International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

Versatile Testing System using Selenium

Siva Kumar Kotamraju, Pavani Valeti, Silpa Chaitanya Prathipati ,

Abstract*: *Testing is exercising or evaluating either a system or system components by Setup Testing is exercising or evaluating either a system or system components by manual means or automated means to verify specific requirements. Many Organizations or companies follow different approaches in verifying applications effectively. Automated testing is most preferred for more interactive and responsive software projects by employing agile methodology in the organizations. Automation is the process to run repeatable tests by utilizing tool for an application testing. Selenium is a group of tools (Selenium IDE, Selenium RC, Selenium Web Driver), each one with a new approach to support test automation

Index Terms*: *Test Automation, Selenium, Automated Testing

I. INTRODUCTION

Testing helps to ensure application meets the originating requirements and expectations after rigorous testing of an application .Testing provides ability ensure that the application when used in unintended ways also function correctly. The 1:10:100 testing rule states that longer it takes to identify the defect the cost of fixing a defect increases exponentially.

II. AUTOMATED TESTING

Types of Testing:

1. Manual Testing: A Tester carries out Manual Testing who performs actions on tested application manually and compares result each step to results specified in the requirements document
2. Automated Testing: Using software tools for test execution

The Benefits of Automated Testing:

- Test execution is significantly faster when automated than what can be accomplished by a human tester manually.
- Execution of test is consistent and can be repeatedly under the same set of conditions eliminating the possibility of error being made by a tester being marked as a defect. This also makes it easier to determine when a defect has been fixed in an application.
- Test scripts can be reused, with little or no modifications, to test features in later versions of the same software application.
- Automated Testing tools provide detailed results of the test execution process, which can help the developer, identify any problem quicker.
- Allows better usage of tester time and resources. Automated tests would allow, for instance, a tester to create test scripts and manage defects during the day while running the automated test unattended overnight.
- Eases the task of comprehensively testing the application by allowing multiple scenarios of the same feature to be easily tested.
- User programming extensions to extract important information from the application and the computer system that may not be easily available when using manual testing approach.
- Follows a well documented approach that allows all defects in an application to be tracked from discovery through repair.

III. MANUAL OR AUTOMATED

Cost

Tools do not perform manual testing but automated testing requires the purchase of an automated testing tool and training of staff to use tool.

Software Environment

The environment in which the application was developed is also a factor. If there is no automated testing tool, how feasible is it to build a test harness to test the application?

Testing Type

Testing for such things as ease of use of an application cannot be done using an automated testing tool and is best done manually, preferably by usability experts.

IV. MANUAL TESTING PROCESS

1. Plan Test: Plan Test will be written by converting the software requirements for the application into test cases
2. Perform Test: Perform each test represented by a test case by performing the user actions. It is important to repeat each step sequentially as described by the test case.
3. Verify Test: Each step with an expected result should be marked as either passed or failed. The existence of a single failed step marks the entire test as failed
4. Track defects: For each test that is failed, a defect should be created. This defect should remain open until developers fix the defect and the test is rerun to verify these.

Revised Manuscript Received on 22 May 2019.

* Correspondence Author

Siva Kumar Kotamraju*, Department of Computer Science & Engineering, Vignan's Nirula Institute of Technology & science for Women, Pedapalakaluru, Guntur

Pavani Valeti, Department of Computer Science & Engineering, Vignan's Nirula Institute of Technology & science for Women, Pedapalakaluru, Guntur

Silpa Chaitanya Prathipati, Department of Computer Science & Engineering, Vignan's Nirula Institute of Technology & science for Women, Pedapalakaluru, Guntur

V. AUTOMATED TESTING PROCESS

1. Plan Test: This is the same test planning process done in the manual approach to testing. The end result of this approach is the creation of test cases.
2. Create Automated Test: This involves automating the steps presented in the test case. For most test automation tools, this process typically involves:
   a) Record the actions carried out by the tester and convert actions into a macro scripting language
   b) Enhancing the test script to include programmatic verification steps and including additionally useful code script
3. Debug Test: As with every program, test the created test script functionality to ensure that it meets the original intent. A test that is not fully debugged could generate false positive by highlighting correct application functionality as being defective.
4. Run Test: Once verification of test is over, run the test by using a scheduling tool or runt the test directly
5. Verify Results: Unlike the manual testing process, where the results are verified during the execution of a test, the automated testing approach generates comprehensive test results that are stored and can be later viewed to determine the presence of defects.

   A big advantage of automated testing is execution of tests unattended .so all necessary information should be directed to the test result for later viewing by testers.
6. Track Defects: For failed tests, a defect must be created and tracked through development and subsequent retests.

VI. TEST AUTOMATION FOR WEB APPLICATIONS

Test automation has particular advantages to improve the efficiency of a software team's testing processes. Test automation supports:

- More Often regression testing
- Developers will get Rapid feedback
- Many rounds of test case execution
- Supporting extreme development and Agile methodologies
- Documentation of test cases in more disciplined manner
- Reporting defects in a customized fashion
- Defects not found during manual testing

VII. TO AUTOMATE OR NOT TO AUTOMATE?

When should one go for automating test cases? Is automation always useful?

It is **not** useful to automate test cases each time. Sometimes manual testing may be more suitable. For example, if the users interface of an application changes at a later date, then automation needs to be changed. Also, there are times, simply building test automation is not just enough. For short term and for tight deadline applications, at present no test automation available, then manual testing is the best solution.

VIII. INTRODUCING SELENIUM

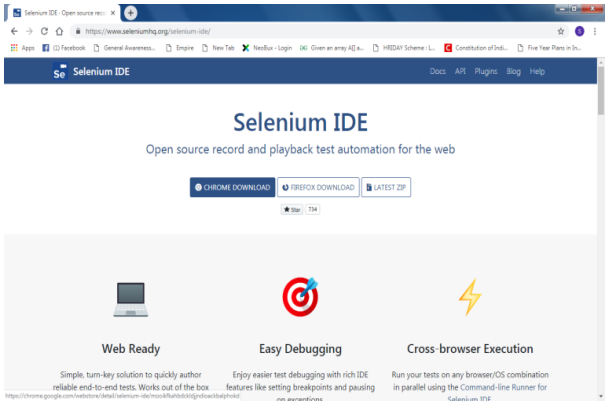
Selenium is a group of various software tools each with a different approach to supporting test automation. The entire suite of tools results in a rich set of testing functions specifically geared to the needs of testing of web applications of all types. These operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behaviour. One of Selenium's key features is the support for executing one's tests on multiple browser platforms.

