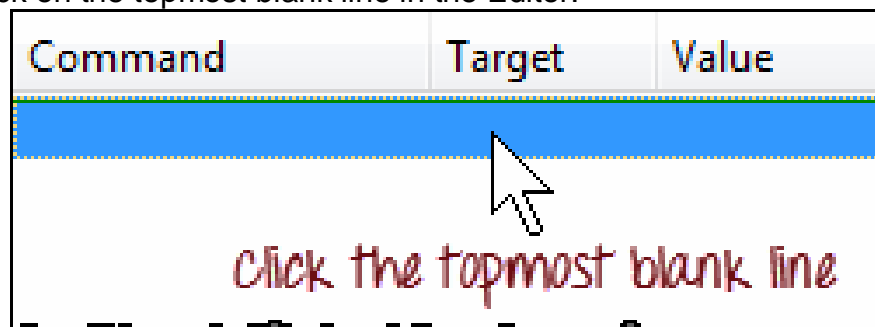
1:

- Open Firefox and Selenium IDE.
- Type the base URL (<http://newtours.demoaut.com/>).
- The record button should be OFF.

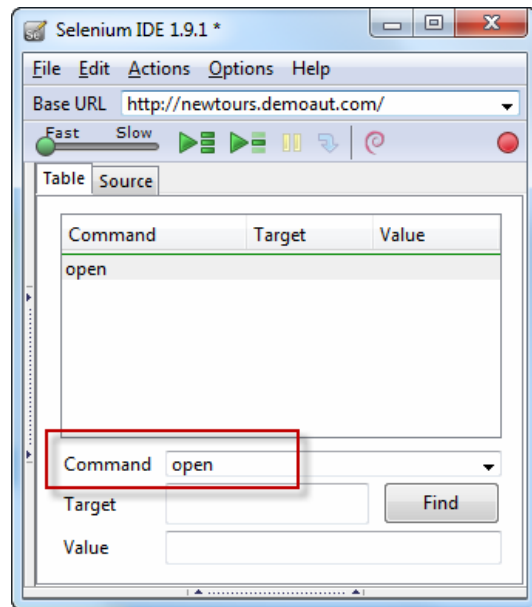


Step 2:

- Click on the topmost blank line in the Editor.

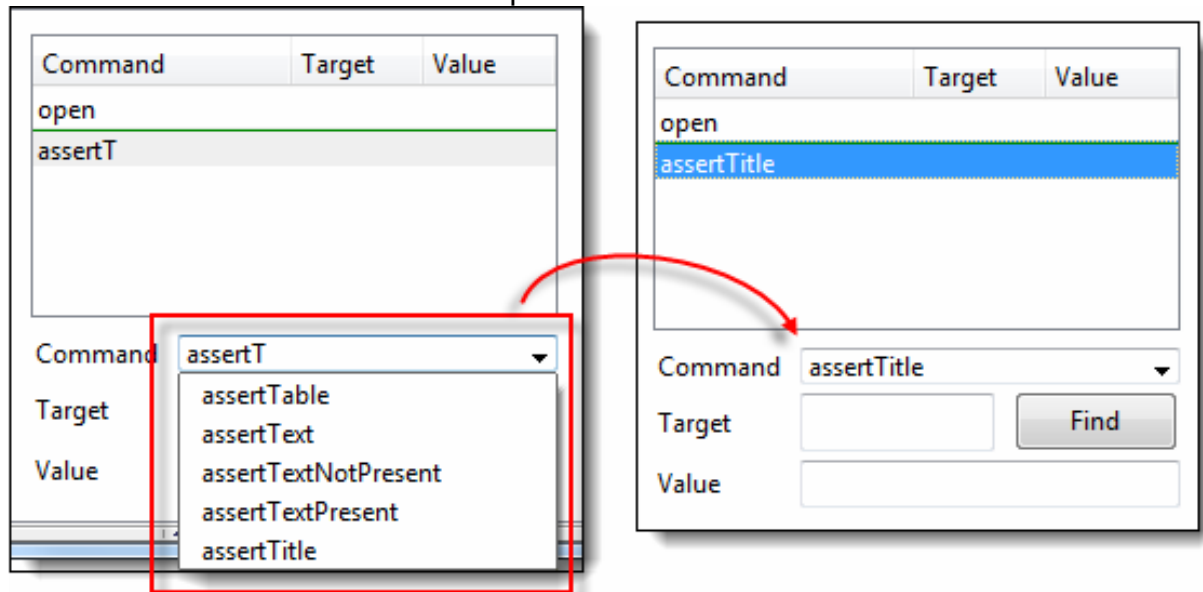


- Type "open" in the Command text box and press Enter.



