INDEX

Sr.No	Date	Practical List	Sign.
1		**Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.**	
2		**Conduct a test suite for any two web sites.**	
3		**Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP.**	
4		**Arithmetic Operation (using junit)**	
5		**Write and test a program to specific web page.**	
6		**Write and test a program to update 10 student records into table into Excel file**	
7		**Write and test a program to select the number of students whose total is more than** **10** **0**	
8		**Write and test a program to provide total number of objects present / available on the page.**	
9		**Write and test a program to get the number of items in a list / combo box. Combo.html:**	
10		**Write and test a program to count number of items present on a desktop.**	

Practical no 1

Aim: Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.

1)What is Selenium?

• Acceptance testing tool for web-apps
• Tests run directly in browser
• Selenium can be deployed on Windows, Linux, and Macintosh.
• Implemented entirely using browser technologies -
- JavaScript
- DHTML
- Frames

• The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases.
• It is Firefox plug-in
• Firefox extension which allows record/play testing paradigm
• Automates commands, but asserts must be entered by hand
• Creates the simplest possible Locator
• Based on Selene's

2)Selenium Components

– Selenium IDE
Created by Shinya Kasatani of Japan
– Selenium RC
Created by Paul Hammant
– Selenium Grid
Developed by Patrick Lightbody

We 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 to install Selenium IDE-

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 IDEicon** 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.

perform test cases on 4 different websites

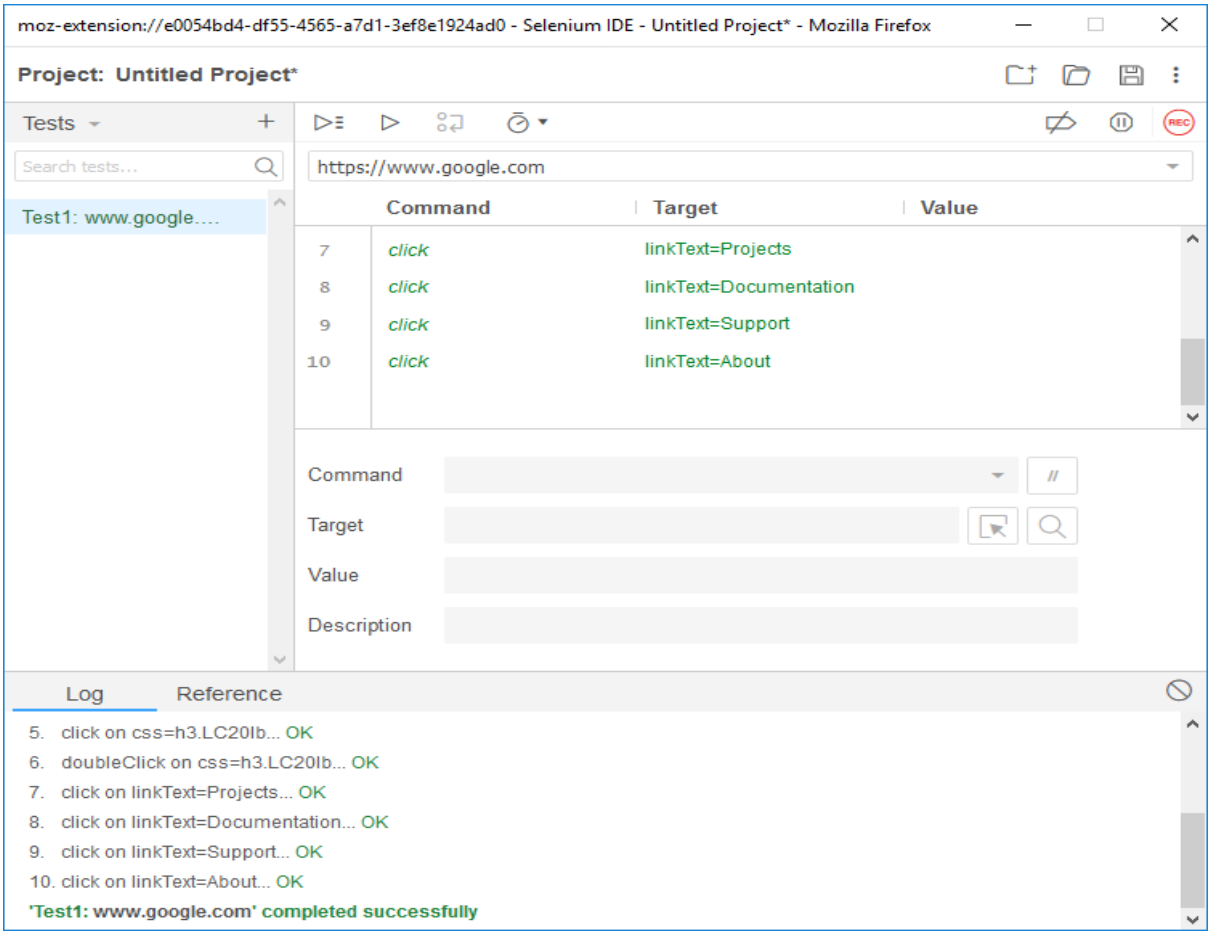
i. http://store.demoqa.com
ii. https://www.seleniumhq.org
iii. htttp://www.google.com
iv. htttp://www.yahoo.com

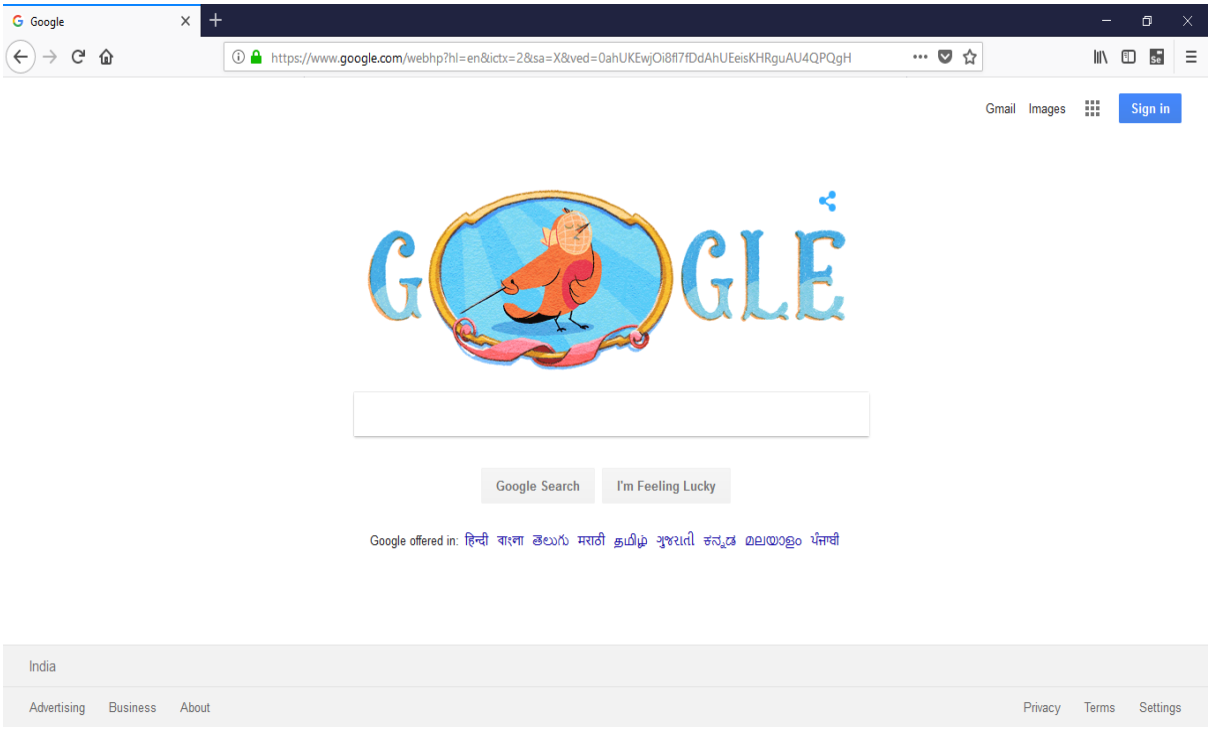
Steps:

- 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).
- In Firefox, navigate to http://demo.guru99.com/test/newtours/. Firefox should take you to the page similar to the one shown below. Navigate over the site which in turn get recorded on the Selenium.
- Go to the Selenium window again and stop the recording, by toggling the Record button again. Then run the current test case to get output.

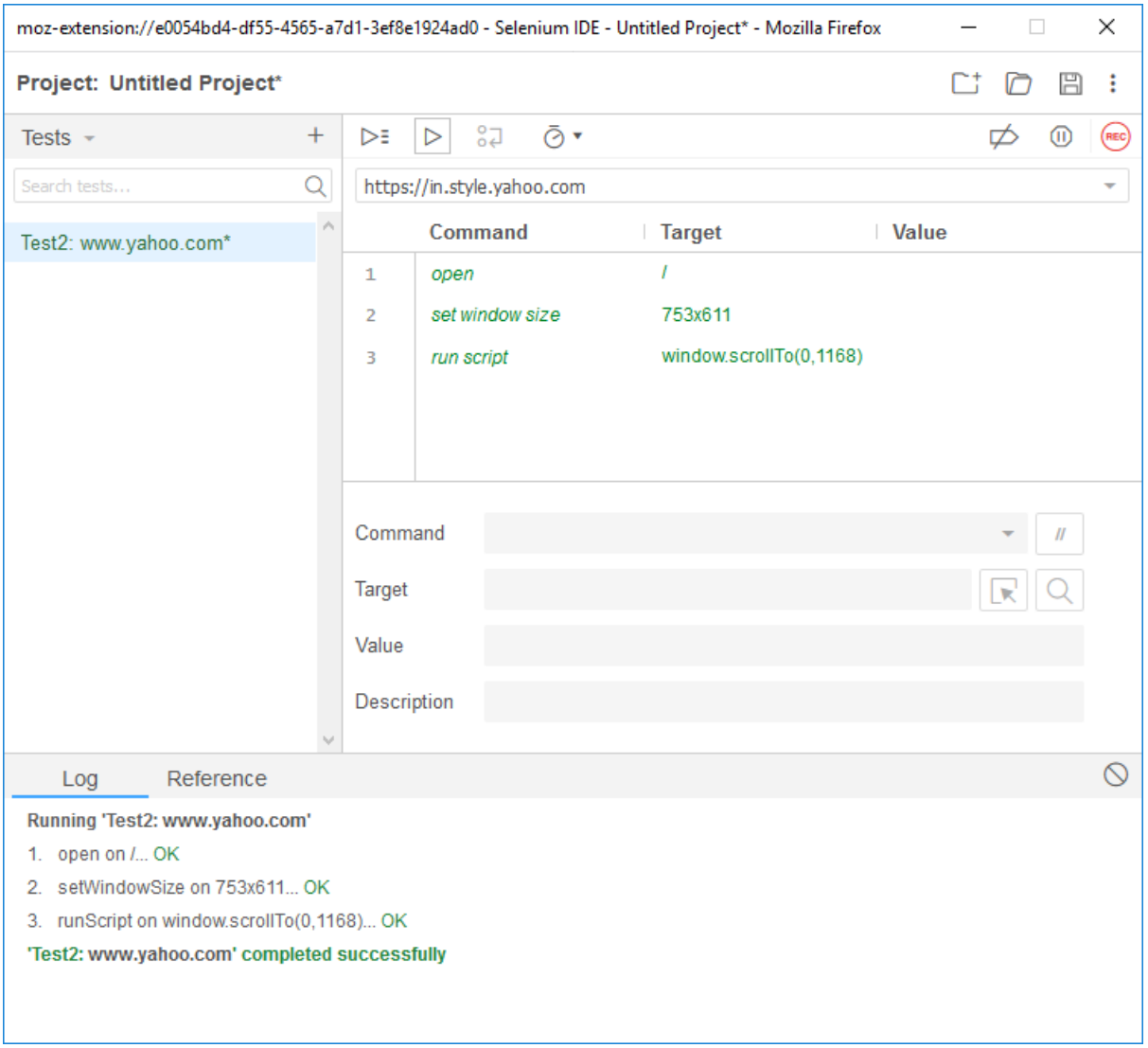
OUTPUT:

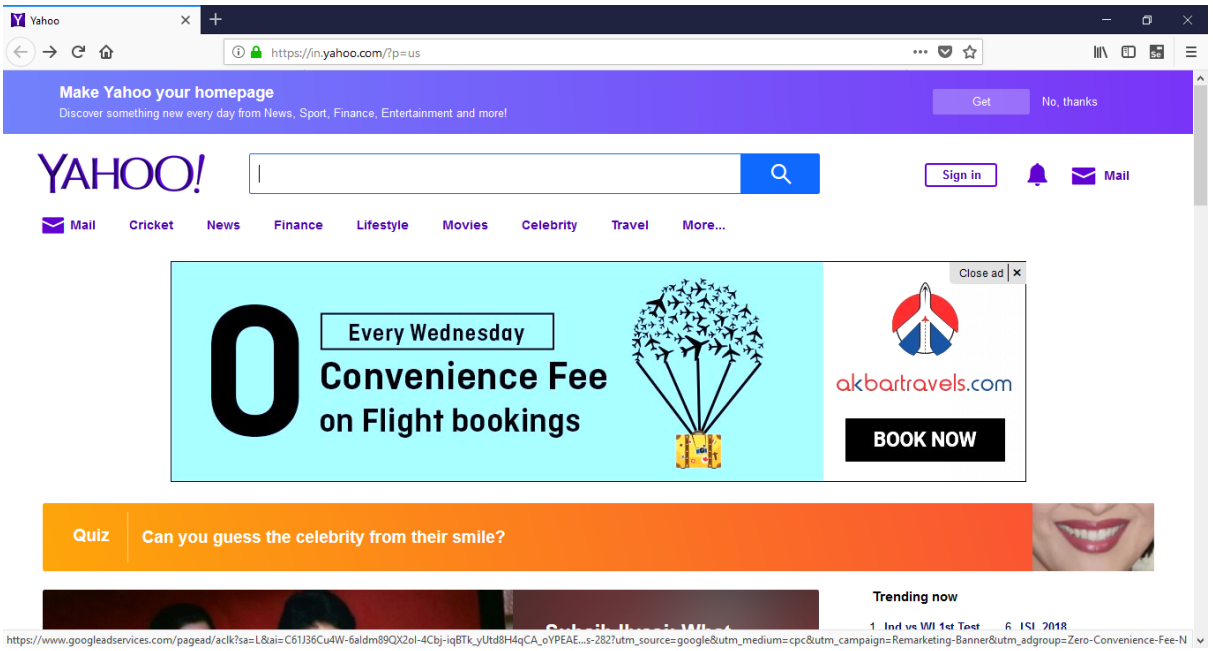
i] Test1: www.google.com


moz-extension://e0054bd4-df55-4565-a7d1-3ef8e1924ad0 - Selenium IDE - Untitled Project* - Mozilla Firefox
Project: Untitled Project*
Tests
Search tests...
Test1: www.google....
https://www.google.com
Command
Target
Value
7 click linkText=Projects
8 click linkText=Documentation
9 click linkText=Support
10 click linkText=About
Command
Target
Value
Description
Log
Reference
5. click on css=h3.LC20lb... OK
6. doubleClick on css=h3.LC20lb... OK
7. click on linkText=Projects... OK
8. click on linkText=Documentation... OK
9. click on linkText=Support... OK
10. click on linkText=About... OK
'Test1: www.google.com' completed successfully

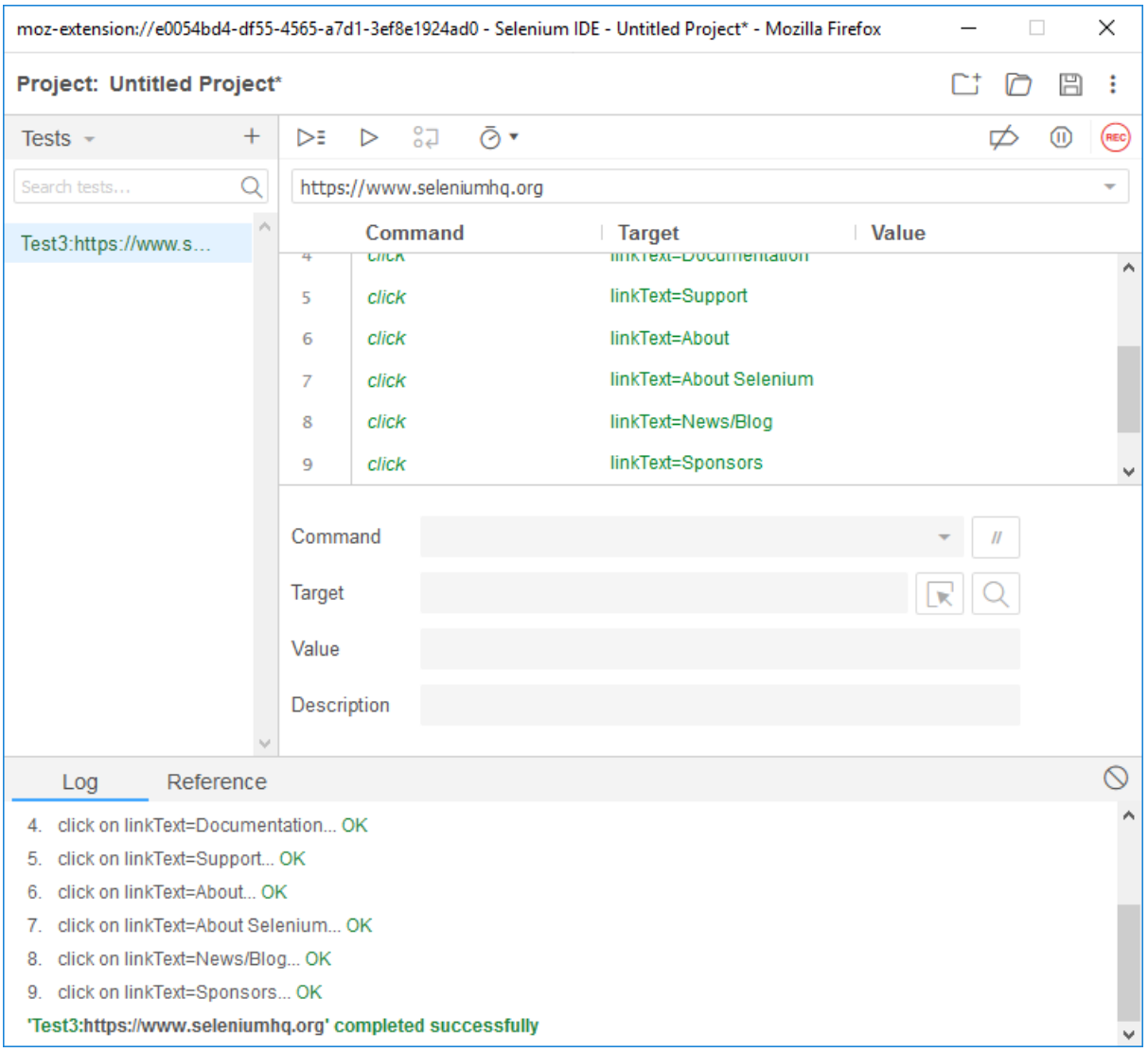

Google
https://www.google.com/webhp?hl=en&ictx=2&sa=X&ved=0ahUKEwjOi8fl7fDdAhUEeisKHRguAU4QPQgH
Gmail
Images
Sign in
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
India
Advertising
Business
About
Privacy
Terms
Settings

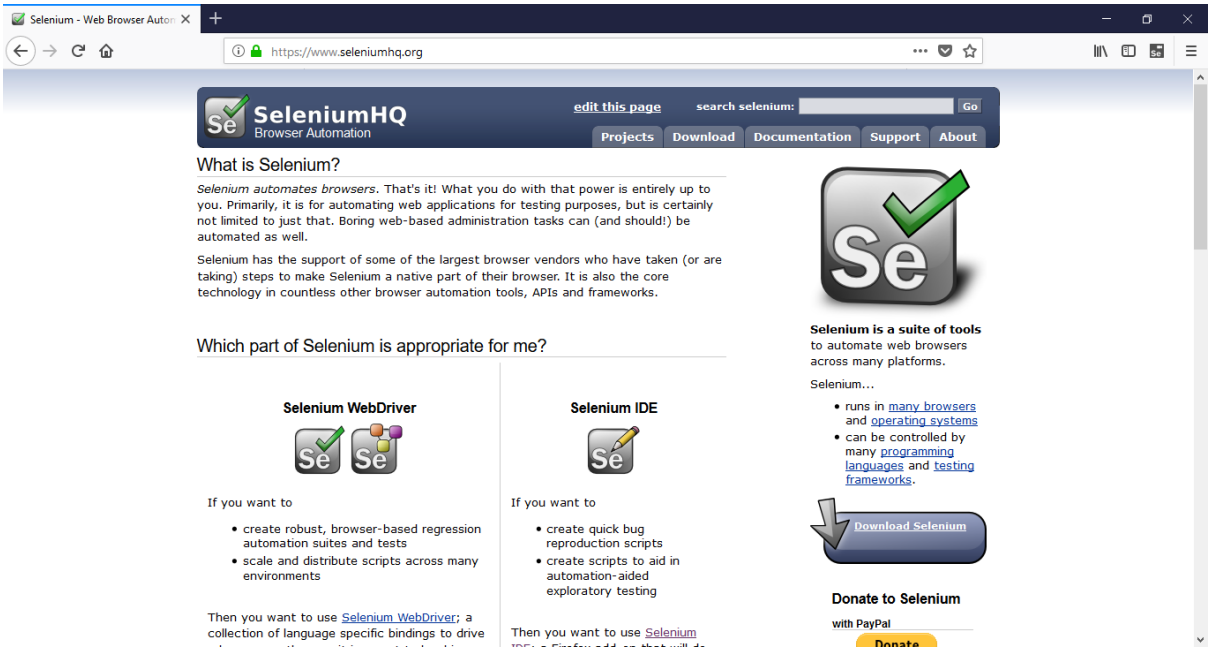
Ii] Test2: www.yahoo.com



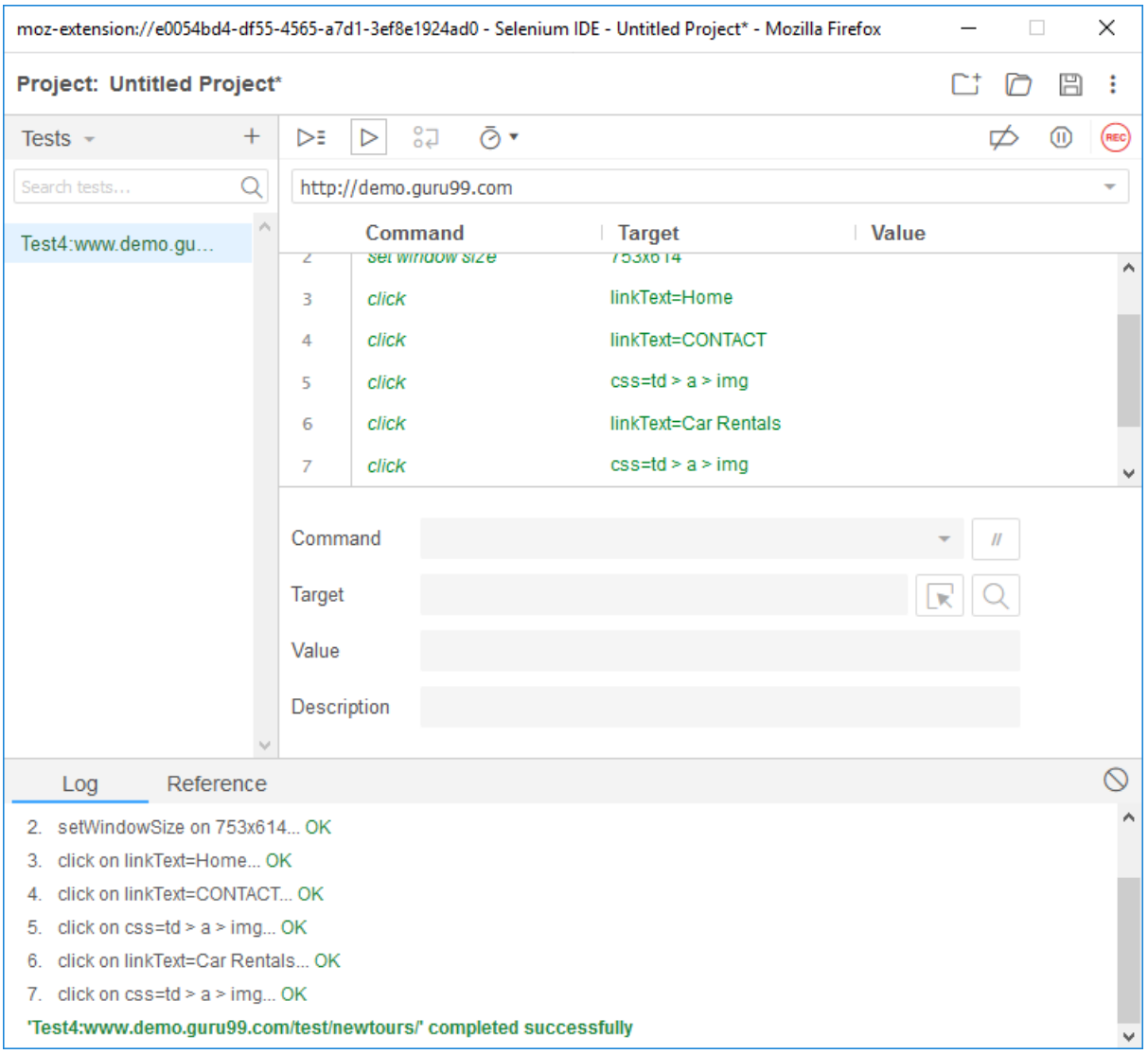


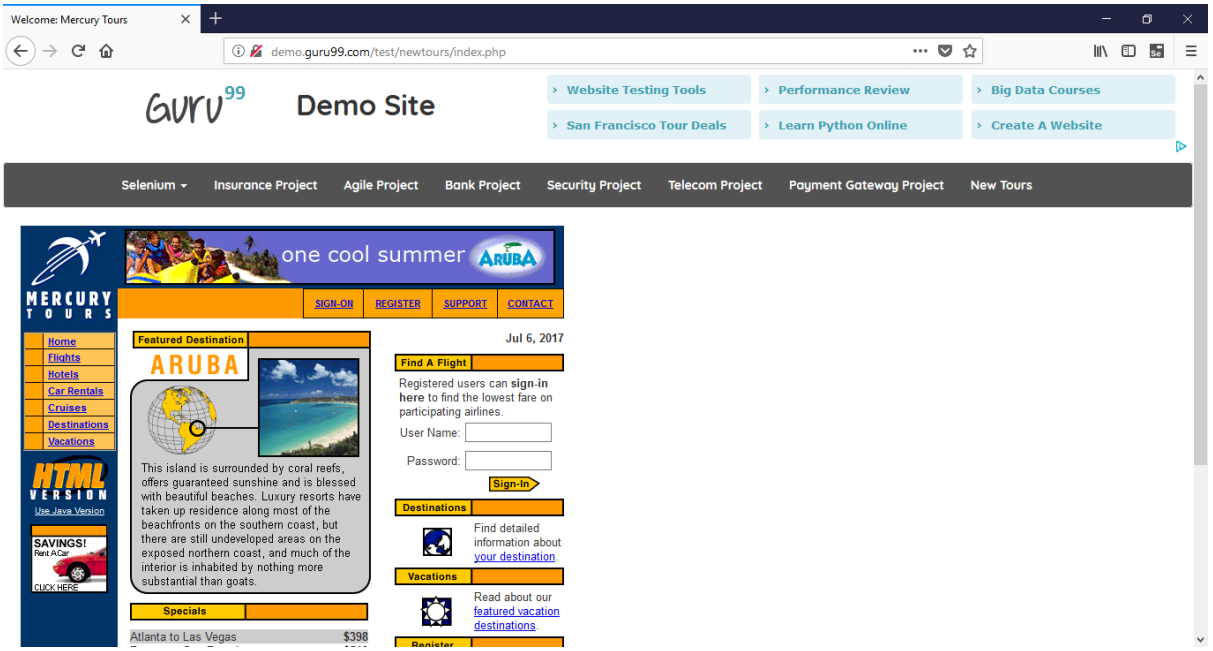
iii] Test3: www.seleniumhq.com





Iv] Test4: www.demo.guru99.in





Practical no -2

Aim: Conduct a Test suite for any two websites.