1) SELENIUM IDE

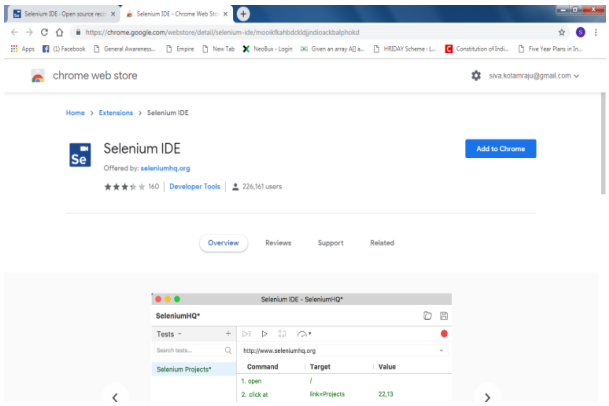
Deploying Selenium IDE:

Step1: Open the URL seleniumhq.org,

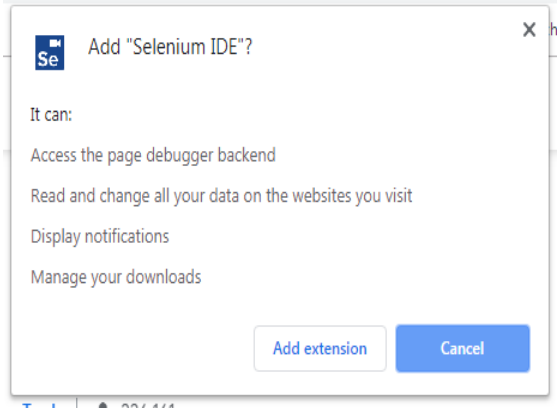
Step2: Click on Selenium IDE



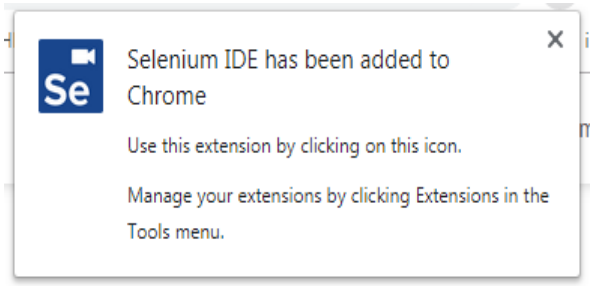
Step3: Click on Chrome download



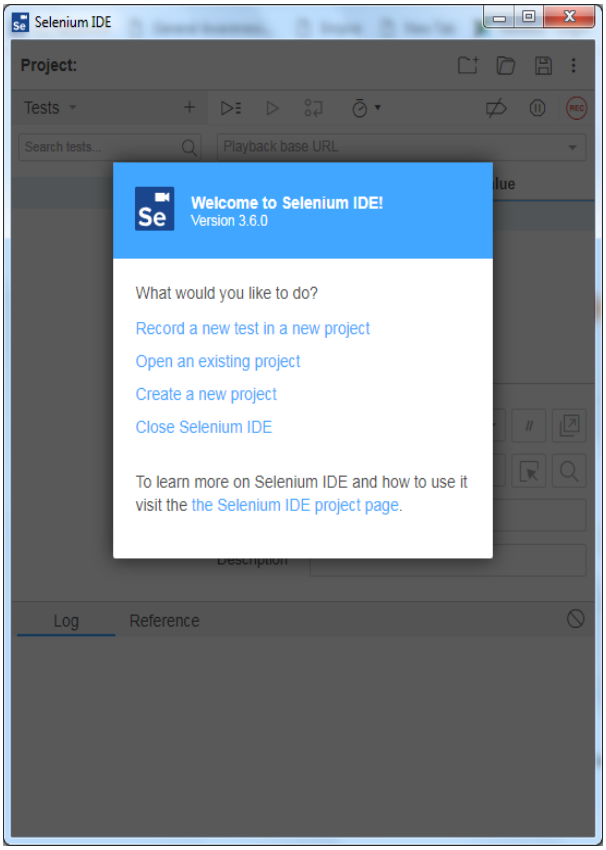
Step4: Click on Add to Chrome



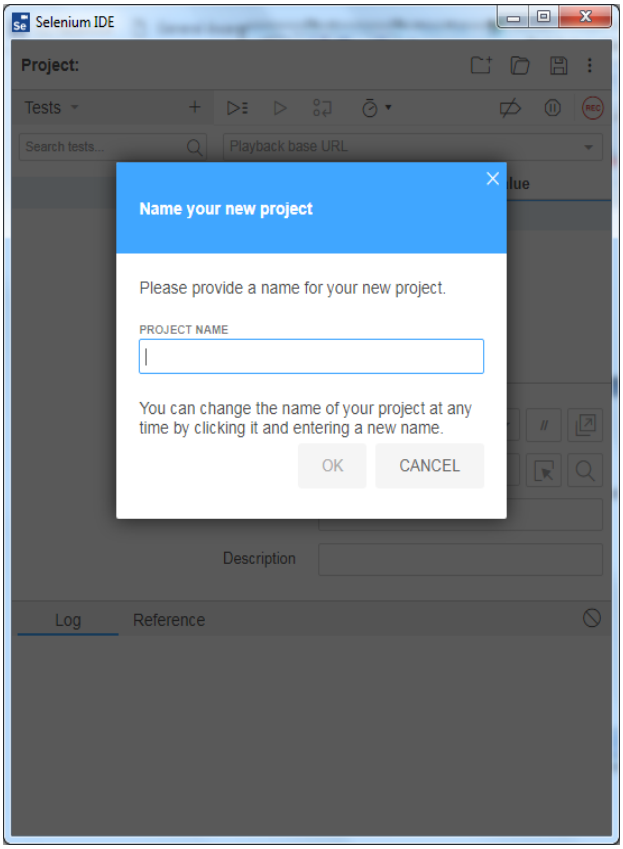
Step5: Click on Add extension



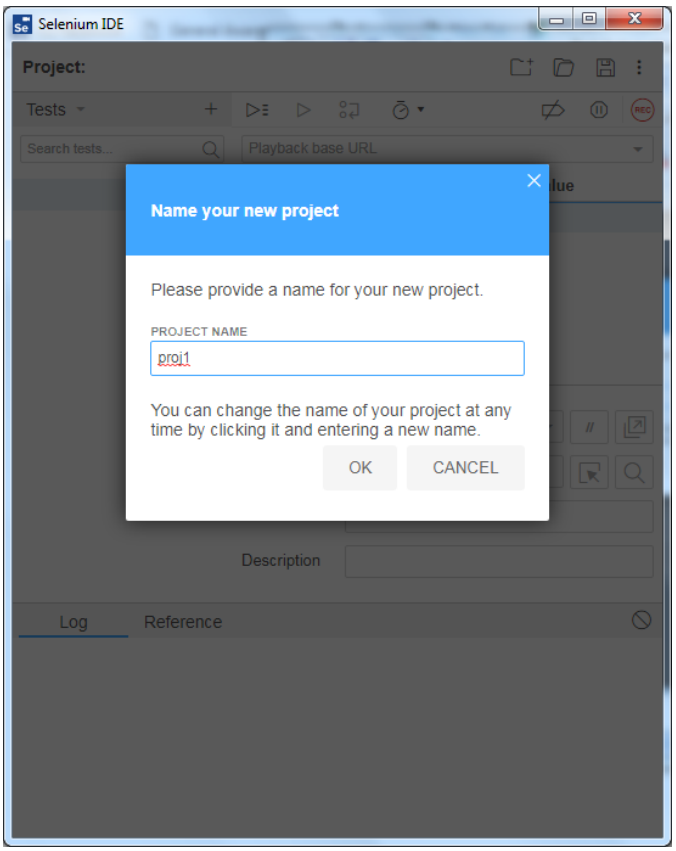
Step6: Click on Selenium IDE Icon on the Browser