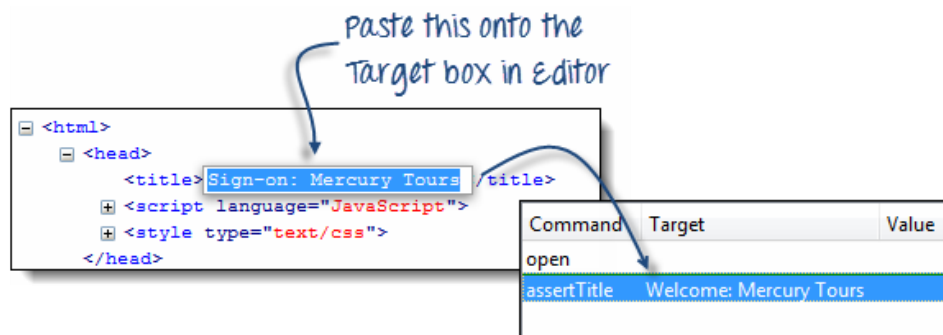
Step 3:

- Navigate Firefox to our base URL and activate Firebug
- In the Selenium IDE Editor pane, select the second line (the line below the “open” command) and create the second command by typing “assertTitle” on the Command box.
- Feel free to use the autocomplete feature.



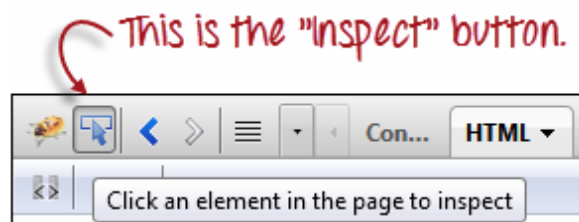
Step 4:

- In Firebug, expand the <head> tag to display the <title> tag.
- Click on the value of the <title> tag (which is “Welcome: Mercury Tours”) and paste it onto the Target field in the Editor.

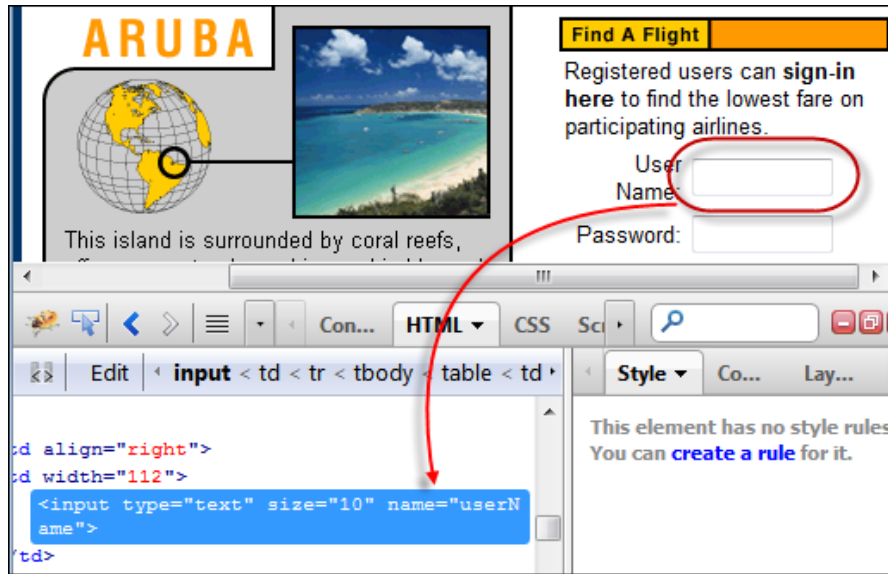


Step 5:

- To create the third command, click on the third blank line in the Editor and key-in “type” on the Command text box.
- In Firebug, click on the “Inspect” button.

