Essential Software Testing

This is a BLOG where I (Tushar Singhal) write about my experiences with Testing Software and Learning in Software Testing :)

SEARCH THIS BLOG

Search

LABELS

- Automation Testing Jobs (83)
- Manual Testing Jobs (27)
- Test Cases (14)
- Selenium Webdriver (7)
- Java Programs Asked In Interviews (5)
- Motivational (5)
- API's Testing (4)
- WebServices and API's Jobs (4)
- Developer Jobs (3)
- PerformanceTestingJob (3)
- Manual Testing (2)
- BusinessAnalystJob (1)
- Java Interview Questions with Answers (1)
- Jobs (1)
- SQL Queries (1)
- Unix (1)

FRIDAY, 16 JUNE 2017

Fwd: Job | WALKIN FOR MANUAL TESTER POSITION on SATURDAY i.e. 17th JUNE 2017 (10:00AM - 1:00PM)

Walkin are scheduled on Saturday i.e. 17th June for Manual Testing Position

Please find below the mandate :

About Us

We are an IT Services, Consulting and a Product Company that helps clients optimize their software investments and manage critical business processes to reduce the total cost of software ownership. You can visit our website www.puresoftware.com. We own a cutting edge technology Product Group (www.arttha.com), an all in one consolidated mobile transaction processing engine, that allows next generation mobile payment processing for different industry verticals. In the last decade and more, we have firmly established ourselves as leaders in project delivery, resource training, software development, automated testing. Our focus has been on creating engagement models that drive cost and time reduction in projects while maintaining the highest software engineering standards. We provide our clients with a unique multi-point engagement model, which enables faster and more meaningful communications between various functional teams involved in the project. We take pride in calling ourselves a CMM level 4 certified organization. Over the past decade, we have provided solutions for various sectors including Telecommunications, Banking, Pharmaceuticals, Casino gaming, ISVs and Software product development organizations.

Skills : .Manual Tester

Experience â€“ 2-5 years

Responsibilities

1 . Should have work experience in Manual Testing

2. Should have experience on Database and Web Services

Work Location : Noida

Salary â€“ Negotiable

Venue Details :

PureSoftware

Plot B 9-A, Green Boulevard (Tower B, 3rd Floor),

Sector 62, Noida (UP)- 201301

Day& Date : Saturday ; 17th June 17

Timings : 10:00AM â€“ 1:00PM

Contact Person : Garima Gupta

Regards,

Garima Gupta

garima.gupta@puresoftware.com

By Tushar Singhal at June 16, 2017 No comments: Links to this post

THURSDAY, 15 JUNE 2017

Introduction to Selenium WebDriver

Selenium WebDriver alias **Selenium 2** is a test tool that allows you to write automated web application UI tests in any programming language against any HTTP website.

Selenium WebDriver is the latest and most popular flavor of Selenium which allows the test developers to write tests using their favorite language to test on different browsers.

Pre-requisites for Selenium WebDriver:

- *Selenium IDE* - Please go through all the Selenium IDE posts (i.e. from Post#1 What is Selenium? to Post#27 Finding Locators for UI elements using the Selenium IDE's recording feature)
- *Java* - Please go through all the 'Java for Selenium' posts (i.e. from Post#28 Java for Selenium to Post# 157 Using lastIndexOf( ) method with StringBuffer)
- *Eclipse IDE* - Please go through Eclipse IDE posts (i.e. from Post#8 Install Eclipse IDE to Post#10 How to create Projects in Eclipse IDE?)

What is Selenium WebDriver?

- Selenium WebDriver allows you to write Automation tests in any supported programming language (say Java).

TOTAL PAGEVIEWS

80773

POPULAR POSTS

Manual Test Cases For 'Email' Functionality

Inbox Check that a newly received email is displayed as highlighted in the Inbox section. Check that a newly received email has correct...

Manual Test Cases For Security Testing

1. Check for SQL injection attacks 2. Secure pages should use HTTPS protocol 3. Page crash should not reveal application or server inf...

Manual Test Cases For Database Testing

1. Check if correct data is getting saved in database upon successful page submit 2. Check values for columns which are not accepting nul...

Manual Test Cases For 'Calculator'

Check if the calculator is normal calculator or scientific calculator Check that all the buttons are present and text written on them is...