Required Software

1. Mozilla Firefox 61 (If Earlier version of Firefox is installed in the system, then that should be uninstalled and version 61 or above has to be installed)
2. Selenium IDE

Prerequisite Knowledge

1. Working of Selenium IDE
2. Knowledge of Selenium Commands

Basic Steps to follow for every test case

1. Open firefox browser
2. Open Selenium IDE
3. Type http://www.google.com/ in playback URL
4. Start Recording
5. Type http://www.google.com/ in browser address bar
6. Type selenium IDE as the search text (Don't use autocomplete option. Type yourself)
7. Press Enter
8. First search result will be seleniumhq website. Click on it

To check the working of assert title command

1. Perform the basic steps
2. Right_click anywhere on the screen.
3. Select Selenium IDE option and choose assert title command
4. Click on the hyperlink – Documentation
5. Browser will be redirected to Documentation page
6. Return to Selenium IDE and Stop Recording
7. Run the test case
8. Save the test case as "valid_assert_title
9. Create a new test case and name it as "invalid_assert_title"
10. Repeat the basic steps. Repeat steps 1 to 6 given above
11. Select assert title command from the steps
12. Change the target from "Selenium IDE" to "Selenese"
13. Stop Recording and Run the test case

To check the working of verify title command

1. Repeat the steps given for assert title command. But, instead of choosing assert title command from Selenium IDE option, choose verify title command. Give appropriate names to test cases (valid_verify_title and invalid_verify_title)

To check the working of assert text command

1. Create a new test case and name it as “valid_assert_text”
2. Perform the basic steps
3. In the seleniumHq page, select and highlight the text selenium IDE from the second paragraph
4. Right click and select selenium ide option and choose assert text command
5. Click on the hyperlink – Documentation
6. Browser will be redirected to Documentation page
7. Return to Selenium IDE, Stop Recording
8. Run the test case
9. Create a new test case and name it as “invalid_assert_text”
10. Repeat the basic steps. Repeat steps 3 to 7 given above
11. Select assert text command from the steps
12. Change the value from “Selenium IDE” to “Selenese”
13. Stop Recording and Run the test case

To check the working of verify text command

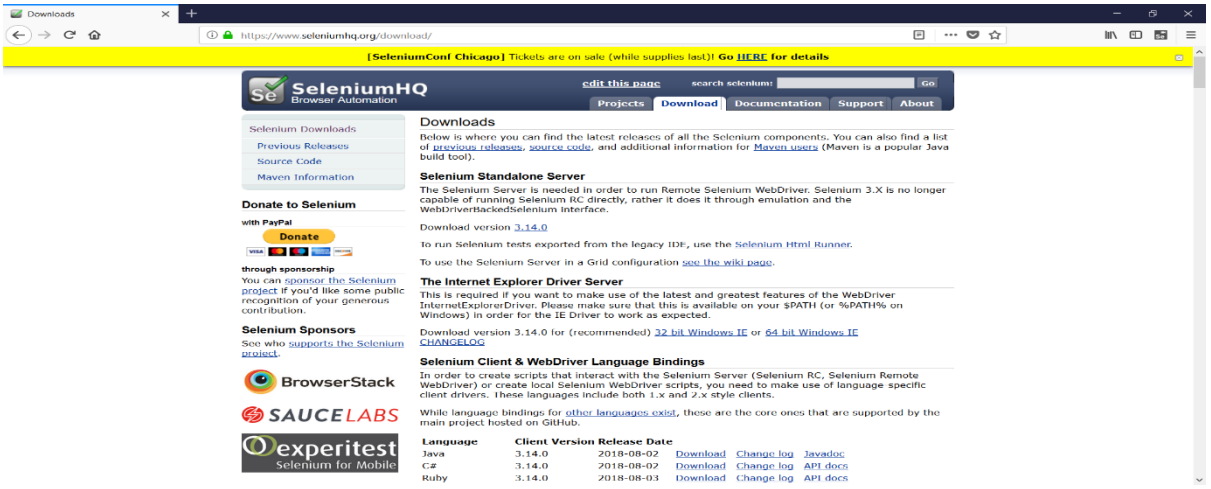
Repeat the steps given for assert text command. But, instead of choosing assert text command from Selenium IDE option, choose verify text command. Give appropriate names to test cases.

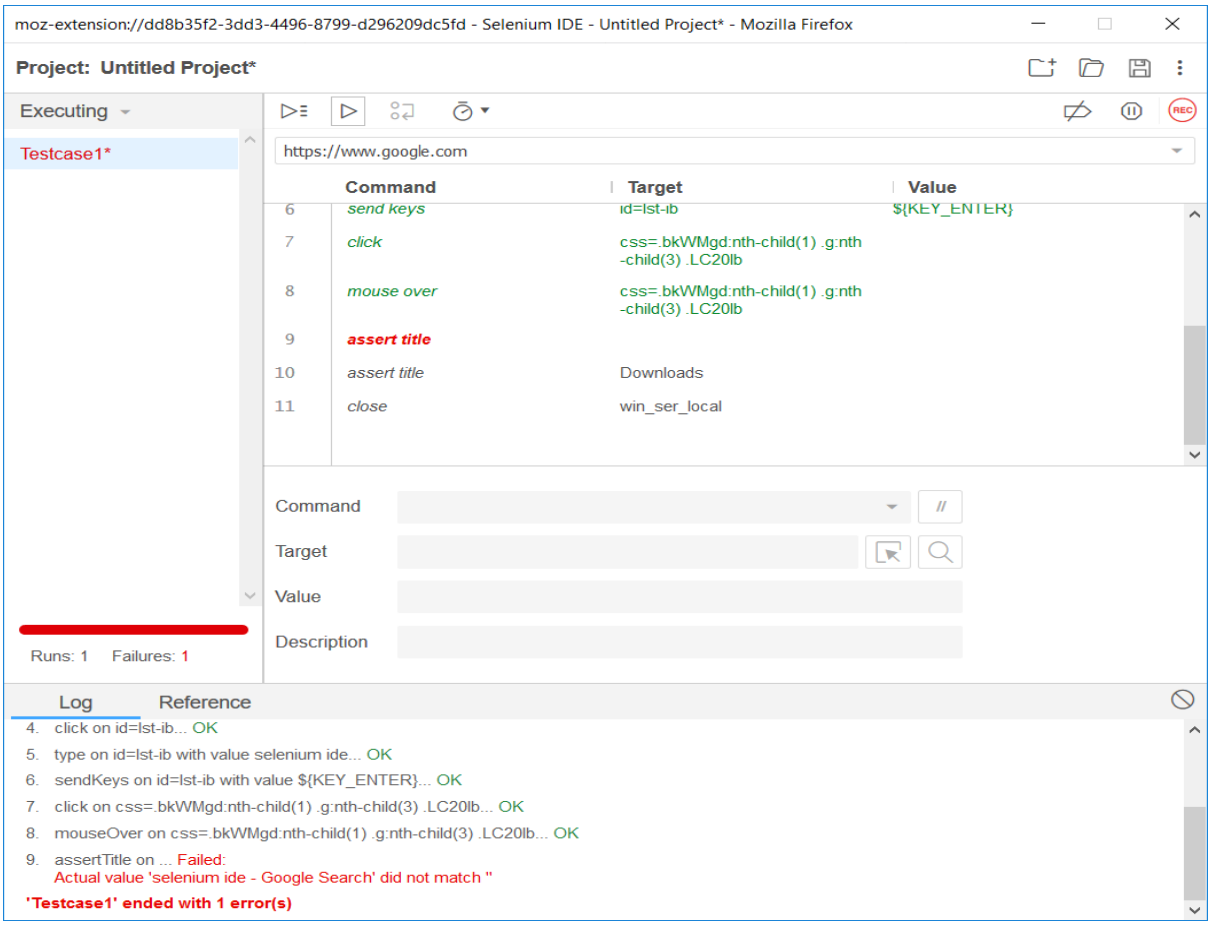
To check the working of verify element present

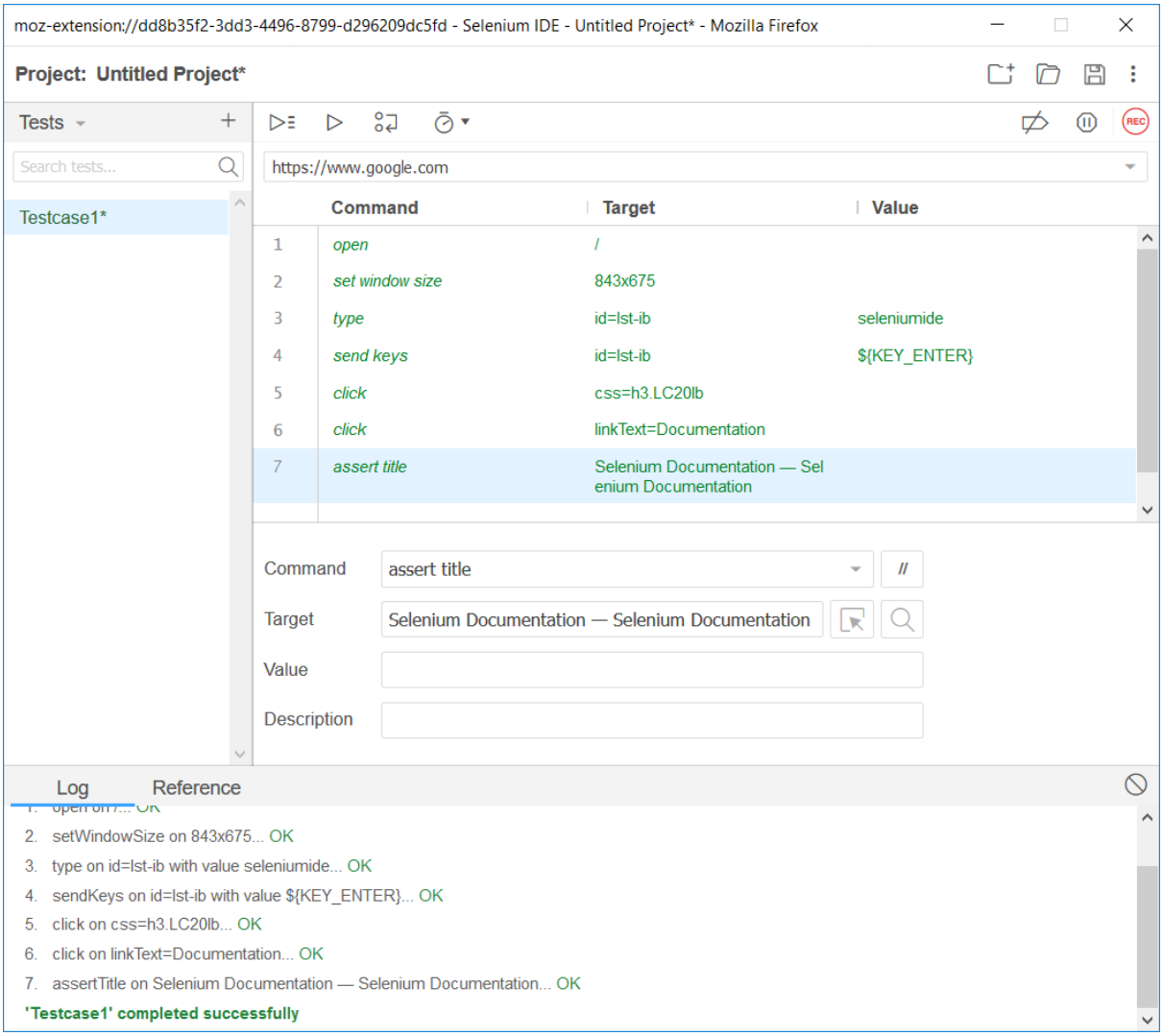
1. Perform the basic steps
2. Click on the hyperlink documentation
3. Return to Selenium IDE screen
4. Stop Recording
5. Click on the empty space at the end in the command-target-value area. Add a new command at the end
6. In command option, choose verify element present
7. In target option, type //div/p
8. Save the test case as valid_element and run the test case
9. Run the test case
10. Add a new test case and rename as invalid_element
11. Repeat steps from 1 to 6 given above
12. In target option, type //div/link
13. Run the test case