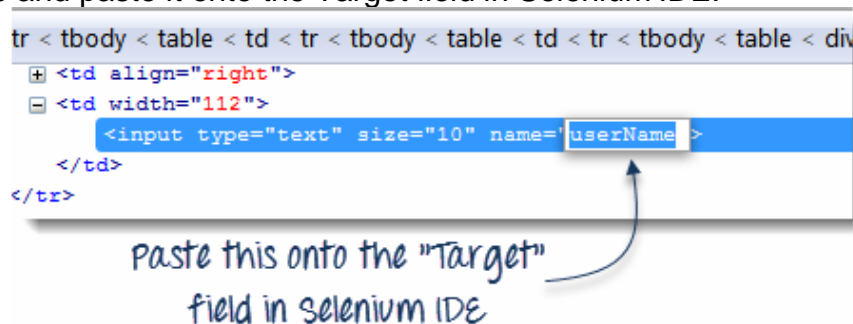


- Click on the User Name text box. Notice that Firebug automatically shows you the HTML code for that element.



Step 6:

- Notice that the User Name text box does not have an ID, but it has a NAME attribute. We shall, therefore, use its NAME as the locator. Copy the NAME value and paste it onto the Target field in Selenium IDE.



- Still in the Target text box, prefix "userName" with "name=", indicating that Selenium IDE should target an element whose NAME attribute is "userName."

Command	Target	Value
open		
assertTitle	exact:Welcome: Mercury Tours	
type	name=userName	

- Type "invalidUN" in the Value text box of Selenium IDE. Your test script should now look like the image below. We are done with the third command. Note: Instead of invalidUN , you may enter any other text string. But Selenium IDE is case sensitive and you type values/attributes exactly like in application.

Command	Target	Value
open		
assertTitle	Welcome: Mercury Tours	
type	name=userName	invalidUN

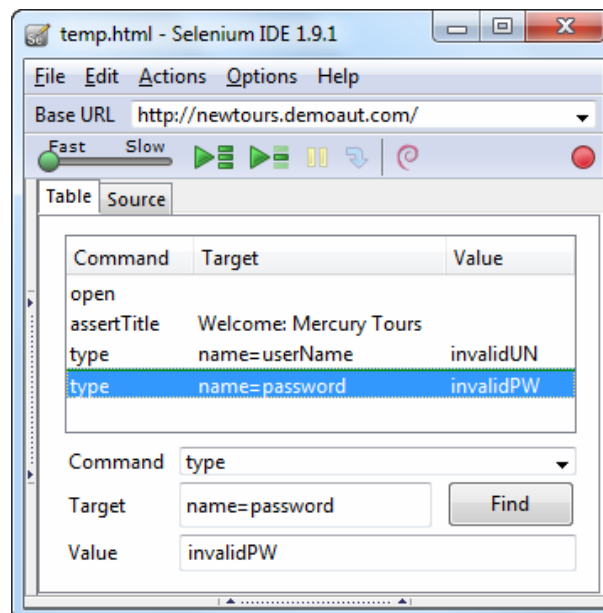
Step 7:

- To create the fourth command, key-in “type” on the Command text box.
- Again, use Firebug’s “Inspect” button to get the locator for the “Password” text box.

```
<tr>
  <td align="right">
    <td width="112">
      <input type="password" size="10" name="password">
    </td>
  </tr>
</tr>
```

The Password text box only has the NAME attribute so we will use it as our locator

- Paste the NAME attribute (“password”) onto the Target field and prefix it with “name=”
- Type “invalidPW” in the Value field in Selenium IDE. Your test script should now look like the image below.



Step 8:

- For the fifth command, type “clickAndWait” on the Command text box in Selenium IDE.
- Use Firebug’s “Inspect” button to get the locator for the “Sign In” button.

```
<td width="112">
  <div align="center">
    <input width="58" type="image" height="17" border="0"
      alt="Sign-In" src="/images
      /btn_signin.gif" value="Login" name="login">
    </div>
  </td>
```

Again, the only available locator is the NAME attribute so this will be the one that we shall use.

- Paste the value of the NAME attribute ("login") onto the Target text box and prefix it with "name=".
- Your test script should now look like the image below.