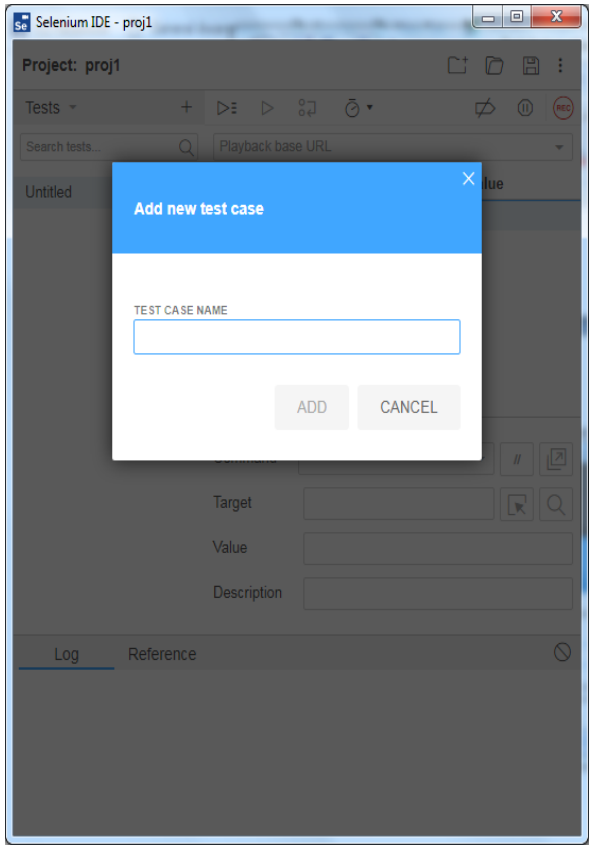
Step7: Click on Create a new project



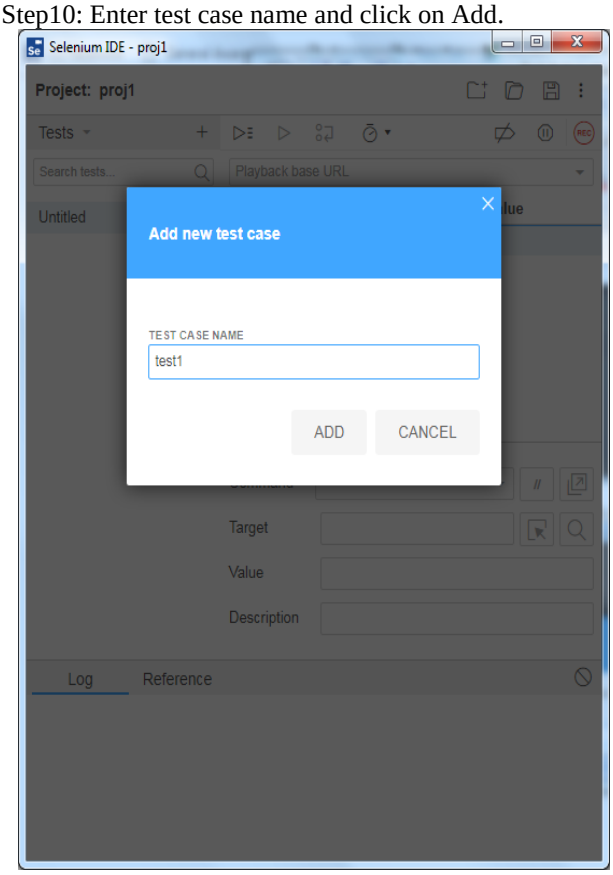
Step8: Enter Project name and click on OK



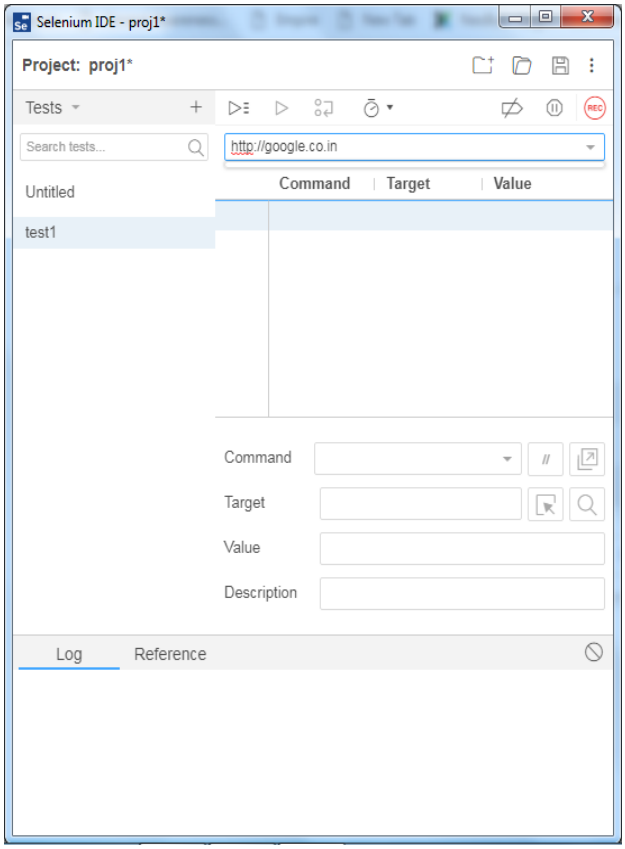
Step9: Click on '+' button to add new test



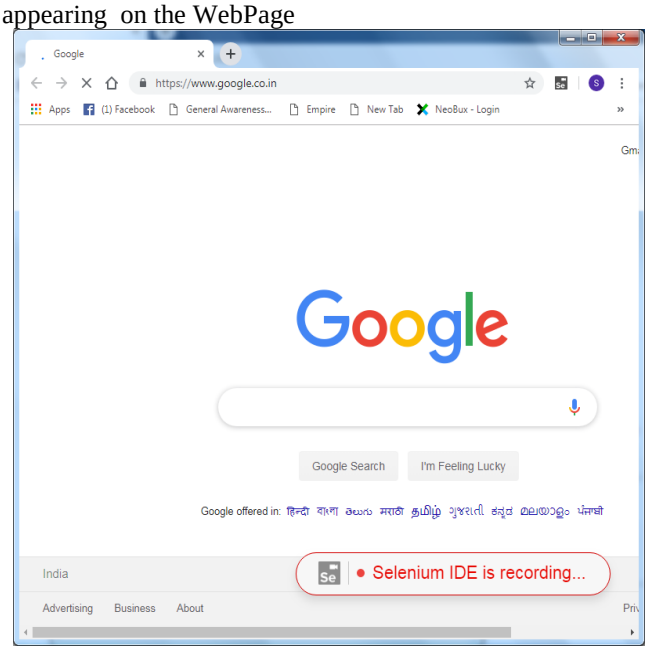
Step10: Enter test case name and click on Add.