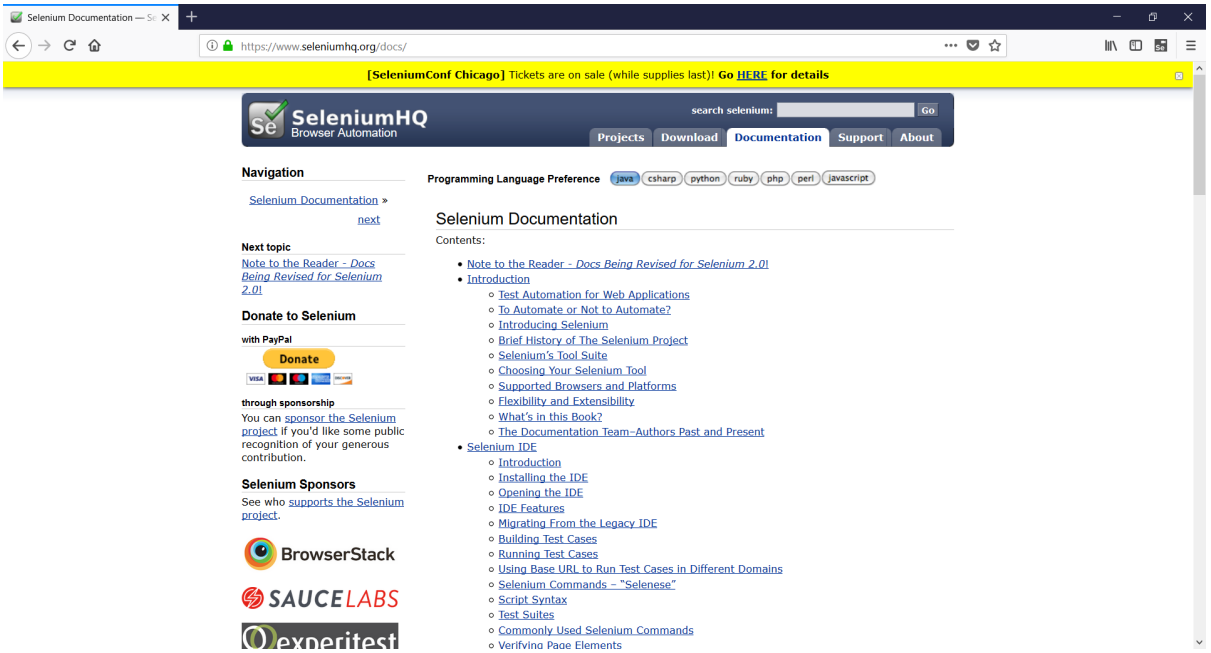
OUTPUT:

i]Assert title:

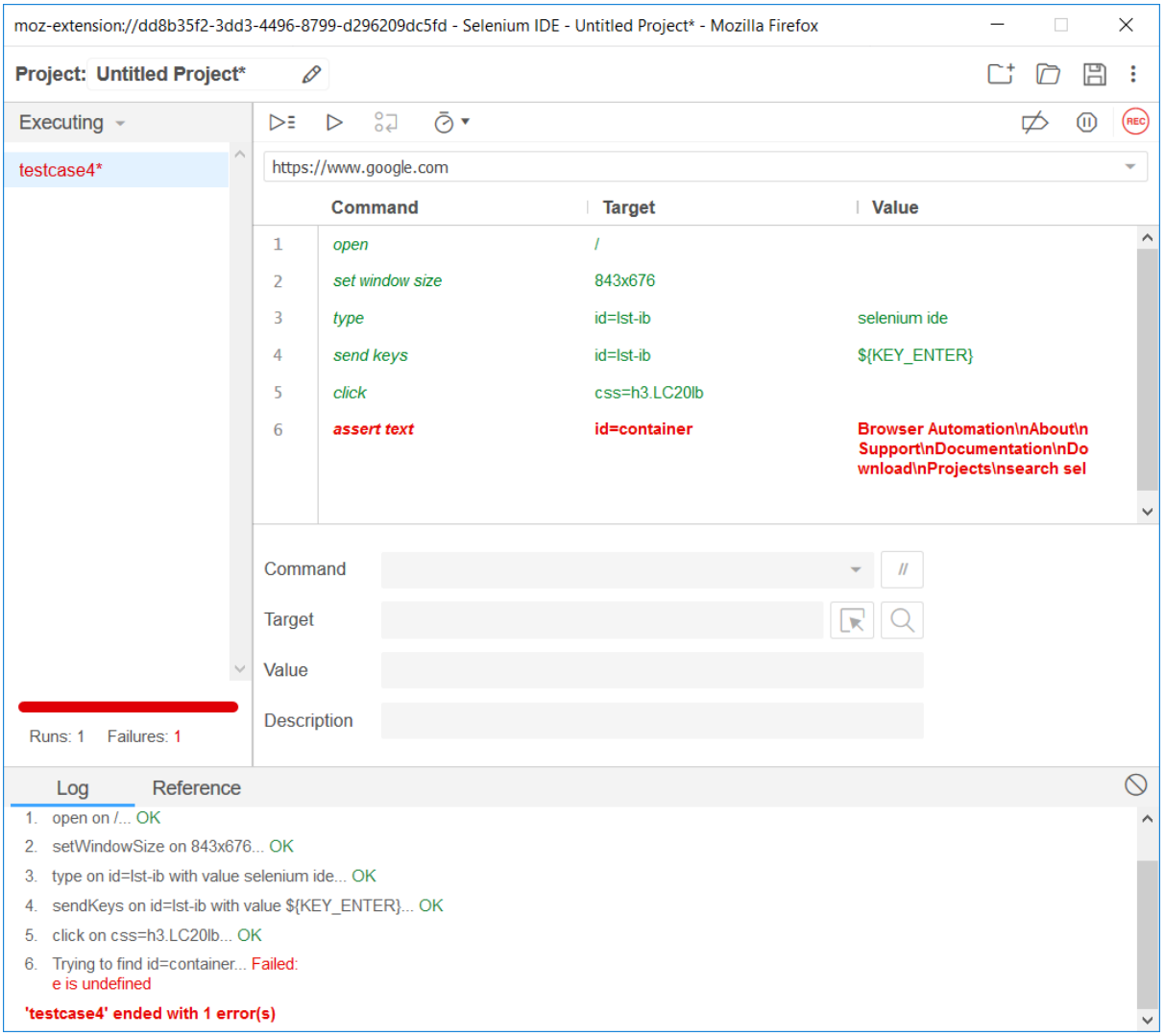

Downloads
https://www.seleniumhq.org/download/
[SeleniumConf Chicago] Tickets are on sale (while supplies last)! Go HERE for details
SeleniumHQ
Browser Automation
edit this page
search selenium:
Go
Projects
Download
Documentation
Support
About
Selenium Downloads
Previous Releases
Source Code
Maven Information
Donate to Selenium
with PayPal
Donate
through sponsorship
You can sponsor the Selenium project if you'd like some public recognition of your generous contribution.
Selenium Sponsors
See who supports the Selenium project.
BrowserStack
SAUCELABS
experitest
Selenium for Mobile
Downloads
Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and additional information for Maven users (Maven is a popular Java build tool).
Selenium Standalone Server
The Selenium Server is needed in order to run Remote Selenium WebDriver. Selenium 3.X is no longer capable of running Selenium RC directly, rather it does it through emulation and the WebDriverBackedSelenium interface.
Download version 3.14.0
To run Selenium tests exported from the legacy IDE, use the Selenium Html Runner.
To use the Selenium Server in a Grid configuration see the wiki page.
The Internet Explorer Driver Server
This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your $PATH (or %PATH% on Windows) in order for the IE Driver to work as expected.
Download version 3.14.0 for (recommended) 32 bit Windows IE or 64 bit Windows IE
CHANGELOG
Selenium Client & WebDriver Language Bindings
In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language specific client drivers. These languages include both 1.x and 2.x style clients.
While language bindings for other languages exist, these are the core ones that are supported by the main project hosted on GitHub.
Language
Client Version
Release Date
Java 3.14.0 2018-08-02 Download Change log Javadoc
C# 3.14.0 2018-08-02 Download Change log API docs
Ruby 3.14.0 2018-08-03 Download Change log API docs

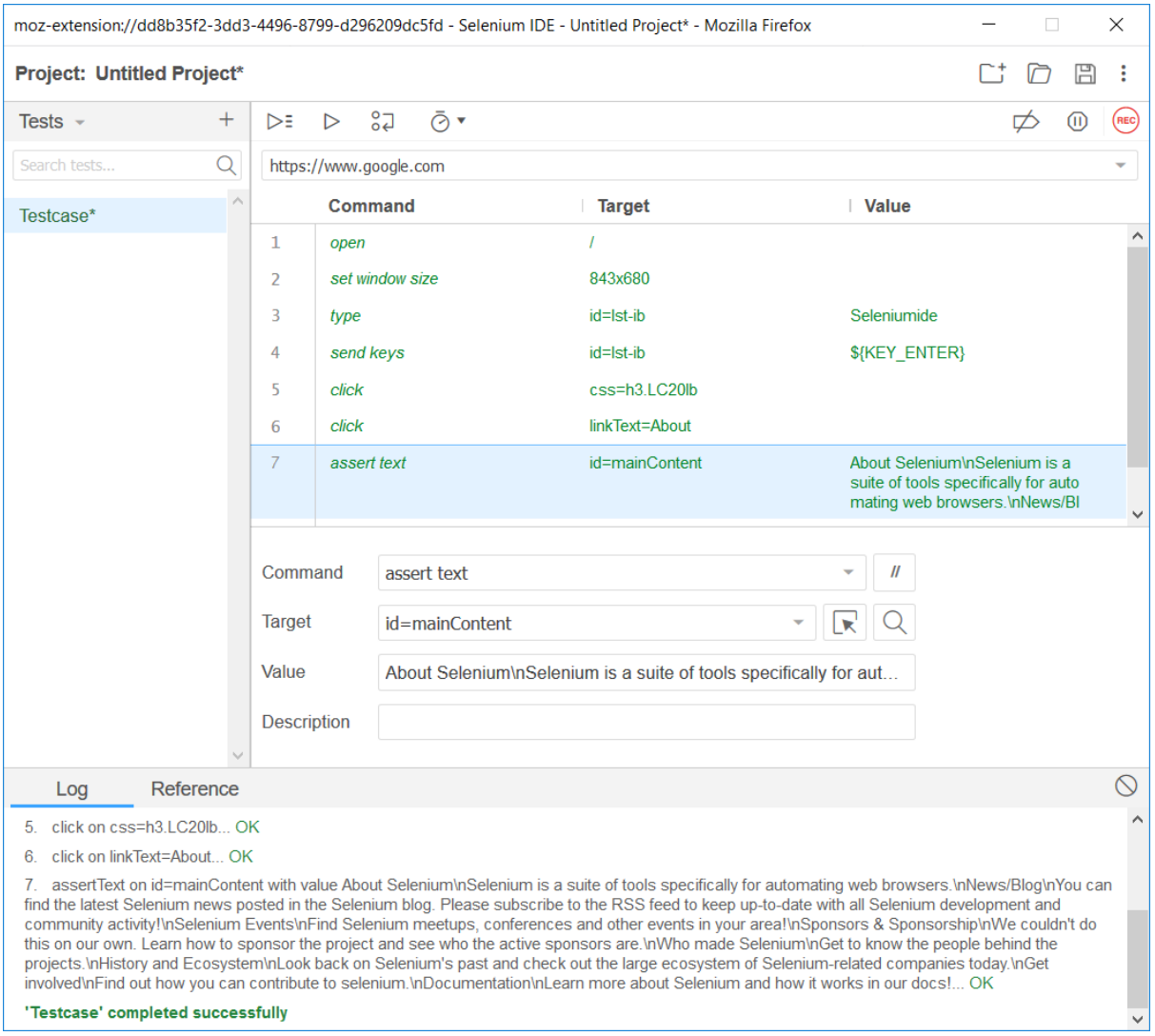

moz-extension://dd8b35f2-3dd3-4496-8799-d296209dc5fd - Selenium IDE - Untitled Project* - Mozilla Firefox
Project: Untitled Project*
Executing
Testcase1*
https://www.google.com
Command
Target
Value
6 send keys id=lst-ib ${KEY_ENTER}
7 click css=.bkWMgd:nth-child(1) .g:nth-child(3) .LC20lb
8 mouse over css=.bkWMgd:nth-child(1) .g:nth-child(3) .LC20lb
9 assert title
10 assert title Downloads
11 close win_ser_local
Command
Target
Value
Description
Runs: 1 Failures: 1
Log
Reference
4. click on id=lst-ib... OK
5. type on id=lst-ib with value selenium ide... OK
6. sendKeys on id=lst-ib with value ${KEY_ENTER}... OK
7. click on css=.bkWMgd:nth-child(1) .g:nth-child(3) .LC20lb... OK
8. mouseOver on css=.bkWMgd:nth-child(1) .g:nth-child(3) .LC20lb... OK
9. assertTitle on ... Failed:
Actual value 'selenium ide - Google Search' did not match ''
'Testcase1' ended with 1 error(s)