Job | Sr.QA API testing Engineer Professionals Opening with IT Company @Noida

Hi, This has reference to your CV advertised on Portal We would like to know if you are contemplating a change at this point of your car...

- Selenium WebDriver is also known as Selenium 2.
- Selenium WebDriver is an open source and free tool.
- Selenium WebDriver is the main Selenium Project currently (i.e. Selenium RC or Selenium 1 was the main project earlier). But Selenium 2 still runs Selenium 1's Selenium RC interface for backwards compatibility.
- Selenium 2 can run Selenium RC command and the concept is known as Backed Selenium.
- Selenium WebDriver has support for almost all browsers
- Selenium WebDriver has support for several programming languages like Java, C#, Ruby, Python, Pearl, Javascript and PHP)

How Selenium WebDriver works ?

In order to know how Selenium Web Driver works:

- First you have to understand how the Selenium RC works by going through the Selenium RC Post # 1
- Next find out what is the difference between Selenium RC and Selenium WebDriver by going through the below content.

Difference between Selenium WebDriver and Selenium RC:

In Selenium RC we have to use Selenium RC Server as mediator between the Automation Code and the Browser. Generally Browsers wont allow any external code to get executed on any Website they are holding. Since our Selenium Automation code is treated as external code by the Browsers, we required a kind of mediator who can communicate with the browsers, so that the browser can allow the execution of Selenium Automation code on the website they are holding. Hence Selenium RC Server acts as a proxy to allow the Selenium Automation Code to get executed on the Application Under Test in Browsers. So in Selenium RC we have to make Selenium command calls to the browsers via Selenium RC Server.

But in Selenium WebDriver, we don't require any Server which acts as a mediator. We can make direct Selenium command calls to the browsers.This is possible because of the Browser drivers used in the Selenium Automation code. There is a separate driver for each browser. Suppose if we want to run our Selenium Automation code on Internet Explorer browser, we have to the use the Internet Explorer WebDriver. Here WebDriver of browser will communicate with the browser. We use WebDriver API (Application Programming Interface) code in the our Selenium Automation Code and hence our Final Automation Tests containing the WebDriver API code can directly communicate with the browsers.

If your browser and automation tests will all run on the same machine, then your tests only use the WebDriver API, you do not need to run the Selenium-Server (i.e. WebDriver will run the browser directly).

Are there any cases where you have to use Selenium-Server with Selenium-WebDriver ?

Yes. As we have to use Selenium-Grid concept for distributing our tests over multiple machines and run. In order to run the browser on a remote machine, we have to use Selenium-Server for communicating with the browsers on remote machines and running the tests on remote machines.

By Tushar Singhal at June 15, 2017 No comments: Links to this post

WEDNESDAY, 14 JUNE 2017

Selenium WebDriver - Basic Action Commands And Operations With Examples

1. Creating New Instance Of Firefox Driver

```
WebDriver driver = new FirefoxDriver();
```

2. Command To Open URL In Browser

```
driver.get("http://essentialsoftwaretesting.blogspot.in");
```

3. Clicking on any element or button of webpage

```
driver.findElement(By.id("submitButton")).click();
```

4. Store text of targeted element in variable

```
String dropdown = driver.findElement(By.tagName("select")).getText();
```

5. Typing text in text box or text area.

```
driver.findElement(By.name("fname")).sendKeys("My First Name");
```

6. Applying Implicit wait in webdriver

```
driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
```

7. Applying Explicit wait in webdriver with WebDriver canned conditions.

```
WebDriverWait wait = new WebDriverWait(driver, 15);

wait.until(ExpectedConditions.textToBePresentInElementLocated(By.xpath("//div[@id='timeLeft']"), "Time left: 7 seconds"));
```

8. Get page title in selenium webdriver

```
driver.getTitle();
```

9. Get Current Page URL In Selenium WebDriver

```
driver.getCurrentUrl();
```

10. Get domain name using java script executor

```
JavascriptExecutor javascript = (JavascriptExecutor) driver;

String CurrentURLUsingJS=(String)javascript.executeScript("return document.domain");
```

11. Generate alert using webdriver's java script executor interface

```
JavascriptExecutor javascript = (JavascriptExecutor) driver;

javascript.executeScript("alert('Test Case Execution Is started Now..');");
```

12. Selecting or Deselecting value from drop down in selenium webdriver.

Select By Visible Text

```
Select mydrpdwn = new Select(driver.findElement(By.id("Carlist")));

mydrpdwn.selectByVisibleText("Audi");
```

It will select value from drop down list using visible text value = "Audi".