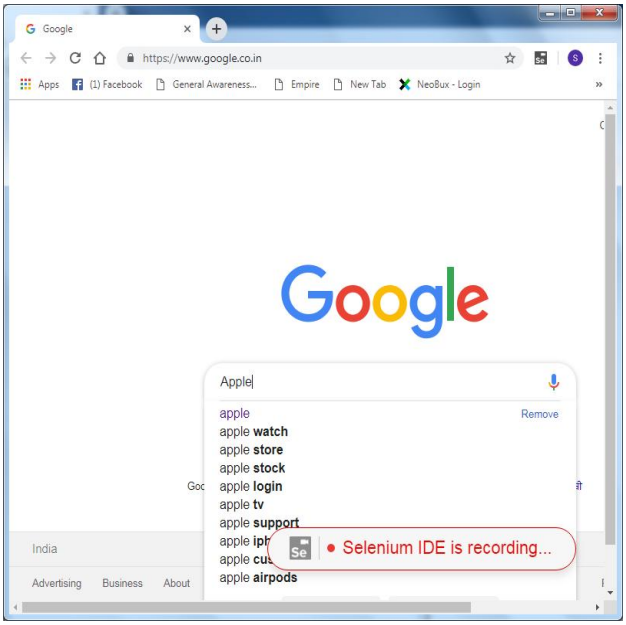
Step11:Enter playback base URL as http://www.google.co.in and click on REC button



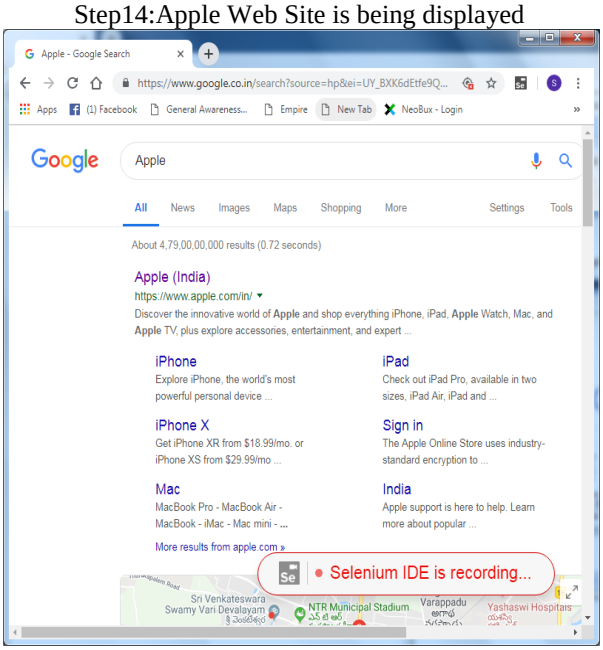
Step12: See that "Selenium IDE is recording" message is appearing on the WebPage



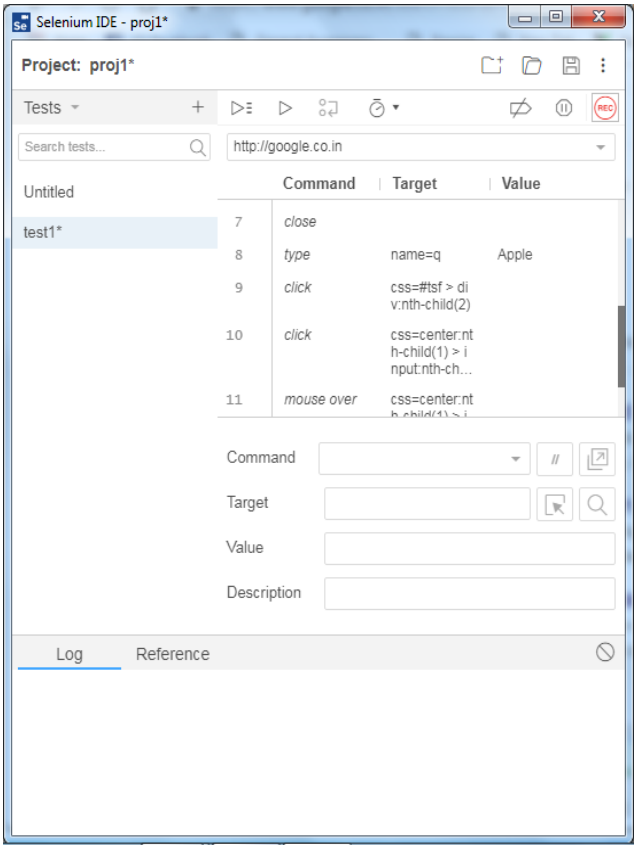
Step13: Enter Apple in the Search box



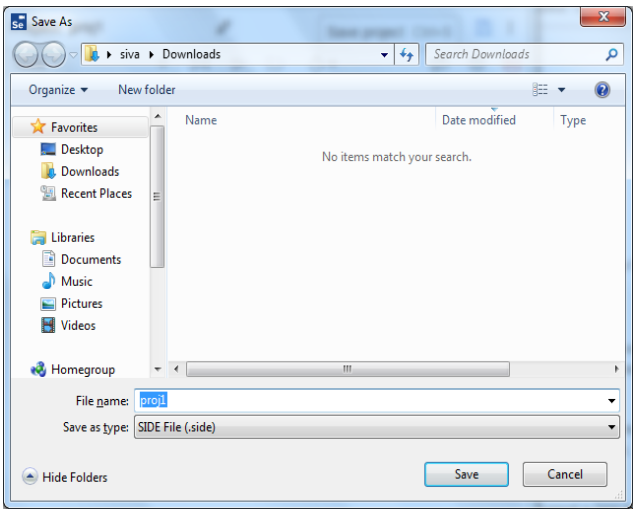
Step14:Apple Web Site is being displayed



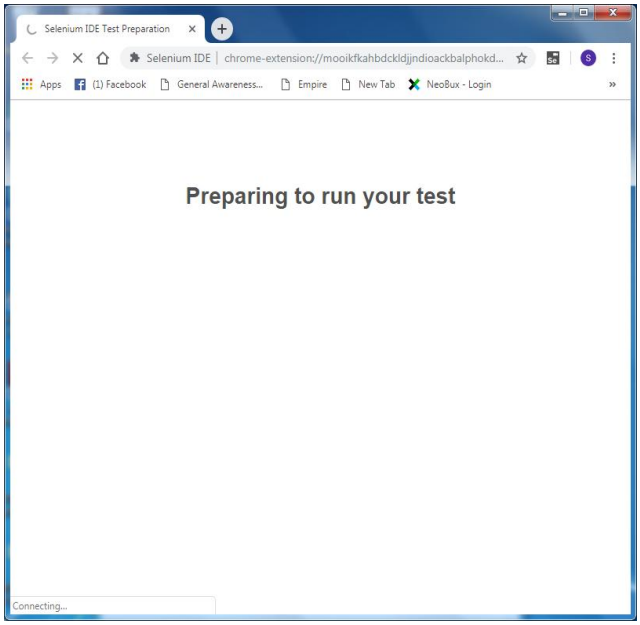
Step15:Click on Stop Recording REC button