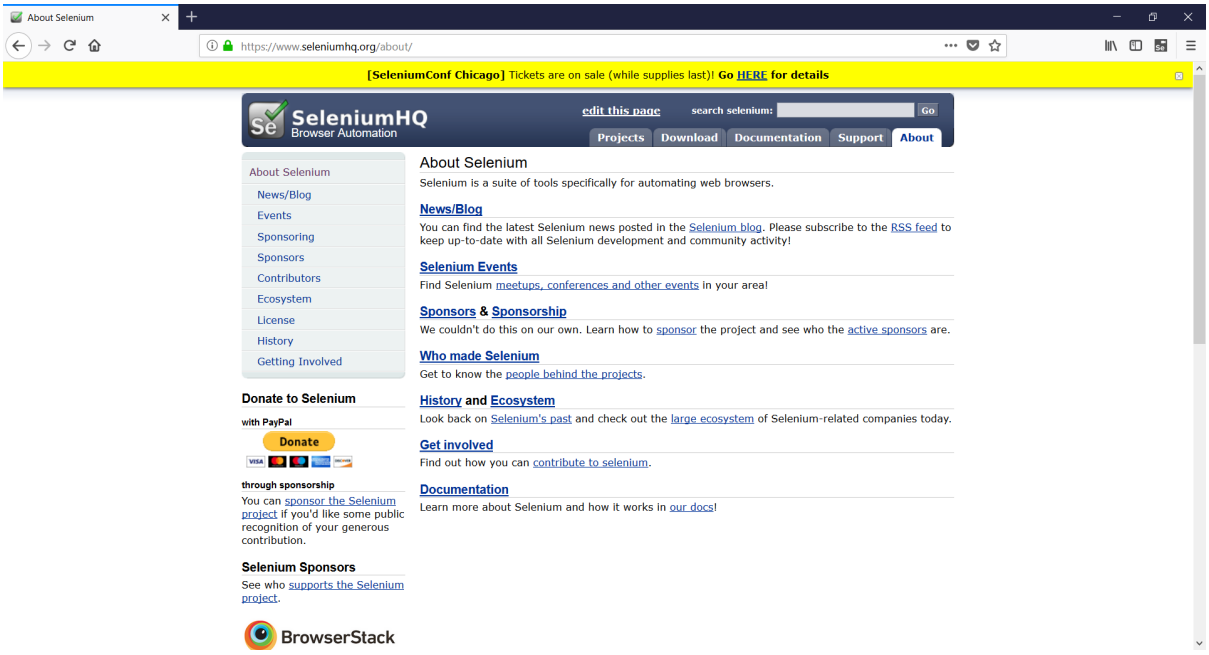

moz-extension://dd8b35f2-3dd3-4496-8799-d296209dc5fd - Selenium IDE - Untitled Project* - Mozilla Firefox
Project: Untitled Project*
Tests
Search tests...
Testcase1*
https://www.google.com
Command
Target
Value
1 open /
2 set window size 843x675
3 type id=lst-ib seleniumide
4 send keys id=lst-ib ${KEY_ENTER}
5 click css=h3.LC20lb
6 click linkText=Documentation
7 assert title Selenium Documentation — Selenium Documentation
Command assert title
Target Selenium Documentation — Selenium Documentation
Value
Description
Log
Reference
2. setWindowSize on 843x675... OK
3. type on id=lst-ib with value seleniumide... OK
4. sendKeys on id=lst-ib with value ${KEY_ENTER}... OK
5. click on css=h3.LC20lb... OK
6. click on linkText=Documentation... OK
7. assertTitle on Selenium Documentation — Selenium Documentation... OK
'Testcase1' completed successfully


https://www.seleniumhq.org/docs/
[SeleniumConf Chicago] Tickets are on sale (while supplies last)! Go HERE for details
SeleniumHQ
Browser Automation
search selenium:
Go
Projects
Download
Documentation
Support
About
Navigation
Selenium Documentation »
next
Next topic
Note to the Reader - Docs Being Revised for Selenium 2.0!
Donate to Selenium
with PayPal
Donate
through sponsorship
You can sponsor the Selenium project if you'd like some public recognition of your generous contribution.
Selenium Sponsors
See who supports the Selenium project.
BrowserStack
SAUCELABS
Programming Language Preference
java
csharp
python
ruby
php
perl
javascript
Selenium Documentation
Contents:
Note to the Reader - Docs Being Revised for Selenium 2.0!
Introduction
Test Automation for Web Applications
To Automate or Not to Automate?
Introducing Selenium
Brief History of The Selenium Project
Selenium's Tool Suite
Choosing Your Selenium Tool
Supported Browsers and Platforms
Flexibility and Extensibility
What's in this Book?
The Documentation Team-Authors Past and Present
Selenium IDE
Introduction
Installing the IDE
Opening the IDE
IDE Features
Migrating From the Legacy IDE
Building Test Cases
Running Test Cases
Using Base URL to Run Test Cases in Different Domains
Selenium Commands – "Selenese"
Script Syntax
Test Suites
Commonly Used Selenium Commands
Verifying Page Elements

ii] Assert Text:




moz-extension://dd8b35f2-3dd3-4496-8799-d296209dc5fd - Selenium IDE - Untitled Project* - Mozilla Firefox
Project: Untitled Project*
Tests
Search tests...
Testcase*
https://www.google.com
Command
Target
Value
1 open /
2 set window size 843x680
3 type id=lst-ib Seleniumide
4 send keys id=lst-ib ${KEY_ENTER}
5 click css=h3.LC20lb
6 click linkText=About
7 assert text id=mainContent About Selenium\nSelenium is a suite of tools specifically for auto mating web browsers.\nNews/Bl
Command assert text
Target id=mainContent
Value About Selenium\nSelenium is a suite of tools specifically for aut...
Description
Log
Reference
5. click on css=h3.LC20lb... OK
6. click on linkText=About... OK
7. assertText on id=mainContent with value About Selenium\nSelenium is a suite of tools specifically for automating web browsers.\nNews/Blog\nYou can find the latest Selenium news posted in the Selenium blog. Please subscribe to the RSS feed to keep up-to-date with all Selenium development and community activity!\nSelenium Events\nFind Selenium meetups, conferences and other events in your area!\nSponsors & Sponsorship\nWe couldn't do this on our own. Learn how to sponsor the project and see who the active sponsors are.\nWho made Selenium\nGet to know the people behind the projects.\nHistory and Ecosystem\nLook back on Selenium's past and check out the large ecosystem of Selenium-related companies today.\nGet involved\nFind out how you can contribute to selenium.\nDocumentation\nLearn more about Selenium and how it works in our docs!... OK
'Testcase' completed successfully


About Selenium
https://www.seleniumhq.org/about/
[SeleniumConf Chicago] Tickets are on sale (while supplies last)! Go HERE for details
SeleniumHQ
Browser Automation
edit this page
search selenium:
Go
Projects
Download
Documentation
Support
About
About Selenium
News/Blog
Events
Sponsoring
Sponsors
Contributors
Ecosystem
License
History
Getting Involved
About Selenium
Selenium is a suite of tools specifically for automating web browsers.
News/Blog
You can find the latest Selenium news posted in the Selenium blog. Please subscribe to the RSS feed to keep up-to-date with all Selenium development and community activity!
Selenium Events
Find Selenium meetups, conferences and other events in your area!
Sponsors & Sponsorship
We couldn't do this on our own. Learn how to sponsor the project and see who the active sponsors are.
Who made Selenium
Get to know the people behind the projects.
History and Ecosystem
Look back on Selenium's past and check out the large ecosystem of Selenium-related companies today.
Get involved
Find out how you can contribute to selenium.
Documentation
Learn more about Selenium and how it works in our docs!
Donate to Selenium
with PayPal
Donate
through sponsorship
You can sponsor the Selenium project if you'd like some public recognition of your generous contribution.
Selenium Sponsors
See who supports the Selenium project.
BrowserStack

Practical no 3

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

Description

Selenium-RC is the solution for tests that need more than simple browser actions and linear execution.

Selenium-RC uses the full power of programming languages to create more complex tests like reading and writing files, querying a database, emailing test results.

You'll want to use Selenium-RC whenever your test requires logic not supported by Selenium-IDE.

What logic could this be? For example, Selenium-IDE does not directly support:
- Condition statements
- Iteration
- Logging and reporting of test results
- Error handling, particularly unexpected errors
- Database testing
- Test case grouping
- Re-execution of failed tests
- Test case dependency
- Screenshot capture of test failures

Although these tasks are not supported by Selenium directly, all of them can be achieved by using programming techniques with a language-specific Selenium-RC client library.

Installation of Selenium RC and Eclipse

Download Eclipse

1. Go to URL – http://www.eclipse.org/downloads/
2. Select **Eclipse IDE for Java Developers** (Click on Windows 32 bit platform)
3. Click on OK button and save to a local drive (i.e. C: or D:,etc)
4. Unzip the downloaded zip file and rename that to Eclipse
5. Create one more folder "Eclipse-Workspace" (i.e. C:Eclipse-Workspace)in the same drive where Eclipse is unzipped and renamed.
6. Create Eclipse desktop shortcut (go to C:Eclipse folder –> right click Eclipse.exe and then click on "desktop create shortcut").

We have finished setting up the eclipse.

Now, we need to download Selenium RC server / client driver and configure that to Eclipse

1. Download Selenium server: http://seleniumhq.org/download/
2. Download Selenium Client driver for Java (from Selenium Client Drivers section)

3. Create “Selenium” folder in C: drive and copy the Selenium-server.jar as well as unzip the Selenium Client driver (C:Selenium)

Downloading and unzipping the files into a folder is done.

We need to configure the appropriate Selenium Client driver Jar file to the Eclipse.

Now, follow the following steps:

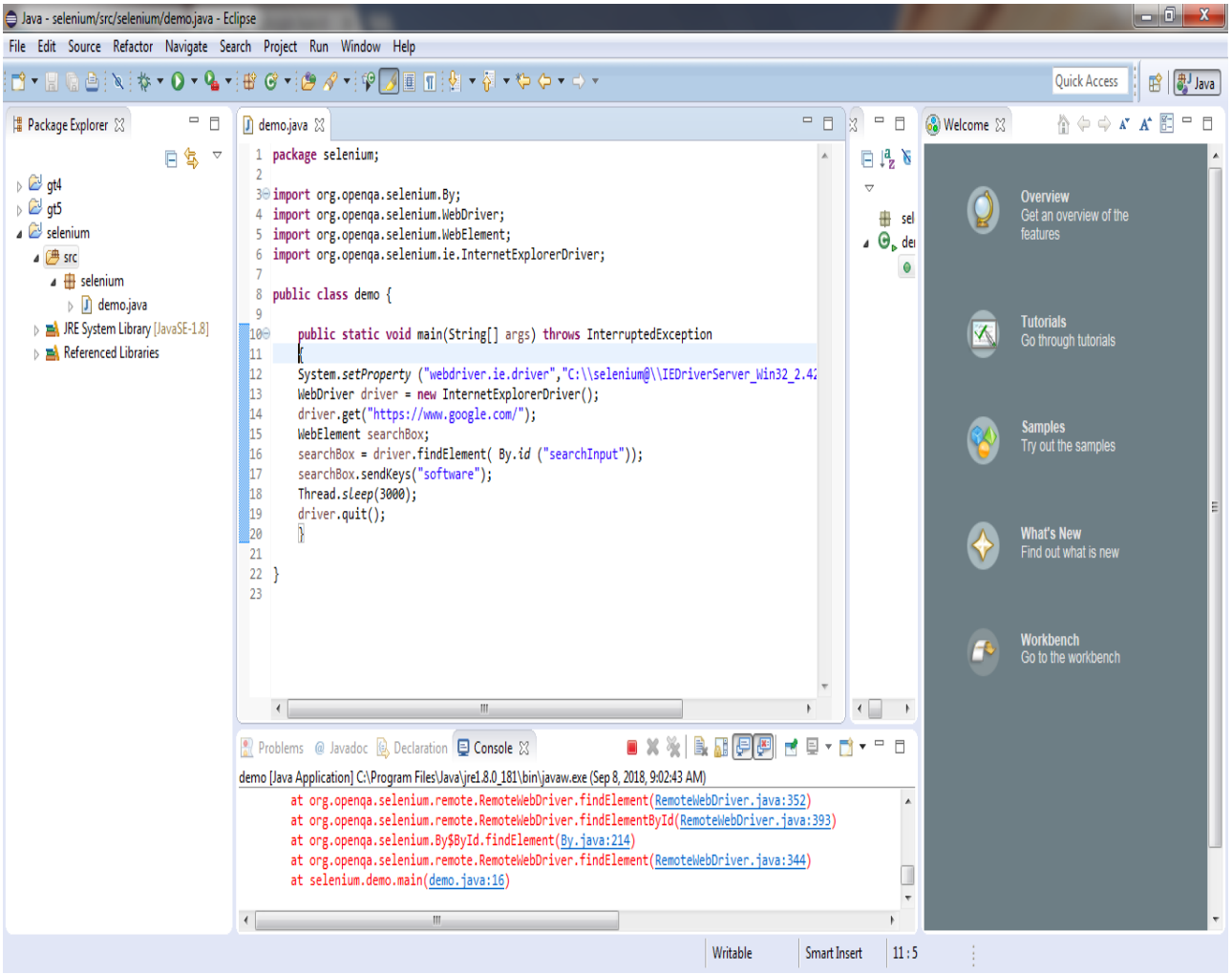
1. Go to Eclipse –> Click File –> New –> Project (from various options need to select just “project”)
2. In Select Wizard –> Click Java –> “Java Project”
3. Give the project name (e.g. DemoTests)
4. Click Finish – Click Yes
5. Now we are done with creation of project and need to configure the Selenium Client driver to this Project
6. Right Click “DemoTests” project
7. Click “Java Build Path”
8. Click Libraries tab
9. Click “Add External JARs” button
10. Select “Selenium Client Drivers” unzipped in C:Selenium folder (Selenium Server JAR file should not be added)
11. Click OK
12.Referenced libraries –> contains both the Selenium Client driver jar files.
13.Create a new class file as “SeleniumDemo” in the “DemoTest” by right click on src folder.
14. Copy the below code in the class file:-