Select By Value

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));

listbox.selectByValue("Italy");
```

It will select value by value = "Italy".

Select By Index

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));

listbox.selectByIndex(0);
```

It will select value by index= 0(First option).

Deselect by Visible Text

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));

listbox.deselectByVisibleText("Russia");
```

It will deselect option by visible text = Russia from list box.

Deselect by Value

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));

listbox.deselectByValue("Mexico");
```

It will deselect option by value = Mexico from list box.

Deselect by Index

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));
```

```
listbox.deselectByIndex(5);
```

It will deselect option by Index = 5 from list box.

Deselect All

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));

listbox.deselectAll();
```

It will remove all selections from list box of software application's page.

isMultiple()

```
Select listbox = new Select(driver.findElement(By.xpath("//select[@name='FromLB']")));

boolean value = listbox.isMultiple();
```

13. Navigate to URL or Back or Forward in Selenium Webdriver

```
driver.navigate().to("http://only-testing-blog.blogspot.com/2014/01/textbox.html");

driver.navigate().back();

driver.navigate().forward();
```

14. Verify Element Present in Selenium WebDriver

```
Boolean iselementpresent = driver.findElements(By.xpath("//input[@id='text2']")).size()!= 0;
```

15. Capturing entire page screenshot in Selenium WebDriver

```
File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(screenshot, new File("D:\\screenshot.jpg"));
```

16. Generating Mouse Hover Event In WebDriver

```
Actions actions = new Actions(driver);

WebElement moveonmenu = driver.findElement(By.xpath("//div[@id='menu1']/div"));

actions.moveToElement(moveonmenu);

actions.perform();
```

17. Handling Multiple Windows In Selenium WebDriver.

Get All Window Handles.

```
Set<String> AllWindowHandles = driver.getWindowHandles();
```

Extract parent and child window handle from all window handles.

```
String window1 = (String) AllWindowHandles.toArray()[0];

String window2 = (String) AllWindowHandles.toArray()[1];
```

Use window handle to switch from one window to other window.

```
driver.switchTo().window(window2);
```

18. Check Whether Element is Enabled Or Disabled In Selenium Web driver.

```
boolean fname = driver.findElement(By.xpath("//input[@name='fname']")).isEnabled();

System.out.print(fname);
```

19. Enable/Disable Textbox During Selenium Webdriver Test Case Execution.

```
JavascriptExecutor javascript = (JavascriptExecutor) driver;

String todisable = "document.getElementsByName('fname')[0].setAttribute('disabled', '');";
```

```
javascript.executeScript(todisable);

String toenable = "document.getElementsByName('lname')[0].removeAttribute('disabled');";

javascript.executeScript(toenable);
```

20. Selenium WebDriver Assertions With TestNG Framework

assertEquals

```
Assert.assertEquals(actual, expected);
```

assertEquals assertion helps you to assert actual and expected equal values.

assertNotEquals

```
Assert.assertNotEquals(actual, expected);
```

assertNotEquals assertion is useful to assert not equal values.

assertTrue

```
Assert.assertTrue(condition);
```

assertTrue assertion works for boolean value true assertion.

assertFalse

```
Assert.assertFalse(condition);
```

assertFalse assertion works for boolean value false assertion.

21. Submit() method to submit form

```
driver.findElement(By.xpath("//input[@name='Company']")).submit();
```

It will submit the form.

22. Handling Alert, Confirmation and Prompts Popups

```
String myalert = driver.switchTo().alert().getText();
```

To store alert text.

```
 driver.switchTo().alert().accept();
```

To accept alert.

```
 driver.switchTo().alert().dismiss();
```

To dismiss confirmation.

By Tushar Singhal at June 14, 2017 No comments: Links to this post

Newer Posts Home Older Posts

Subscribe to: Posts (Atom)

ABOUT ME

Tushar Singhal

6+ Years of industry experience in the area of Software Testing (Manual and Automation) with understanding of Test Planning, Test Design, Test Execution and Defect Reporting and Tracking. Experience to work on Tools like Jmeter, Eclipse IDE, Selenium Webdriver, Sikuli, TestNG, Maven, Jenkins and SOAP UI

View my complete profile