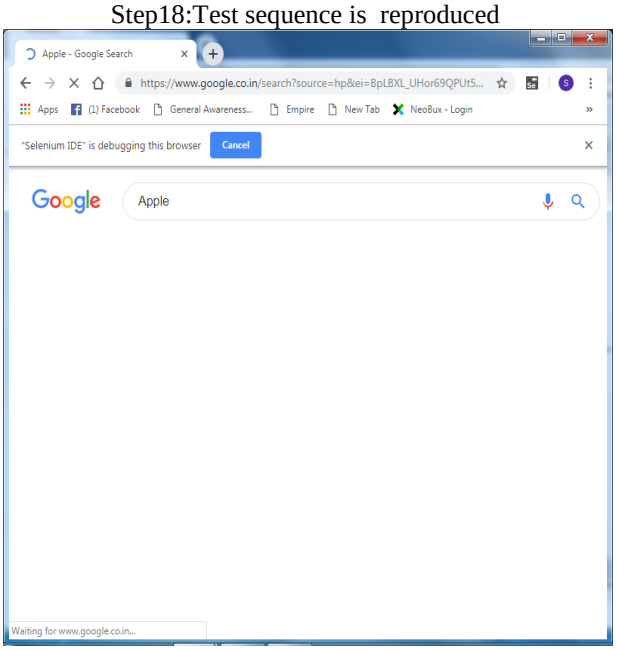
Step16: Click on Save Project button as Proj1 with extension ".side"



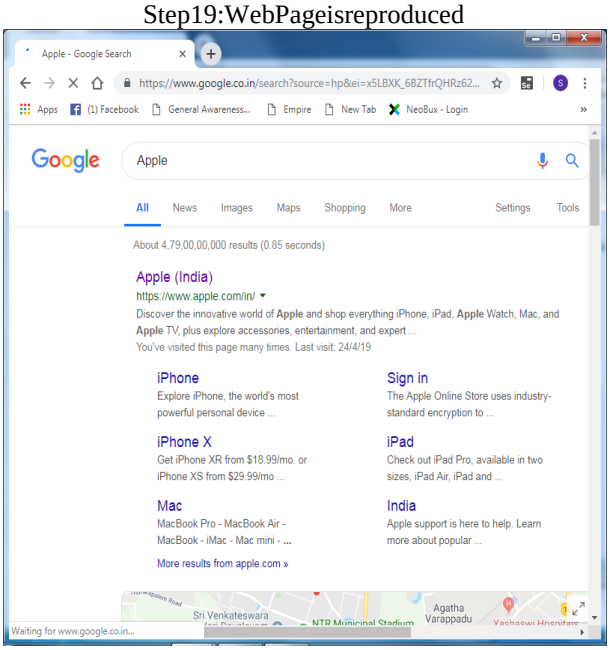
Step17: Click on Current Run test (Arrow button)



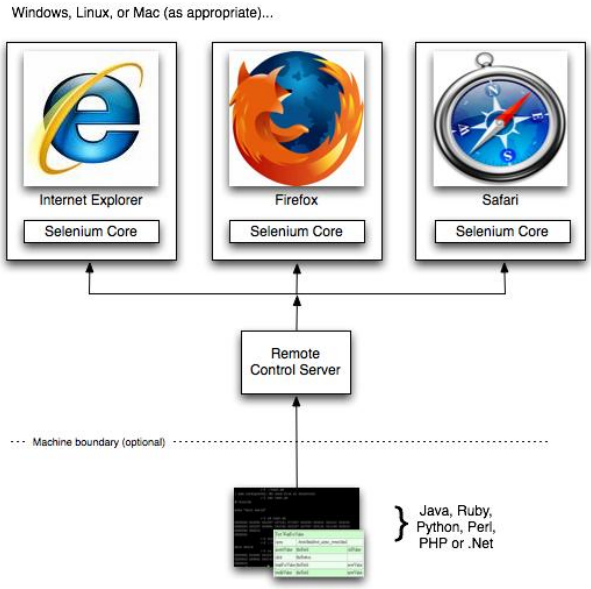
Step18:Test sequence is reproduced



Step19:WebPageisreproduced



2) SELENIUM RC



.