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;

public class hello
{
static String driverPath = "D:\\selenium
pracs\\geckodriver-v0.21.0-win32\\GeckoDriver.exe";
public static WebDriver driver;
public static void main(String args[])
{
int a=10,b=20;
System.out.println("Hi....");
System.out.println(a+b);
System.out.println("Selenium demo.....");

System.setProperty("webdriver.gecko.driver",driverPath);
DesiredCapabilities capabilities = DesiredCapabilities.firefox();
capabilities.setCapability("marionette",true);
driver= new FirefoxDriver(capabilities);
driver.get("https://www.facebook.com/");
driver.manage().window().maximize();
driver.quit();
```

}
}

Now execute the code by right click on the window , then click on run as administrator.

OUTPUT:



Practical no 4

Aim: Arithmetic operation using Junit.

Follow the following steps:

1. Go to Eclipse –> Click File –> New –> Project (from various options need to select just “project”)
2. In Select Wizard –> Click Java –> “Java Project”
3. Give the project name (e.g.Tests)
4. Click Finish – Click Yes
5. Go to Explorer -> Right click on Project name -> Build Path -> Configure Build path -> Libraries -> Add library -> Select Junit -> Click on Junit4 -> Finish.
6. Create a new class file as “Junit” in the “Tests” by right click on src folder.
7. **Write the below code in the class file :-**

```
package test
public class Junit
{
public String Concatenate(String one, String two)
{
  return one + two;
}
public int Multiply( int no1, int no2)
{
  return no1*no2;
}
}
```

8. **Create a Test Runner class to execute above test for Concatenate operation:-**

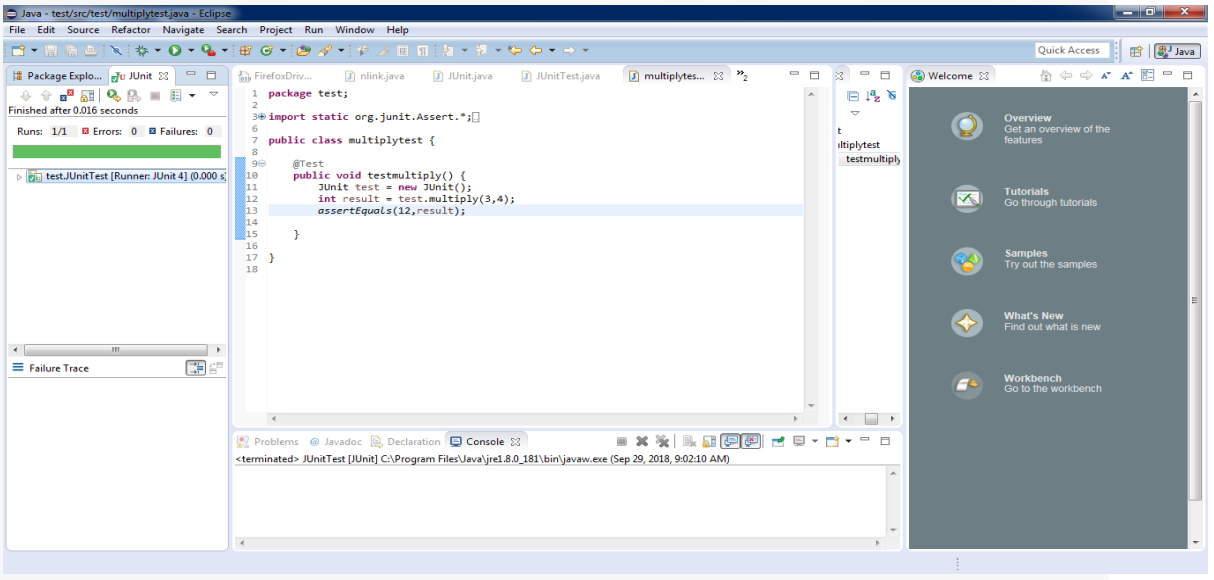
```
package test
import static org.junit.Assert.*;
public class JunitTest
{
public void testConcatenate()
{
Junit test = new Junit();
String result = test.Concatenate(“one”, “two”);
assertEquals (“onetwo”, result);
}
}
```

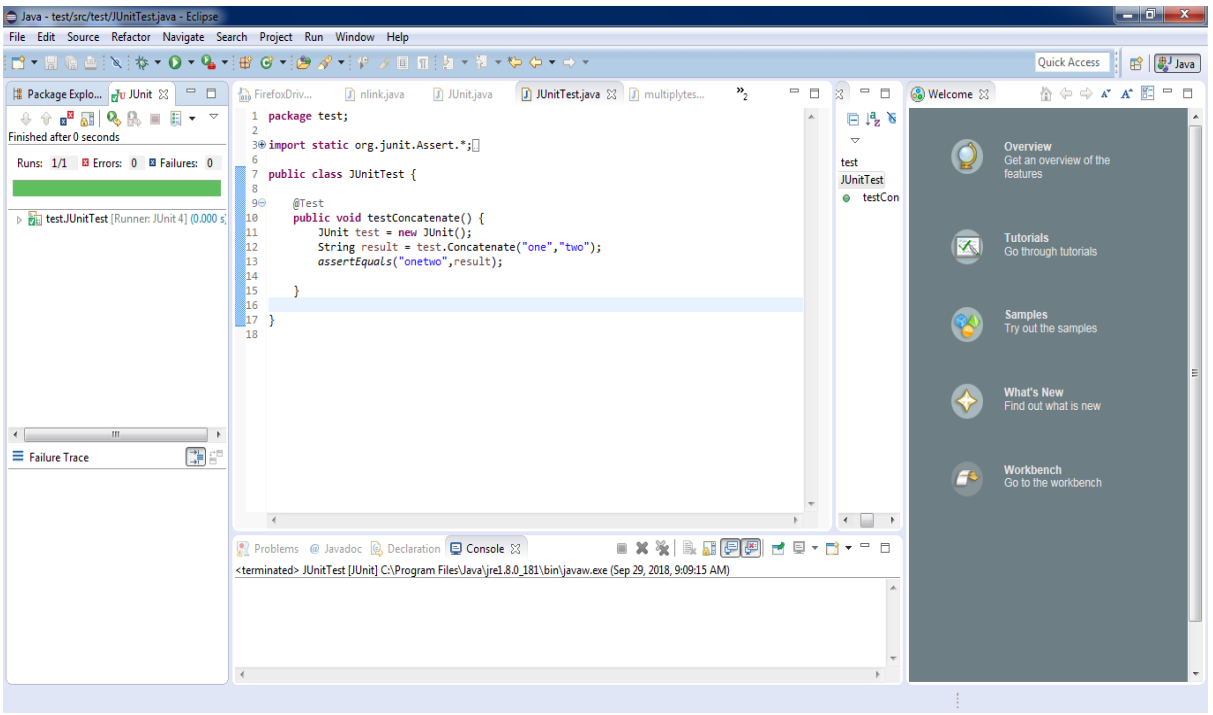
9. **Create a Test Runner class to execute above test for Multiply operation:-**

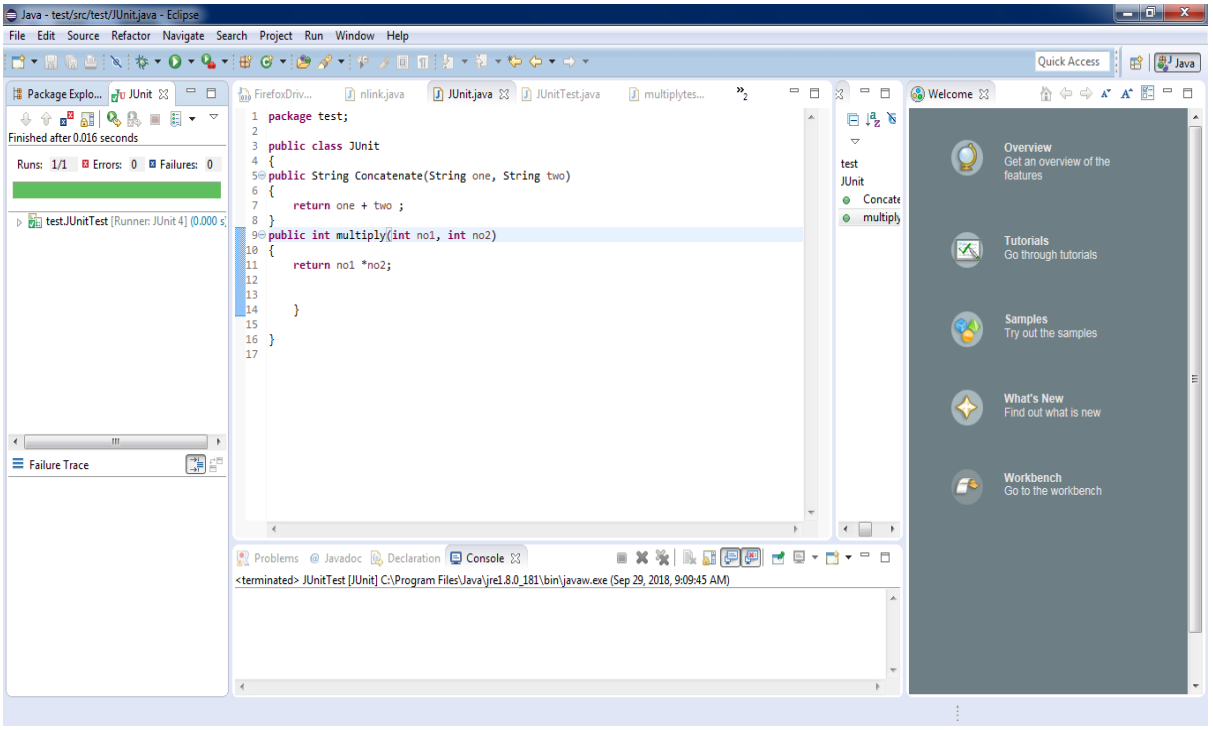
```
package test
import static org.junit.Assert.*;
public class multiplytest
{
public void testmultiply()
{
Junit test = new Junit();
int result = test.multiply(3,4);
assertEquals (12, result);
}
}
```

10. Now execute the code.

OUTPUT:




Java - test/src/test/JUnitTest.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explo... JUnit
Finished after 0 seconds
Runs: 1/1 Errors: 0 Failures: 0
test.JUnitTest [Runner: JUnit 4] (0.000 s)
Failure Trace
FirefoxDriv... nlink.java JUnit.java JUnitTest.java multiplytes...
1 package test;
2
3 import static org.junit.Assert.*;
6
7 public class JUnitTest {
8
9 @Test
10 public void testConcatenate() {
11 JUnit test = new JUnit();
12 String result = test.Concatenate("one","two");
13 assertEquals("onetwo",result);
14
15 }
16
17 }
18
test
JUnitTest
testCon
Problems Javadoc Declaration Console
<terminated> JUnitTest [JUnit] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Sep 29, 2018, 9:09:15 AM)
Welcome
Overview
Get an overview of the features
Tutorials
Go through tutorials
Samples
Try out the samples
What's New
Find out what is new
Workbench
Go to the workbench


Java - test/src/test/JUnit.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explo... JUnit
Finished after 0.016 seconds
Runs: 1/1 Errors: 0 Failures: 0
test.JUnitTest [Runner: JUnit 4] (0.000 s)
Failure Trace
FirefoxDriv... nlink.java JUnit.java JUnitTest.java multiplytes...
1 package test;
2
3 public class JUnit
4 {
5 public String Concatenate(String one, String two)
6 {
7 return one + two ;
8 }
9 public int multiply(int no1, int no2)
10 {
11 return no1 *no2;
12
13
14 }
15
16 }
17
test
JUnit
Concate
multiply
Problems Javadoc Declaration Console
<terminated> JUnitTest [JUnit] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Sep 29, 2018, 9:09:45 AM)
Welcome
Overview
Get an overview of the features
Tutorials
Go through tutorials
Samples
Try out the samples
What's New
Find out what is new
Workbench
Go to the workbench

Practical no 5

Aim: Write and test a program to specific web page. (Using JUnit)

Follow the following steps:

1. Go to Eclipse –> Click File –> New –> Project (from various options need to select just "project")
2. In Select Wizard –> Click Java –> "Java Project"
3. Give the project name (e.g.Junittests)
4. Click Finish – Click Yes
5. Go to Explorer -> Right click on Project name -> Build Path -> Configure Build path -> Libraries -> Add library -> Select Junit -> Click on Junit4 -> Finish.
6. **Create a new class file as "Junit2" in the "Tests" by right click on src folder.**

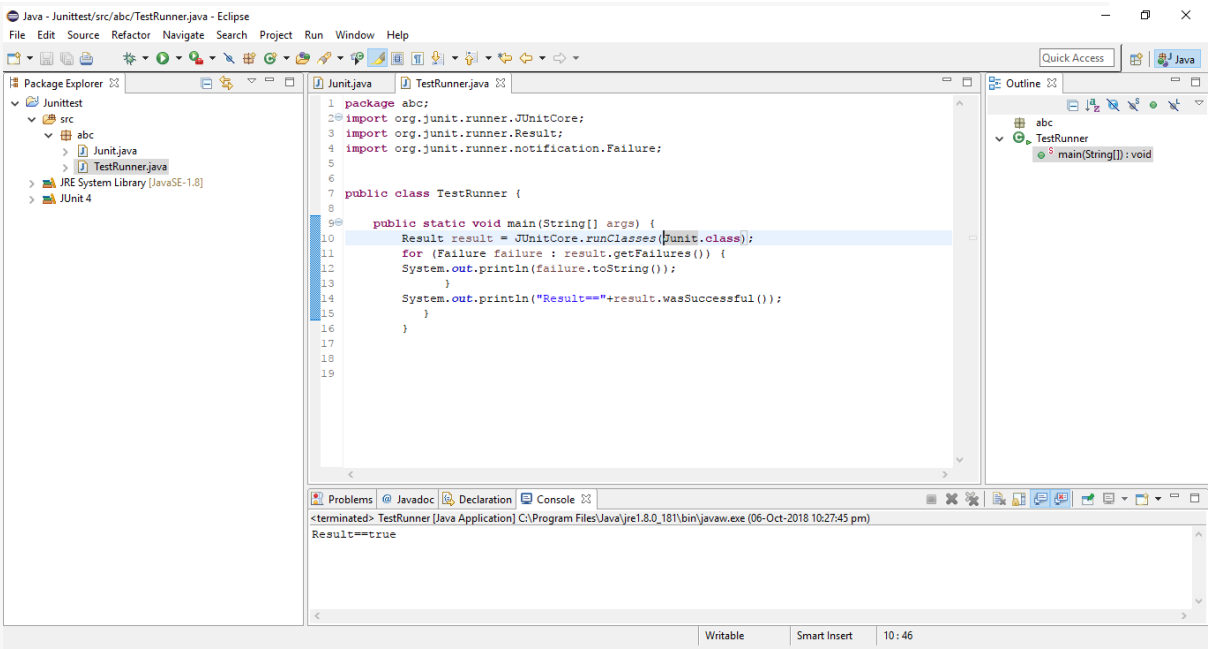
```
import org.junit.Test;
import static org.junit.Assert.assertEquals;
public class TestJunit {
  @Test
public void testSetup() {
    String str= "I am done with Junit setup";
assertEquals("I am done with Junit setup",str);
  }
}
```

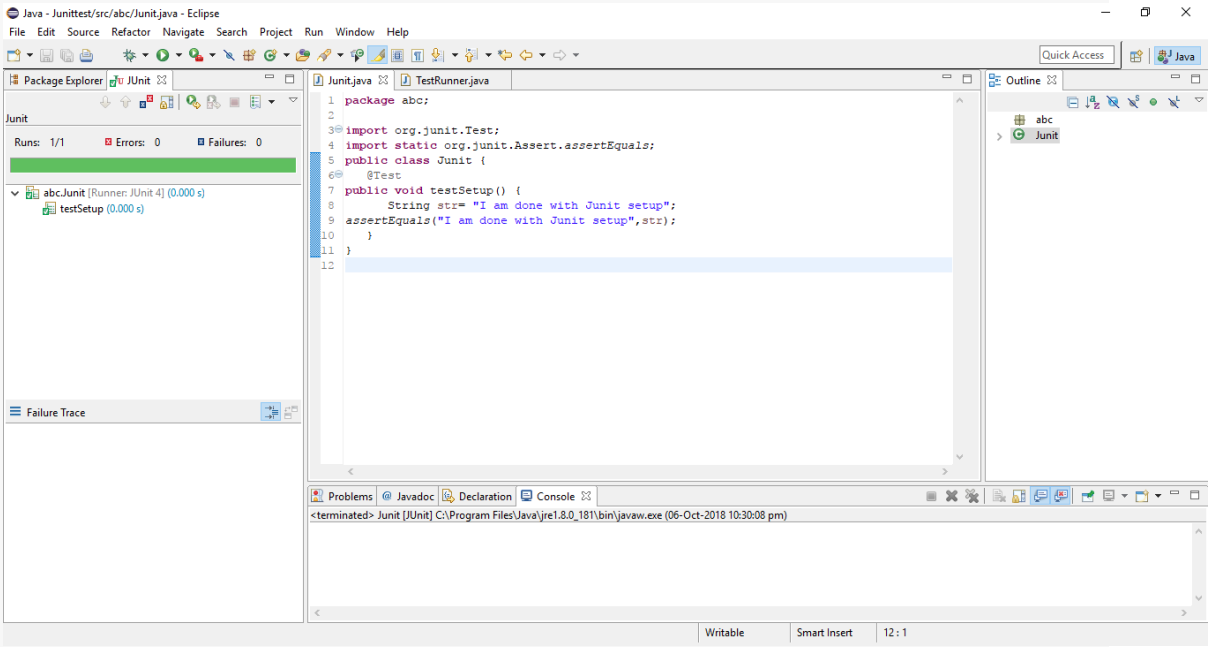
7. **Create a Test Runner class to execute above test.**