The image shows a screenshot of a test script editor. At the top, there is a table with three columns: Command, Target, and Value. The table contains the following rows:

Command	Target	Value
open		
assertTitle	Welcome: Mercury Tours	
type	name=username	invalidUN
type	name=password	invalidPW
clickAndW...	name=login	

The row with Command "clickAndW..." and Target "name=login" is selected. A red arrow points from this row to a detailed view of the command below. This detailed view has a red border and contains the following fields:

Command	clickAndWait
Target	name=login
Value	

Below the detailed view is a "Find" button and another empty "Value" field.

Step 9:
Save the test case

PRACTICAL-2

Problem Statement:

Conduct a test suite for any two web sites.

Theory:

Selenium Web Driver:

WebDriver enables you to **use a programming language** in creating your test scripts(not possible in Selenium IDE).

- You can now use **conditional operations** like if-then-else or switch-case
- You can also perform **looping** like do-while.
- Following programming languages are supported by WebDriver
 - Java • .Net
 - PHP • Python
 - Perl • Ruby
- **You do not have to know all of them. You just need to be knowledgeable in one.**
- Before advent of WebDriver in 2006, there was another, **automation tool called Selenium Remote Control**. Both WebDriver and Selenium RC have following features:
 - They both allow you to **use a programming language** in designing your test scripts.
 - They both allow you to **run your tests against different browsers**.
 - **WebDriver's architecture is simple.**
 - It controls the browser from the OS level
 - All you need are your programming language's IDE (which contains your Selenium commands) and a browser.
 - ☐ **WebDriver can support the headless HtmlUnit browser.**
 - ☐ HtmlUnit is termed as "headless" because it is an invisible browser – it is GUI-less.
 - ☐ It is a very fast browser because no time is spent in waiting for page elements to load. This accelerates your test execution cycles.
 - ☐ Since it is invisible to the user, it can only be controlled through automated means.
 - ☐ **Selenium RC cannot support the headless HtmlUnit browser.** It needs a real, visible browser to operate on.

Limitations of Webdriver:

- ☐ WebDriver Cannot Readily Support New Browsers
- ☐ WebDriver operates on the OS level. Also remember that different browsers communicate with the OS in different ways. If a new browser comes out, it may have a different process of communicating with the OS as compared to other browsers.
- ☐ However, it is up to the WebDriver's team of developers to decide if they should support the new browser or not.

Installing Selenium WebDriver:

- Step 1 – Install Java on your computer
- Step 2 – Install Eclipse IDE
- Step 3 – Download the Selenium Java Client Driver
- Step 4 – Configure Eclipse IDE with WebDriver

- Launch the “eclipse.exe” file inside the “eclipse” folder that we extracted in step 2. If you followed step 2 correctly, the executable should be located on C:\eclipse\eclipse.exe.
- When asked to select for a workspace, just accept the default location.
- Create a new project through File > New > Java Project. Name the project as “myproject”.
- Right-click on the newly created project and select New > Package, and name that package as “mypackage”.

• Create a new Java class under *mypackage* by right-clicking on it and then selecting New > Class, and then name it as “myclass”.

- ☐ Right-click on *myproject* and select Properties.
- ☐ On the Properties dialog, click on “Java Build Path”.
- ☐ Click on the Libraries tab, and then click “Add External JARs..”
- ☐ Navigate to C:\selenium-2.25.0\ (or any other location where you saved the extracted contents of “selenium-2.25.0.zip” in step 3).
- ☐ Add all the JAR files inside and outside the “libs” folder.
- ☐ Finally, click OK and we are done importing Selenium libraries into our project.

Using the Java class “myclass” that we created in the previous tutorial, let us try to create a WebDriver script that would:

- fetch Mercury Tours’ homepage
- verify its title
- print out the result of the comparison
- close it before ending the entire program.

```

1. package mypackage;
2. import org.openqa.selenium.WebDriver;
3. import org.openqa.selenium.firefox.FirefoxDriver;
4. public class myclass {
5.     public static void main(String[] args) {
6.         // declaration and instantiation of objects/variables
7.         WebDriver driver = new FirefoxDriver();
8.         String baseUrl = "http://newtours.demoaut.com";
9.         String expectedTitle = "Welcome: Mercury Tours";
10.        String actualTitle = "";
11.        // launch Firefox and direct it to the Base URL
12.        driver.get(baseUrl);
13.        // get the actual value of the title
14.        actualTitle = driver.getTitle();
15.        if (actualTitle.contentEquals(expectedTitle)){
16.            System.out.println("Test Passed!");
17.        } else {
18.            System.out.println("Test Failed");
19.        }
20.        //close Firefox
21.        driver.close();