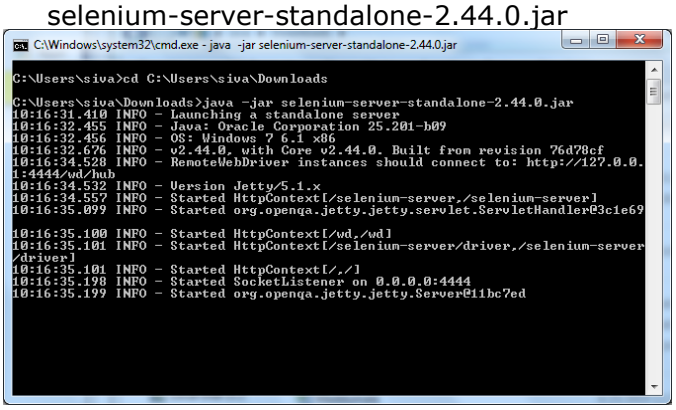
Client Libraries

Installation

Installing Selenium Server

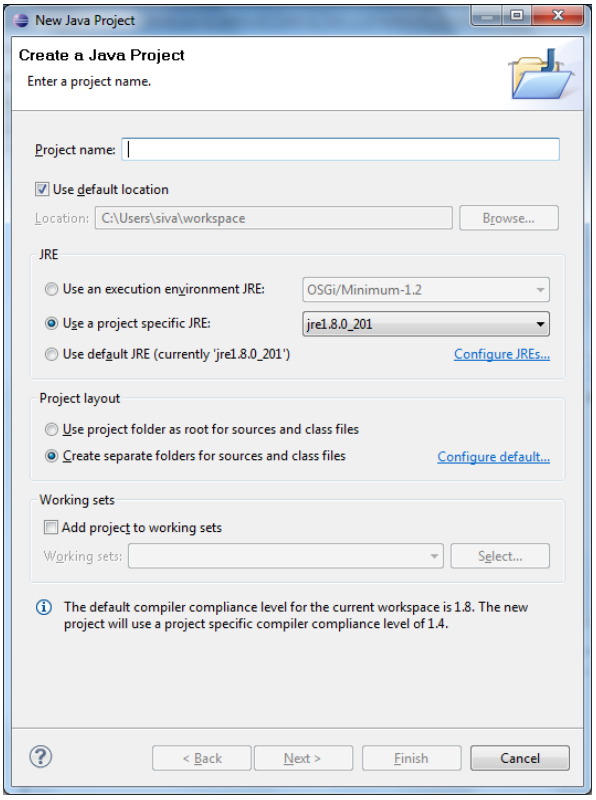
Running Selenium Server

java -jar selenium-server-standalone-2.44.0.jar

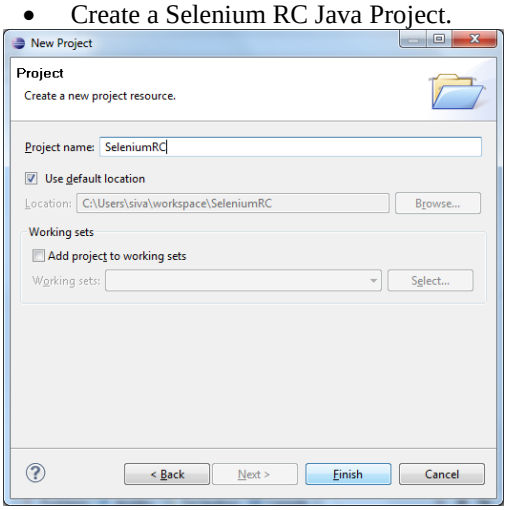


Using the Java Client Driver

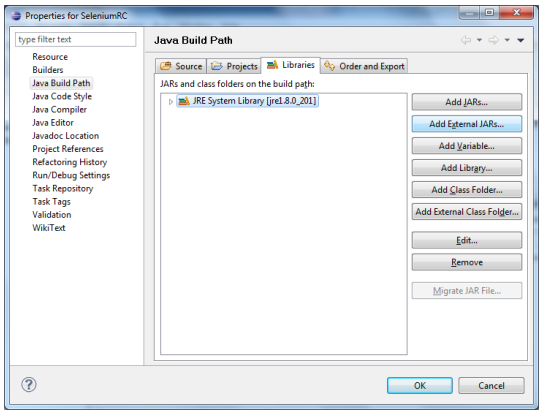
- Download the Selenium Java client driver Zip
- Extract selenium-java-2.53.0.jar, selenium-java-2.53.0-src.jar files
- Open Eclipse IDE



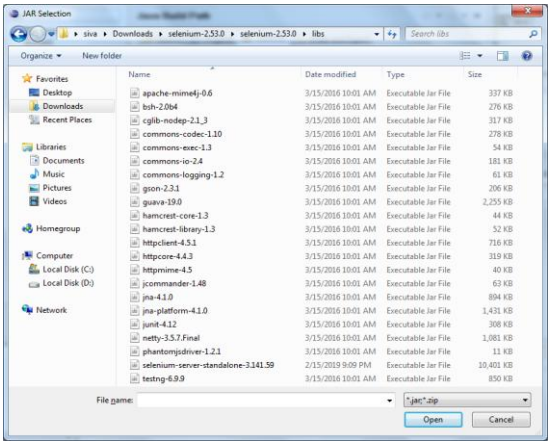
- Create a Selenium RC Java Project.



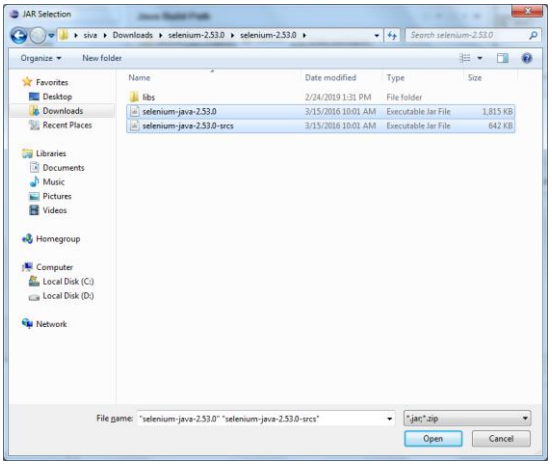
- Configure Build Path under Project Properties and add External Jar



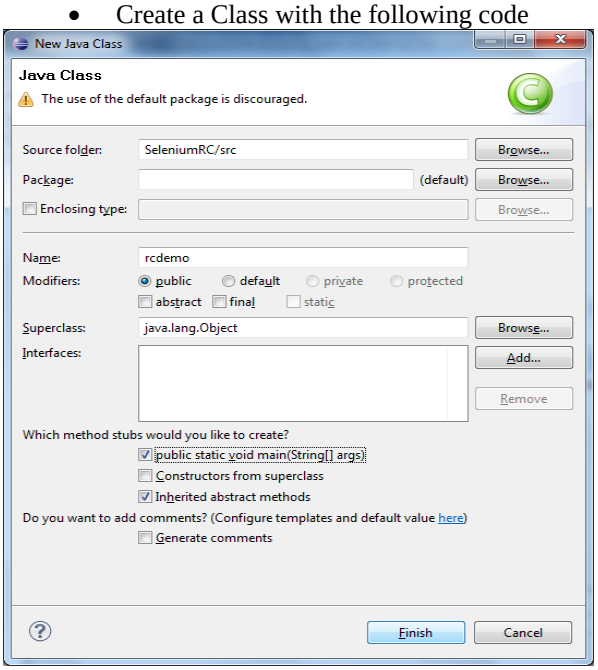
- Add External Libraries



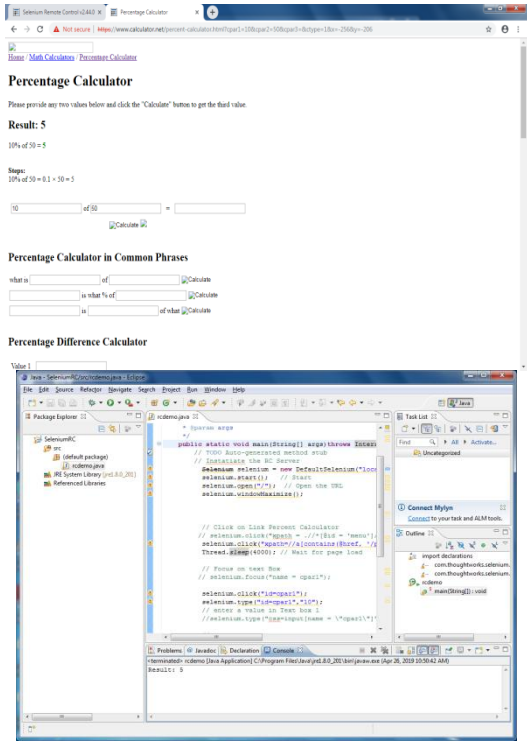
- Add the selenium-java-2.53.0.jar files and selenium-java-2.53.0-src.jar files to your project as references.