```
import org.junit.runner.JUnitCore;
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestRunner {
public static void main(String[] args) {
    Result result = JUnitCore.runClasses(TestJunit.class);
for (Failure failure : result.getFailures()) {
System.out.println(failure.toString());
    }
System.out.println("Result=="+result.wasSuccessful());
  }
}
```

8. Execute the code.

OUTPUT:


Java - Junittest/src/abc/TestRunner.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
Junittest
src
abc
Junit.java
TestRunner.java
JRE System Library [JavaSE-1.8]
JUnit 4
Junit.java
TestRunner.java
1 package abc;
2 import org.junit.runner.JUnitCore;
3 import org.junit.runner.Result;
4 import org.junit.runner.notification.Failure;
5
6
7 public class TestRunner {
8
9 public static void main(String[] args) {
10 Result result = JUnitCore.runClasses(Junit.class);
11 for (Failure failure : result.getFailures()) {
12 System.out.println(failure.toString());
13 }
14 System.out.println("Result=="+result.wasSuccessful());
15 }
16 }
17
18
19
Outline
abc
TestRunner
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> TestRunner [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (06-Oct-2018 10:27:45 pm)
Result==true
Writable
Smart Insert
10 : 46


Java - Junittest/src/abc/Junit.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
JUnit
Junit
Runs: 1/1
Errors: 0
Failures: 0
abc.Junit [Runner: JUnit 4] (0.000 s)
testSetup (0.000 s)
Failure Trace
Junit.java
TestRunner.java
1 package abc;
2
3 import org.junit.Test;
4 import static org.junit.Assert.assertEquals;
5 public class Junit {
6 @Test
7 public void testSetup() {
8 String str= "I am done with Junit setup";
9 assertEquals("I am done with Junit setup",str);
10 }
11 }
12
Outline
abc
Junit
Problems
Javadoc
Declaration
Console
<terminated> Junit [JUnit] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (06-Oct-2018 10:30:08 pm)
Writable
Smart Insert
12 : 1

Practical no 6

Aim: Write and test a program to update 10 student records into table into Excel file

Follow the following steps:

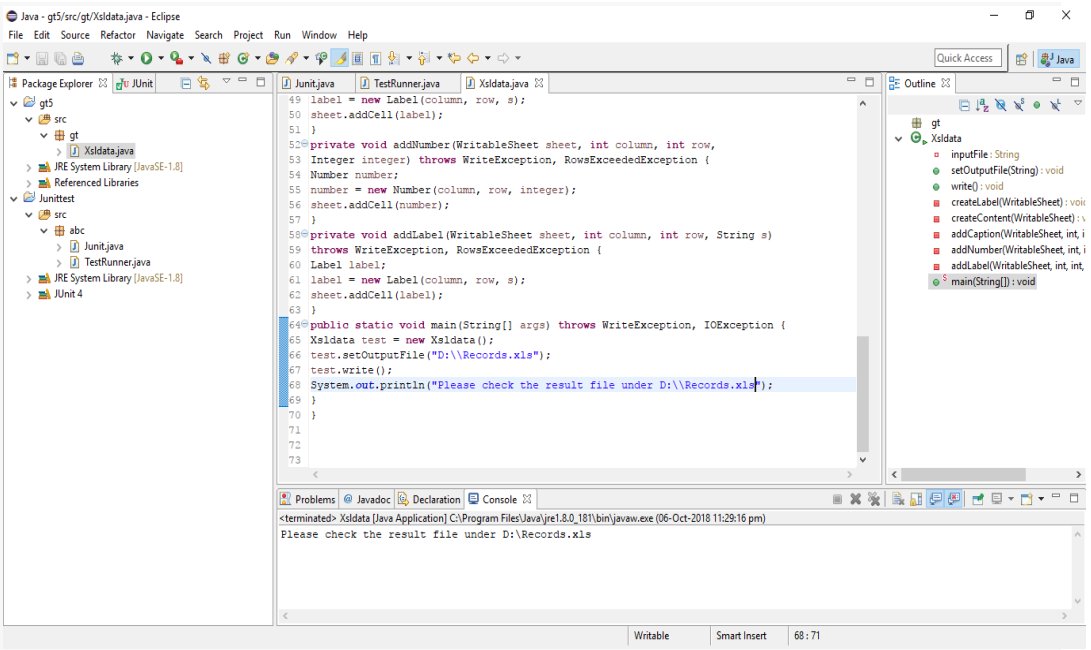
1. Go to Eclipse –> Click File –> New –> Project (from various options need to select just "project")
2. In Select Wizard –> Click Java –> "Java Project"
3. Give the project name (e.g. gt5)
4. Click Finish – Click Yes
5. Right Click "gt5" project
6. Click "Java Build Path"
7. Click Libraries tab
8. Click "Add External JARs" button
9. Add two external jar files: jxl.2.6.jar file & selenium-server-standalone-3.1.0
10. Click OK
11. Create a new class file as "Xsldata" in the "gt5" by right click on src folder.
12. Write the code-

```
package gt;
import java.io.File;
import java.io.IOException;
import jxl.Workbook;
import jxl.write.Label;
import jxl.write.Number;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import jxl.write.WriteException;
import jxl.write.biff.RowsExceededException;
public class Xsldata {
private String inputFile;
public void setOutputFile(String inputFile) {
this.inputFile = inputFile;
}
public void write() throws IOException, WriteException {
File file = new File(inputFile);
WritableWorkbook workbook = Workbook.createWorkbook(file);
workbook.createSheet("Report", 0);
WritableSheet excelSheet = workbook.getSheet(0);
createLabel(excelSheet);
createContent(excelSheet);
workbook.write();
workbook.close();
}
private void createLabel(WritableSheet sheet)
throws WriteException {
```

```
addCaption(sheet, 0, 0, "Student Name");
addCaption(sheet, 1, 0, "Subject 1");
addCaption(sheet, 2, 0, "subject 2");
addCaption(sheet, 3, 0, "subject 3");
addCaption(sheet, 4, 0, "Total");
}
private void createContent(WritableSheet sheet) throws WriteException,
RowsExceededException {
for (int i = 1; i < 10; i++) {
addLabel(sheet, 0, i, "Student " + i);
addNumber(sheet, 1, i, ((i*i)+17));
addNumber(sheet, 2, i, ((i*i)+14));
addNumber(sheet, 3, i, ((i*i)+13));
int total;
total=3*(i*i)+17+14+13;
addNumber(sheet,4,i,total);
}
}
private void addCaption(WritableSheet sheet, int column, int row, String s)
throws RowsExceededException, WriteException {
Label label;
label = new Label(column, row, s);
sheet.addCell(label);
}
private void addNumber(WritableSheet sheet, int column, int row,
Integer integer) throws WriteException, RowsExceededException {
Number number;
number = new Number(column, row, integer);
sheet.addCell(number);
}
private void addLabel(WritableSheet sheet, int column, int row, String s)
throws WriteException, RowsExceededException {
Label label;
label = new Label(column, row, s);
sheet.addCell(label);
}
public static void main(String[] args) throws WriteException, IOException {
Xsldata test = new Xsldata();
test.setOutputFile("F://Jayshree//sample.xls");
test.write();
System.out.println("Please check the result file under F://Jayshree//sample.xls");
}
}
```

13. Create a file in a folder in any drive for showing the excel sheet(output).

OUTPUT:


Java - gt5/src/gt/Xsldata.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
JUnit
gt5
src
gt
Xsldata.java
JRE System Library [JavaSE-1.8]
Referenced Libraries
Junittest
src
abc
Junit.java
TestRunner.java
JRE System Library [JavaSE-1.8]
JUnit 4
Junit.java
TestRunner.java
Xsldata.java
label = new Label(column, row, s);
sheet.addCell(label);
}
private void addNumber(WritableSheet sheet, int column, int row,
Integer integer) throws WriteException, RowsExceededException {
Number number;
number = new Number(column, row, integer);
sheet.addCell(number);
}
private void addLabel(WritableSheet sheet, int column, int row, String s)
throws WriteException, RowsExceededException {
Label label;
label = new Label(column, row, s);
sheet.addCell(label);
}
public static void main(String[] args) throws WriteException, IOException {
Xsldata test = new Xsldata();
test.setOutputFile("D:\\Records.xls");
test.write();
System.out.println("Please check the result file under D:\\Records.xls");
}
}
Outline
gt
Xsldata
inputFile : String
setOutputFile(String) : void
write() : void
createLabel(WritableSheet) : voi
createContent(WritableSheet) :
addCaption(WritableSheet, int, i
addNumber(WritableSheet, int, i
addLabel(WritableSheet, int, int,
main(String[]) : void
Problems
Javadoc
Declaration
Console
<terminated> Xsldata [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (06-Oct-2018 11:29:16 pm)
Please check the result file under D:\Records.xls
Writable
Smart Insert
68 : 71

Practical no 7

Aim: Write and test a program to select the number of students whose total is more than 100.

Follow the following steps:

1. Go to Eclipse –> Click File –> New –> Project (from various options need to select just “project”)
2. In Select Wizard –> Click Java –> “Java Project”
3. Give the project name (e.g. gt5)
4. Click Finish – Click Yes
5. Right Click “gt5” project
6. Click “Java Build Path”
7. Click Libraries tab
8. Click “Add External JARs” button
9. Add two external jar files: jxl.2.6.jar file & selenium-server-standalone-3.1.0
10. Click OK
11. Create a new class file as “Xsldata” in the “gt5” by right click on src folder.

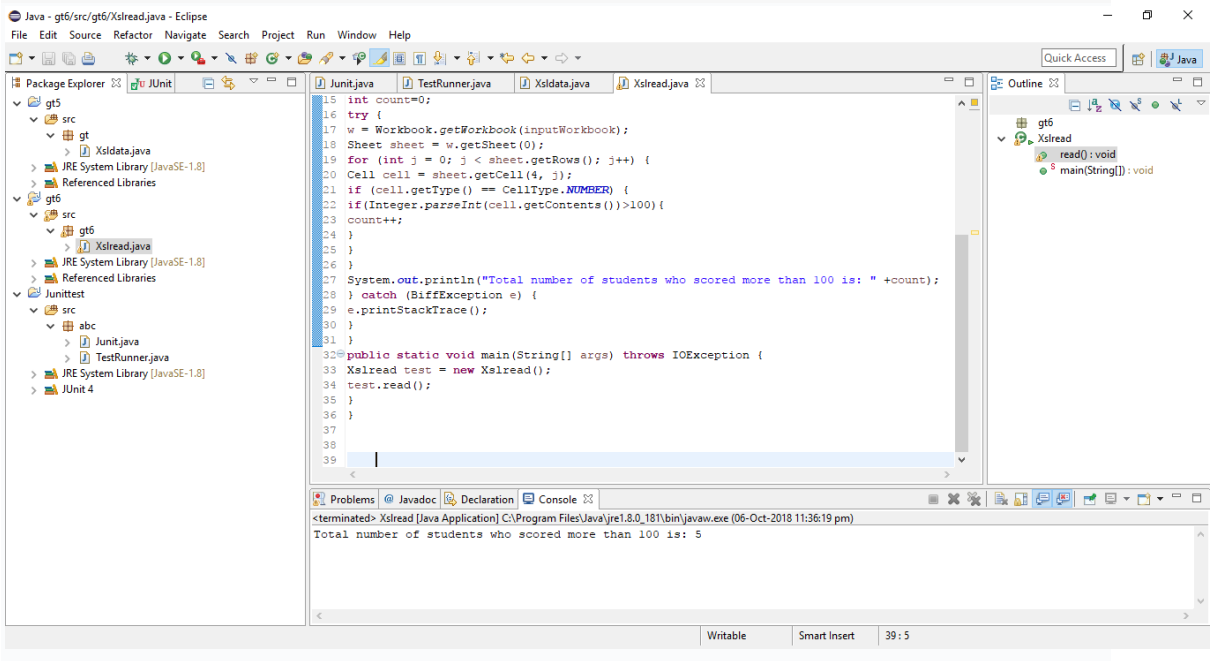
Write the code-