```

```
22. System.exit(0);
23. }
24. }
```

Running the Test:

- There are two ways to execute code in Eclipse IDE.
 - On Eclipse's menu bar, click Run > Run.
 - Press Ctrl+F11 to run the entire code.
 - If you did everything correctly, Eclipse would output "Test Passed!"
- These are basic examples. More on selenium need to be studied.

PRACTICAL-3

AIM: Install Selenium server and demonstrate it using a script in Java/PHP.

Solution: (Do the following steps)

→Open eclipse →Launch→File→New→Java project

→Project name(first) →finish

→right click on first(left side top corner)→new→Class→Name(hello)→finish

Then type the following code

```
package hello;
import org.testng.annotations.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class First
{
    @Test
    public void a()
    {
        System.setProperty("webdriver.gecko.driver",
            "C:\\\\Users\\patra\\Desktop\\STQA\\geckodriver-v0.21.0-
            win32\\geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com");

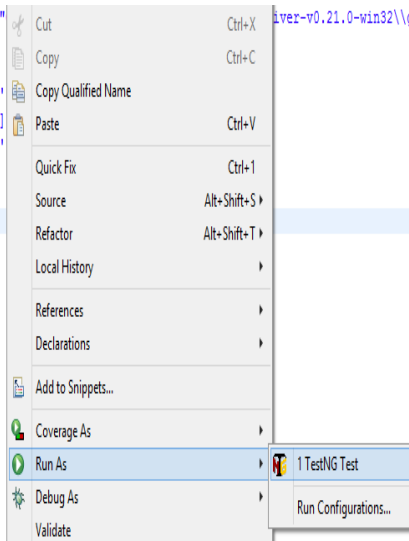
        driver.findElement(By.xpath("//*[@id='email']")).sendKeys("abc");
        driver.findElement(By.xpath("//*[@id='pass']")).sendKeys("abc");
        driver.findElement(By.xpath("//*[@id='u_0_2']")).click();
    }
}
```

OUTPUT

```

System.setProperty("webdriver.gecko.driver", "
WebDriver driver = new FirefoxDriver();
driver.get("https://www.facebook.com");
driver.findElement(By.xpath("//*[@id='email']
driver.findElement(By.xpath("//*[@id='pass']
driver.findElement(By.xpath("//*[@id='u_0_2']

```



Forgotten account?

Create an account

It's free and always will be.

Birthday

14

▼

Oct

▼

1993

▼

Why do I need to provide my date of birth?

☐ Female
 ☐ Male

By clicking Sign Up, you agree to our [Terms](#), [Data Policy](#) and [Cookie Policy](#). You may receive SMS notifications from us and can opt out at any time.

PRACTICAL – 4

AIM: Code to update 10 student records into table into Excel file

Selenium Tests with Microsoft Excel

Download the jxl.jar from “<http://jexcelapi.sourceforge.net/>”

→Open eclipse →Launch→File→New→Java project

→Project name(proj2) →finish

→right click on proj2(left side top corner)→new→Class→Name(DemoTest)→finish

Then type the following code

```

package proj2;

import jxl.Sheet;
import jxl.Workbook;
//Code to update 10 student records into table into Excel file
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
import jxl.write.*;
import java.io.*;

public class DemoTest {
    @BeforeClass
    public void f1()
    {}
}