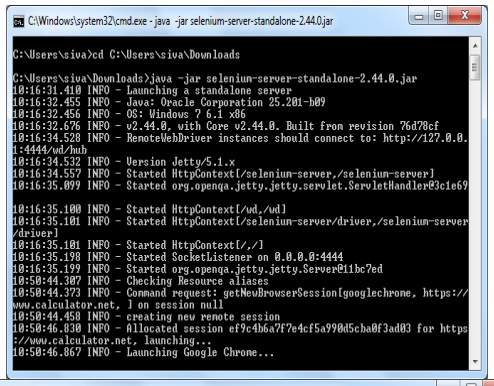


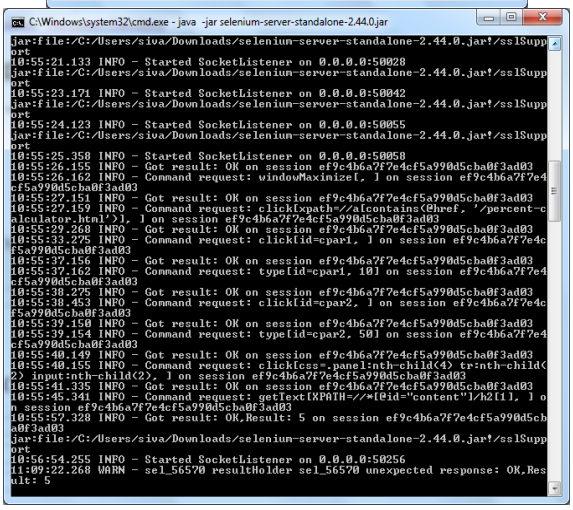
- Create a Class with the following code



```
import com.thoughtworks.selenium.DefaultSelenium;
import com.thoughtworks.selenium.Selenium;

public class rcdemo {
  public static void main(String[] args) throws InterruptedException {

    // Instatiate the RC Server
    Selenium selenium = new DefaultSelenium("localhost", 4444 , "googlechrome",
"https://www.calculator.net");
    selenium.start();  // Start
    selenium.open("/"); // Open the URL
    selenium.windowMaximize();
  dbgfa36 cea3


    // Click on Link Percent Calculator
    // selenium.click("xpath = //*[@id = 'menu']/div[4]/div[3]/a");
    selenium.click("xpath=//a[contains(@href, 'percent-calculator.html')]");
    Thread.sleep(4000); // Wait for page load

    // Focus on text Box
    // selenium.focus("name = cpar1");

    selenium.click("id=cpar1");
    selenium.type("id=cpar1","10");
    // enter a value in Text box 1
    //selenium.type("css=input[name = \"cpar1\"]","10");

    // enter a value in Text box 2
    //selenium.focus("name = cpar2");
    // selenium.type("css = input[name = \"cpar2\"]","50");
   selenium.click("id=cpar2");
    selenium.type("id=cpar2","50");
    // Click Calculate button
    //selenium.click("xpath = //*[@id = 'content']/table/tbody/tr/td[2]/input");
    selenium.click("css=.panel:nth-child(4) tr:nth-child(2) input:nth-child(2)");
    // verify if the result is 5
    // String result = selenium.getText("//*[@id = 'content']/p[2]");
    Thread.sleep(4000);
    String result = selenium.getText("XPATH=//*[@id="content"]/h2[1]");
    System.out.println(result);
  }
}
```



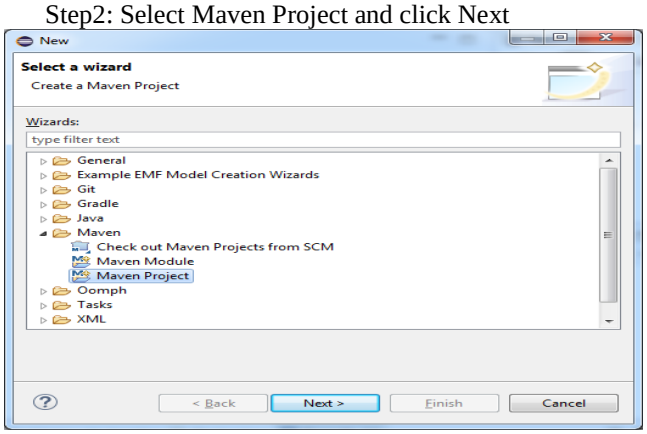




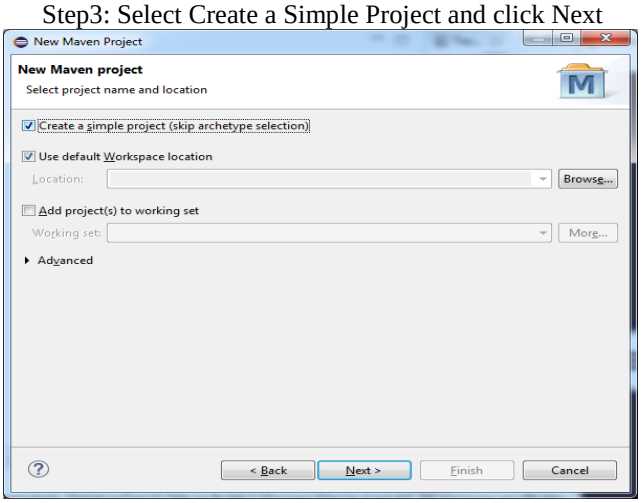
3) SELENIUM WEBDRIVER

Step1: Open Eclipse IDE and Press CTRL+N
File->New->Other

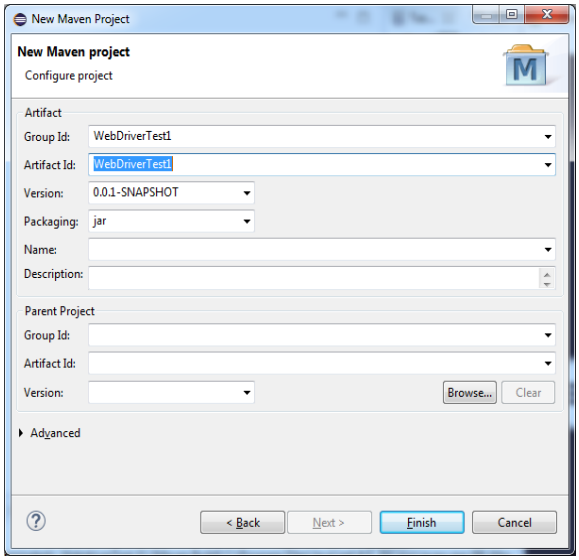
Step2: Select Maven Project and click Next



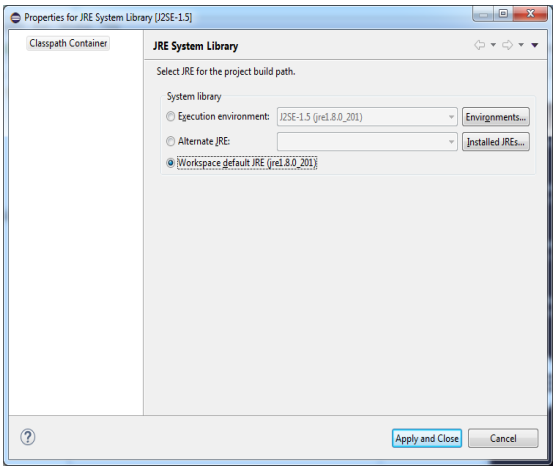
Step3: Select Create a Simple Project and click Next