```
package gt;
import java.io.File;
import java.io.IOException;
import jxl.Cell;
import jxl.CellType;
import jxl.Sheet;
import jxl.Workbook;
import jxl.read.biff.BiffException;
public class Xslread {
public void read() throws IOException {
File inputWorkbook = new File("f:/Jayshree/Sample.xls");
Workbook w;
boolean flag=false;
int count=0;
try {
w = Workbook.getWorkbook(inputWorkbook);
Sheet sheet = w.getSheet(0);
for (int j = 0; j < sheet.getRows(); j++) {
Cell cell = sheet.getCell(4, j);
if (cell.getType() == CellType.NUMBER) {
if(Integer.parseInt(cell.getContents())>100){
count++;
}
}
}
System.out.println("Total number of students who scored more than 100 is: " +count);
} catch (BiffException e) {
e.printStackTrace();
}
}
```

```
public static void main(String[] args) throws IOException {
Xslread test = new Xslread();
test.read();
}
}
```

12. Create a file in a folder in any drive for showing the excel sheet(output).

OUTPUT:



Practical no 8

Aim: Write and test a program to provide total number of objects present / available on the page.

Documentation :

Using the web resource –
https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/WebElement.html

The following methods are used :

1)Click()

2)findElement(By by)

3)findElements(By by)

4)getAttribute(java.lang.String name)

5)getCssValue(java.lang.String propertyname)

6)getTagName()

7)getText()

8)sendKeys(java.lang.CharSequence... keysToSend)

OC.java :

```
package gt;
import org.openqa.selenium.By;
import org.openqa.selenium.Platform;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
public class OC {
static String driverPath = "C:\\xampp\\htdocs\\sw\\node-test\\GeckoDriver.exe";
public static WebDriver driver;
public static void main(String args[])
{
System.setProperty("webdriver.gecko.driver",driverPath);
DesiredCapabilities capabilities = new DesiredCapabilities();
capabilities = DesiredCapabilities.firefox();
capabilities.setBrowserName("firefox");
capabilities.setVersion("38.0.5");
capabilities.setPlatform(Platform.WINDOWS);
capabilities.setCapability("marionette", false);
driver= new FirefoxDriver(capabilities);
driver.get("http://toolsqa.wpengine.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:

```
Total links are246
Link 0 Link name
Link 1 Link name
Link 2 Link name
Link 3 Link name
Link 4 Link name
Link 5 Link name
Link 6 Link name
Link 7 Link name
Link 8 Link name HOME
Link 9 Link name TUTORIAL
.
.
.
Link 243 Link name
Link 244 Link name
Link 245 Link name
BUILD SUCCESSFUL (total time: 1 minute 10 seconds)
```

Practical no 9

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

Combo.html:

```
<html>
<body>
<select>
<option>Maharashtra</option>
<option>Gujarat</option>
<option>Rajasthan</option>
<option>Madhya Pradesh</option>
</select>
</body>
</html>
```

ListCount.java :

```
package gt;
import org.openqa.selenium.By;
import org.openqa.selenium.Platform;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import com.thoughtworks.selenium.*;
/**
*
* @author Jayasree.Ravi
*/
public class ListCount extends SeleneseTestCase{
static String driverPath = "C:\\xampp\\htdocs\\sw\\node-test\\GeckoDriver.exe";
public static WebDriver driver;
public static void main(String args[])
{
System.setProperty("webdriver.gecko.driver",driverPath);
DesiredCapabilities capabilities = new DesiredCapabilities();
capabilities = DesiredCapabilities.firefox();
capabilities.setBrowserName("firefox");
capabilities.setVersion("38.0.5");
capabilities.setPlatform(Platform.WINDOWS);
capabilities.setCapability("marionette", false);
driver= new FirefoxDriver(capabilities);
driver.get("http://localhost:1111/sw//combo.html");
```

```
java.util.List<WebElement> optionCount = driver.findElements(By.xpath("//select/option"));
System.out.println("Number of Options in the Combo Box is "+ optionCount.size());
for (int i = 0; i<optionCount.size(); i=i+1)
{
System.out.println(optionCount.get(i).getText());
}
}
}
```

OUTPUT:

```
Number of Options in the Combo Box is 4
Maharashtra
Gujarat
Rajasthan
Madhya Pradesh
BUILD SUCCESSFUL (total time: 44 seconds)
```

Practical no 10

Aim: Write and test a program to count the number of items present on a desktop.

1)Write this java test code in eclipse. Since we are not using any Selenium object here no need of selenium RC Server to run at the command prompt for this program.

2)Write a visual basic script with the following code and save as count.vbs under the desktop.

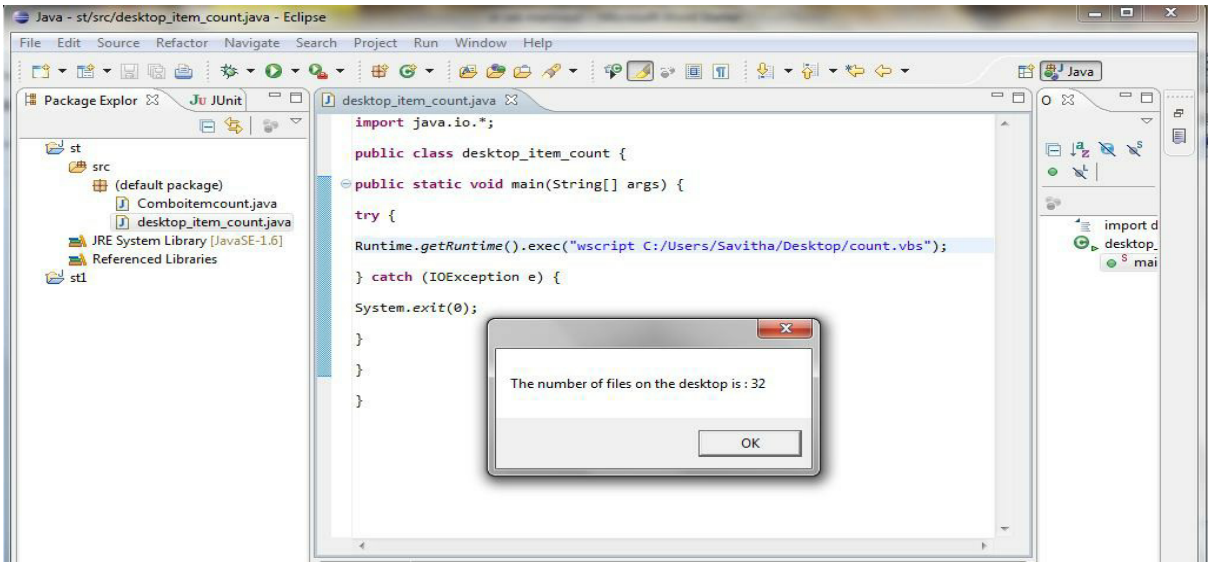
```
Set fso = createobject("Scripting.FileSystemObject")

DesktopPath = CreateObject("WScript.Shell").SpecialFolders("Desktop") 'Files count

 msgbox "The number of files on the desktop is : "
&fso.GetFolder(DesktopPath).Files.Count 'Folders count

msgbox "The number of folders on the desktop is : "
&fso.GetFolder(DesktopPath).SubFolders.Count
```

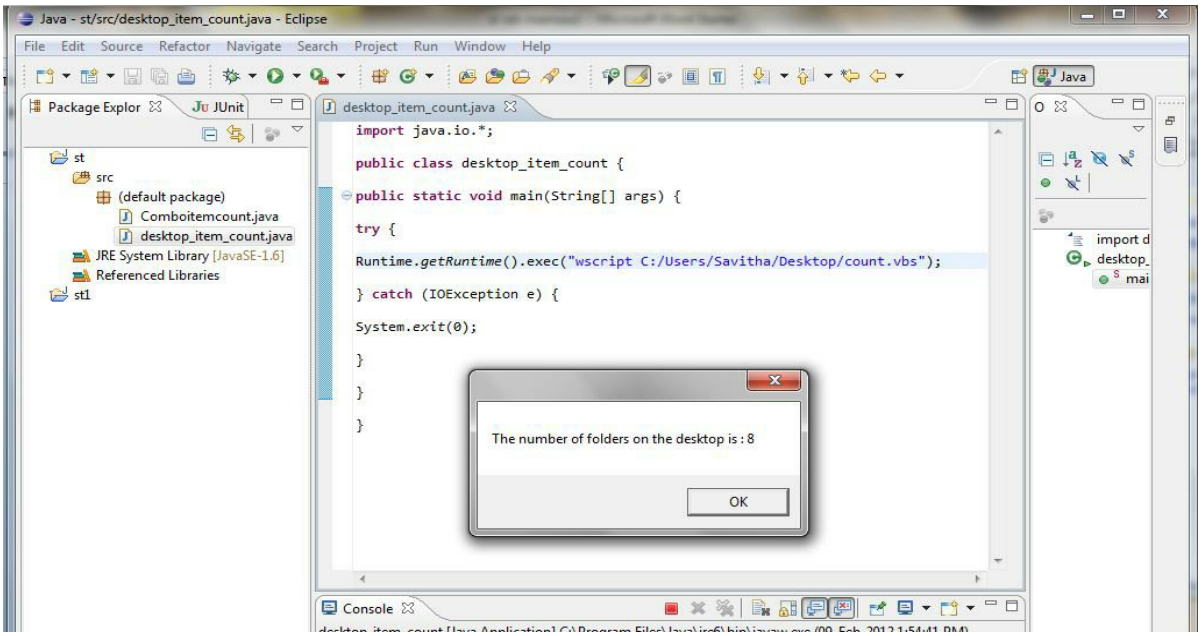
· In eclipse write the following code.

```
 import java.io.*;

 public class desktop_item_count

{

 public static void main(String[] args)

{

 Try

 {

Runtime.getRuntime().exec("wscript C:/Users/Savitha/Desktop/count.vbs");

 }

 catch (IOException e)

{

System.exit(0);

 }

 }

 }
```

Note: Don’t forget to change the desktop file path for your system

OUTPUT:


Java - st/src/desktop_item_count.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explor
JUnit
st
src
(default package)
Comboitemcount.java
desktop_item_count.java
JRE System Library [JavaSE-1.6]
Referenced Libraries
st1
desktop_item_count.java
import java.io.*;
public class desktop_item_count {
public static void main(String[] args) {
try {
Runtime.getRuntime().exec("wscript C:/Users/Savitha/Desktop/count.vbs");
} catch (IOException e) {
System.exit(0);
}
}
}
The number of files on the desktop is : 32
OK


Java - st/src/desktop_item_count.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explor
JUnit
st
src
(default package)
Comboitemcount.java
desktop_item_count.java
JRE System Library [JavaSE-1.6]
Referenced Libraries
st1
desktop_item_count.java
import java.io.*;
public class desktop_item_count {
public static void main(String[] args) {
try {
Runtime.getRuntime().exec("wscript C:/Users/Savitha/Desktop/count.vbs");
} catch (IOException e) {
System.exit(0);
}
}
}
The number of folders on the desktop is : 8
OK
Console