```



```

@Test
public void testImportexport1() throws Exception {
    FileInputStream fi = new
    FileInputStream("C:\\Users\\patra\\Desktop\\STQA\\Book1.xls");
    Workbook w = Workbook.getWorkbook(fi);
    Sheet s = w.getSheet(0);
    String a[][] = new String[s.getRows()][s.getColumns()];
    FileOutputStream fo = new FileOutputStream("C:\\Users\\patra\\
        Desktop\\STQA\\e1.xls");
    WritableWorkbook wwb = Workbook.createWorkbook(fo);
    WritableSheet ws = wwb.createSheet("result1", 0);
    for (int i = 0; i < s.getRows(); i++)
    for (int j = 0; j < s.getColumns(); j++)
    {
        a[i][j] = s.getCell(j, i).getContents();
        Label l2 = new Label(j, i, a[i][j]);
        ws.addCell(l2);
        Label l1 = new Label(6, 0, "Result");
        ws.addCell(l1);
    }
    for (int i = 1; i < s.getRows(); i++) {
        for (int j = 2; j < s.getColumns(); j++)
        {
            a[i][j] = s.getCell(j, i).getContents();
            int x = Integer.parseInt(a[i][j]);
            if (x > 35)
            {
                Label l1 = new Label(6, i, "pass");
                ws.addCell(l1);
            }
            else
            {
                Label l1 = new Label(6, i, "fail");
                ws.addCell(l1);
                break;
            }
        }
        wwb.write();
        wwb.close();
    }
}

```

OUTPUT: create two excel file

book1.xls

rno	name	s1	s2	s3	tot
101	raj	36	45	13	94
102	rani	60	74	71	205
103	suraj	25	60	36	121
104	nikhil	85	70	62	217
105	ajay	50	25	18	93

e1.xls(create empty)

rno	name	s1	s2	s3	tot	result
101	raj	36	45	13	94	fail

102	rani	60	74	71	205	pass
103	suraj	25	60	36	121	fail
104	nikhil	85	70	62	217	pass
105	ajay	50	25	18	93	fail

PRACTICAL-5

```

package pract6;
import jxl.read.*;
import jxl.write.*;
import jxl.Sheet;
import jxl.Workbook;
import java.io.*;
import org.testng.annotations.Test;
public class mypackage {
    @Test
    public void testImportexport1() throws Exception
    {
        FileInputStream fi = new FileInputStream("C:\\Users\\patra\\Desktop\\STQA\\p6\\sampledata.xls");

        Workbook w = Workbook.getWorkbook(fi);
        Sheet s = w.getSheet(0);
        String a[][] = new String[s.getRows()][s.getColumns()];
        FileOutputStream fo = new FileOutputStream("C:\\Users\\patra\\Desktop\\STQA\\p6\\result.xls");

        WritableWorkbook ww = Workbook.createWorkbook(fo);
        WritableSheet ws = ww.createSheet("result", 0);
        int c=0;
        for (int i = 0; i < s.getRows(); i++) {
            for (int j = 0; j < s.getColumns(); j++)
            {
                if(i >= 1)
                {
                    String b= new String();
                    b=s.getCell(3,i).getContents();
                    int x= Integer.parseInt(b);
                    if( x < 60)
                    {
                        c++;
                        break;
                    }
                }
                a[i][j] = s.getCell(j, i).getContents();
                Label l2 = new Label(j, i-c, a[i][j]);
            }
        }
    }
}

```

```

        ws.addCell(12);
    }
}
wwb.write();
wwb.close();
}
}

```

OUTPUT: create two excel file

book1.xls

rno	name	s1	s2	s3	tot
101	raj	36	45	13	94
102	rani	60	74	71	205
103	suraj	25	60	36	121
104	nikhil	85	70	62	217
105	ajay	50	25	18	93

e1.xls(create empty)

rno	name	s1	s2	s3	tot	result
102	rani	60	74	71	205	pass
103	suraj	25	60	36	121	pass
104	nikhil	85	70	62	217	pass