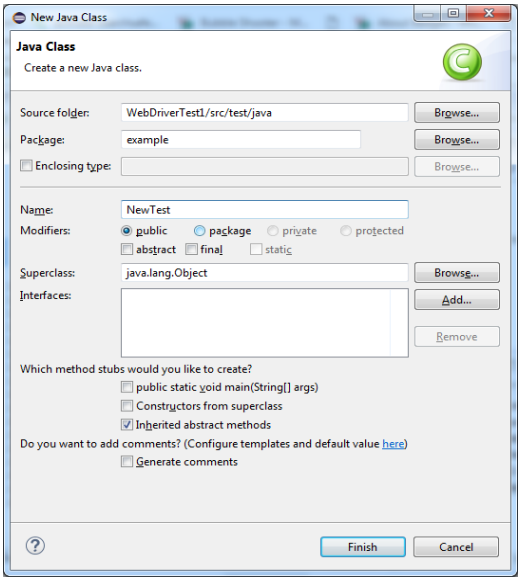
Step4: Give Group Id and Artifact Id as **WebDriverTest1** and click Finish



Step5: Expand WebDriverTest1 and Right Click **JRESystemLibrary** ,Select Properties, Select JRE System library to Workspace default JRE(jre1.8.0_201) and click on apply and close



Step6:Select pom.xml file from project explorer and Add the Selenium, Maven, TestNG, Junit dependencies to pom.xml in the <project> node:

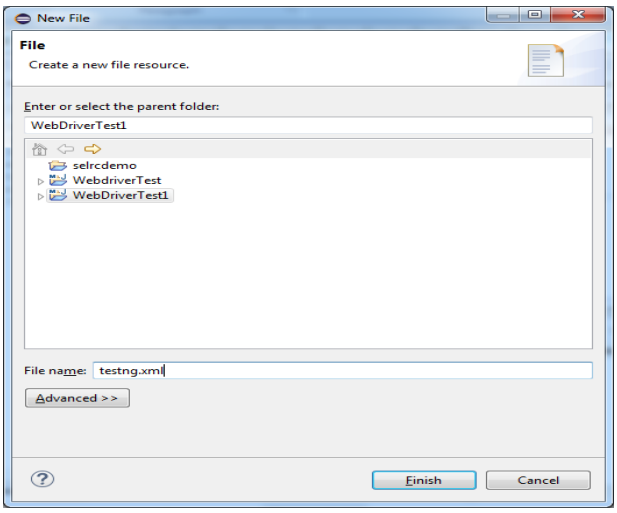


Step9:Create testng.xml file
File->new->FILE and give testng.xml in File name and click on finish

```
<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</antifactId>
        <version>2.45.0</version>
            </dependency>
    <dependency>
        <groupId>org.testng</groupId>
        <artifactId>testng</artifactId>
        <version>6.8</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Step7: Create a New TestNG Class. Enter Package name as "example" and "NewTest" in the **Name**: textbox and click on the **Finish** button as shown in the following screenshot:
Step8: Add following code to the **NewTest** class

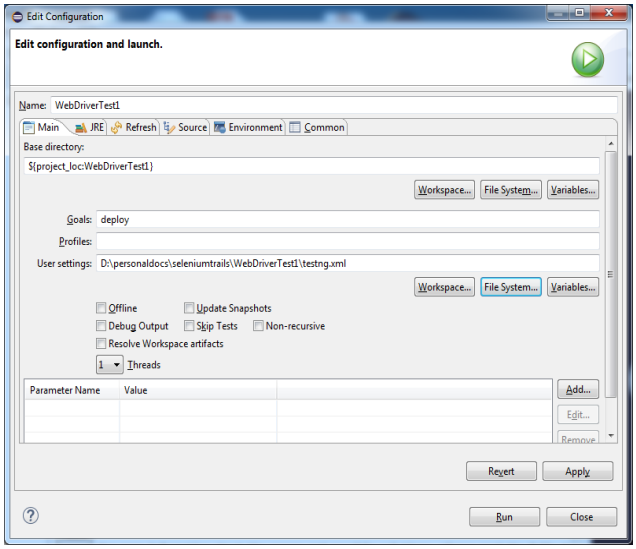
```
package example;
    import org.openqa.selenium.By;
    import org.openqa.selenium.WebDriver;
    import org.openqa.selenium.chrome.ChromeDriver;
    import org.openqa.selenium.firefox.FirefoxDriver;
    import org.testng.Assert;
    import org.testng.annotations.Test;
    import org.testng.annotations.BeforeTest;
    import org.testng.annotations.AfterTest;
    public class NewTest {
        private WebDriver driver;
            @Test
            public void testEasy() {
                driver.get("http://demo.guru99.com/test/guru99home/");

    String title = driver.getTitle();
        Assert.assertTrue(title.contains("Demo Guru99 Page"));
            }

            @BeforeTest
            public void beforeTest() {
                System.setProperty("webdriver.chrome.bin", "C:\\Program Files
(x86)\\Google\\Chrome\\Application\\chrome.exe");
                System.setProperty("webdriver.chrome.driver",
"D:\\chromedriver_win32\\chromedriver.exe");
                driver = new ChromeDriver();
            }
            @AfterTest
            public void afterTest() {
                driver.quit();
            }
        }
```



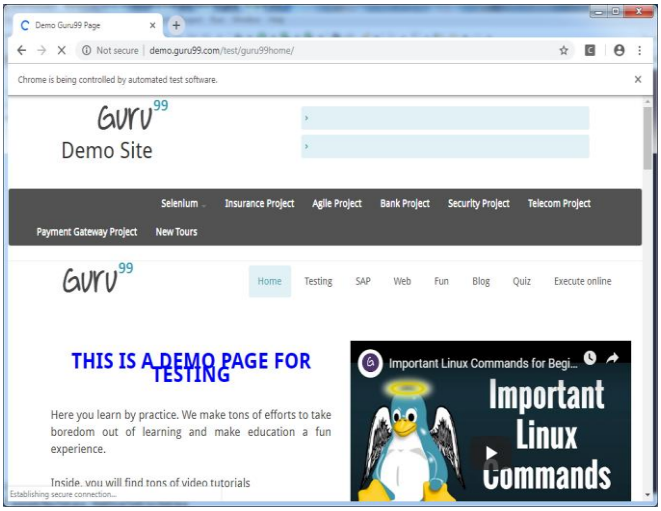
Step10: Add the following code to testing.xml
<settings name="*example*">
<test name="*testngTest*">
<classes>

```
 <class name="example.NewTest" />
 </classes>
 </test>
</settings>
```

Step11:
Put deploy in Goals and select the testng.xml file in the Project Directory and click on Run



Step 12:The following page is displayed in Google chrome Browser



IX. CONCLUSION

Deployed Selenium IDE ,Selenium RC and Selenium WebDriver.In Selenium IDE,process of recoring tests and playback recorded tests.In Selenium RC,remote access to web application, executing test case there and returning the results .In Selenium WebDriver deployed an application using Maven Build.Selenium Grid is the future work

REFERENCES

1. Win Runner in simple steps by Hakeem Shittu, 2007Genixpress.
2. https://www.guru99.com/maven-jenkins-with-selenium-complete-tutorial.html
3. https://www.youtube.com/watch?v=eNCzuCxuEXM&list=PL9ooVrP1hQOFP9H8Y15DVGCA6GavhgJ8a&index=28
4. https://www.seleniumhq.org