PRACTICAL-6

```

package pract7;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Test;

public class p7 {
    @Test
    public static void a()
    {

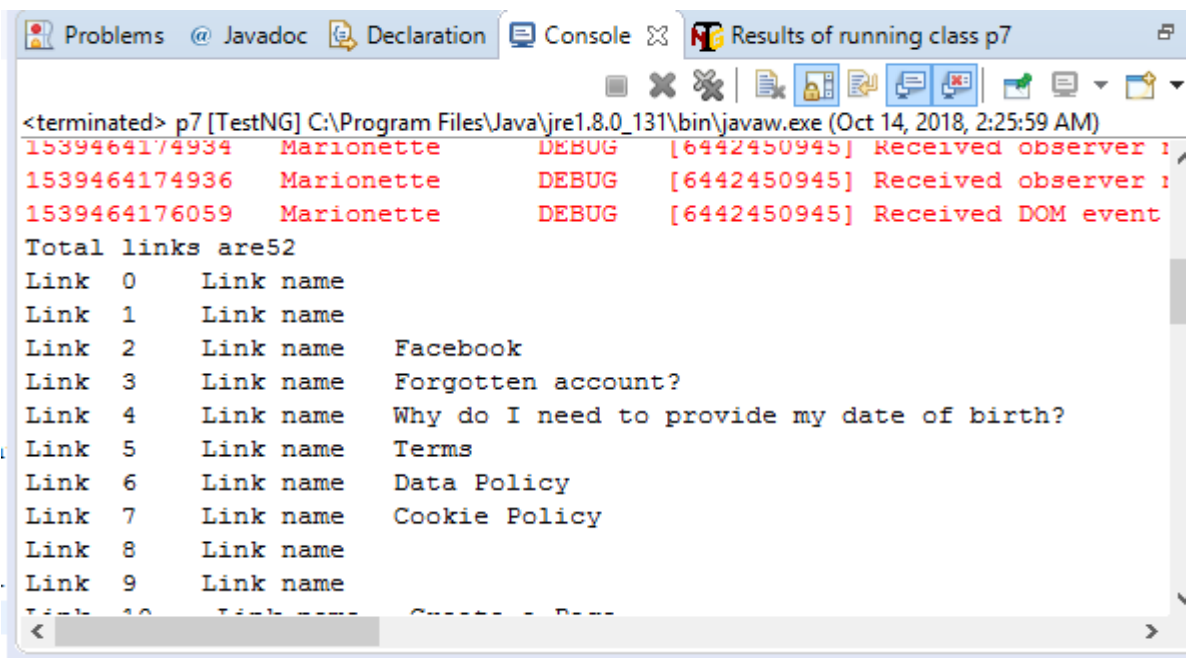
```

```

System.setProperty("webdriver.gecko.driver", "C:\\\\Users\\\\patra\\\\Desktop\\\\STQA\\\\geckodriver-v0.21.0-
win32\\geckodriver.exe");
WebDriver driver = new FirefoxDriver();
driver.get("https://www.facebook.com");
java.util.List<WebElement> links = driver.findElements(By.tagName("a"));
System.out.println("Total links are"+links.size());
for (int i = 0; i<links.size(); i=i+1)
{
System.out.println("Link "+ i + "   Link name  "+ links.get(i).getText());
}
}
}

```

OUTPUT



PRACTICAL-7

AIM: Write and test a program to get the number of items in a list / combo box.

- Write this java test code in eclipse. before running this code makes sure your selenium RC server is running)
- Write Combocount.html as shown below and save under desktop

Follow the same procedure as above

```

<html>
<body>
<select>
<option>Volvo</option>
<option>Express</option>
<option>Mercedes</option>
<option>RajaHamsa</option>
</select>
</body>
</html>

```

- Write the below code in eclipse
- Make sure that the path of your combocount.html file must be correct for your system , in this

example it is

file:///C:/Users/Savitha/Desktop/combocount.html

```

package pract8;
import com.thoughtworks.selenium.*;
    public class combolist extends SeleneseTestCase
{
    @Test
    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "*chrome",
        "file:///C:\\Users\\patra\\Desktop\\STQA\\combo.html");
        selenium.start(); }
    public void testloginlogout(){
        selenium.setSpeed("1000");
        selenium.open("file:///C:/Users/Savitha/Desktop/combocount.html");
        selenium.waitForPageToLoad("30000");
        selenium.windowMaximize();
        String[] selectelements = new String[1000];
        selectelements= selenium.getSelectOptions("//select");
        System.out.println("The number of options present in the given select box is "
        +selectelements.length);

    }
}

```

OUTPUT :

[Selenium](#) Functional Testing for Web Apps

Open Source From [ThoughtWorks](#) and Friends

Show Log

☐ Slow Mode

7dde4bdafab1499eaa873512140b85e8

Command History:

```
getTitle()  
setContext (7dde4bdafab1499eaa873512140b85e8)  
open (file:///C:/Users/Savitha/Desktop/combocount.html)  
waitForPageToLoad(30000)  
windowMaximize()  
getSelectOptions (//select)
```

