Essential Software Testing

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

Showing posts with label **Selenium Webdriver**. Show all posts

FRIDAY, 9 JUNE 2017

Selenium Webdriver - Xpath Explanation with Example...!!!

You need to provide any element locator(like id, name, css path, xpath etc.) in target column of selenium window to locate that specific element to perform some action on it and you are already aware about that. In previous post, we have learn about how to identify element id or name.If you have worked with selenium IDE then you knows that sometimes elements does not contains id or name. Locating element by xpath is the another way of locating element and you can use it as a alternative of id or name of element.

Xpath in XML document shows the direction of element location through nodes and attributes. Let we try to understand how to identify Xpath of element with examples.

Above given image is taken from http://www.wikipedia.org/. Look into the image there are three fields 1. Input text box 2. select drop down and 3. input button. And bellow of those fields there is expansion of relative XML nodes through firebug. As you see in image, you can use "id=searchInput" or "name=search" to identify input text box to type something in to it as bellow given example.

New Test		
Command	**Target**	**Value**
open	http://www.wikipedia.org/	
type	id=searchInput	ID Example

or

New Test		
Command	**Target**	**Value**
open	http://www.wikipedia.org/	
type	name=search	Name Example

Xpath Tutorials
Now if you want to identify same element (input textbox) with xpath then you can use any of the bellow given syntax in to the target column with type command in above example.

Locating element using Xpath with Examples for input text box

1. Identifying Xpath using full path of XML
xpath=//body/div[3]/form/fieldset/input[2] //// Here //body is the main root node, /div[3] describes the 3rd div child node of parent node body, /form describes the child node form of parent node div[3], /fieldset describes the child node fieldset of parent node form, /input[2] describes the 2nd input child node of parent node fieldset.

New Test		
Command	**Target**	**Value**
open	http://www.wikipedia.org/	
type	xpath=//body/div[3]/form/fieldset/input[2]	Xpath Example1

2. Writting Xpath using last()
xpath=//body/div[3]/form/fieldset/input[last()-2] //// Here /input[last()-2] describes the 3rd upper input node(input[2]) from last input node.
xpath=//body/div[3]/form/fieldset/*[last()-3] //// Here /*[last()-3] describes the 4th upper node(input[2]) from last node.

New Test		
Command	**Target**	**Value**
open	http://www.wikipedia.org/	
type	xpath=//body/div[3]/form/fieldset/input[last()-2]	Xpath Example2

3. Xpath locator using @ and attribute
xpath=//body/div[3]/form/fieldset/input[@type='search'] //// Here /input[@type='search'] describes the input node having attribute type='search'.

New Test		
Command	**Target**	**Value**
open	http://www.wikipedia.org/	
type	xpath=//body/div[3]/form/fieldset/input[@type='search']	Xpath Example3

4. Xpath expression using @ and attribute
xpath=//body/div[3]/form/fieldset/input[@accesskey='F'] //// Here /input[@accesskey='F'] describes the input node having attribute @accesskey='F'. Another way of same is as bellow.

New Test		
Command	**Target**	**Value**
open	http://www.wikipedia.org/	
type	xpath=//body/div[3]/form/fieldset/input[@accesskey='F']	Xpath Example4

5. Xpath syntax using @ and attribute
xpath=//input[@accesskey='F'] //// Here //input[@accesskey='F'] describes the input node having attribute @accesskey='F'. Try it by using it in above example.

6. Xpath example using @ and attribute
xpath=//input[@type='search'] //// Here /input[@type='search'] describes the input node having attribute type='search'. Try it by using it in above example.

7. XML Xpath using /descendant:: keyword
xpath=//div[@class='search-container']/descendant::input[@accesskey='F'] //// Here i have used descendant in between. In this case i have described only starting node div with attribute class='search-container' and final node input with accesskey='F' attribute. So not need to describe in between nodes. Try it by using it in above example.

8. Xpath query example using contains keyword
xpath=//input[contains(@id, "searchInput")] ////Here i have used contains keyword to identify id attribute with text "searchInput". Try it by using it in above example.

9. xpath using and with attributes
xpath=//input[contains(@id, "searchInput") and contains(@accesskey,"F")] ////In this example, It will look at two attributes in input node. Try it by using it in above example.

10. XML xpath value value using position()
xpath=//div[@class='search-container']/descendant::input[position()=2] ////This xpath will select input node which is on number 2 position and it is for input text box as shown in image. Try it by using it in above example.

11. Using starts-with keyword
xpath=//input[starts-with(@type, "s")] //// In this example, It will find input node with attribute is 'type' and its value is starting with 's' (here it will get type = 'search').

12. Using OR (|) condition with xpath
xpath=//input[@accesskey='F'] | //input[@id='searchInput']

xpath=//input[@accesskey='F' or @id='searchInput'] //// In both these example, it will find input text box with accesskey='F' or @id='searchInput'. If any one found then it will locate it. Very useful when elements appears alternatively.

13. Using wildcard * with to finding element xpath
xpath=//*[@accesskey='F']

14. Finding nth child element of parent
xpath=//body/*[3]/form/fieldset/*[2] ////This xpath is for search text box. Here, /*[3] describes the 3rd child element of body which is div[3]. Same way *[2] describes the 2nd child element of fieldset which is input[2]

All above examples are for input text box. Now let me write Xpath for drop down.

Xpath Examples for drop down
1. xpath=//body/div[3]/form/fieldset/select
2. xpath=//body/div[3]/form/fieldset/select[last()]
3. xpath=//body/div[3]/form/fieldset/select[@id='searchLanguage']
4. xpath=//body/div[3]/form/fieldset/select[@name='language']
5. xpath=//div[@class='search-container']/descendant::select[@name='language']
6. xpath=//select[contains(@id, "searchLanguage")]
7. xpath=//div[@class='search-container']/descendant::select[position()=1]
8. xpath=//body/div[3]/form/fieldset/select[count(*)>1]

New Test		
Command	**Target**	**Value**

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...

open	http://www.wikipedia.org/	
select	xpath=//div[@class='search-container']/descendant::select[position()=1]	label=English

Other Xpath Example

1. Finding xpath for target link 'url'

//a[@href='//meta.wikimedia.org/wiki/List_of_Wikipedias'] ////This xpath example will find link with given URL (//meta.wikimedia.org/wiki/List_of_Wikipedias) on the page.

2. Finding xpath of element with no child

xpath=//img[count(*)=0] ////This xpath is for wikipedia text logo which is display on top of the page. This xpath will find that image element which have not any child element. Here image node is last and it has not any child element.

xpath=//div[2]/descendant::img[count(*)=0] //// This xpath is for wikipedia logo image which is display under logo text.

Reference Blog http://selenium-suresh.blogspot.in

By Tushar Singhal at June 09, 2017 No comments: Links to this post

WEDNESDAY, 15 JUNE 2016

Selenium WebDriver Interview Questions With Answers - Part 1

Selenium WebDriver Interview Question with Answer - Part 1

1. How will you find an element using Selenium?

There are different ways to find an element in a web page they are

- ID
- Name
- Tag
- Attribute
- CSS
- Linktext
- PartialLink Text
- Xpath etc

2. Explain the difference between single and double slash in X-path?

Single slash '/ '

- Single slash (/) start selection from the document node
- It allows you to create 'absolute' path expressions

Double Slash '// '

- Double slash (//) start selection matching anywhere in the document
- It enables to create 'relative' path expressions

3. List out the technical challenges with Selenium?

Technical challenges with Selenium are

- Selenium supports only web based applications
- It does not support the Bitmap comparison
- For any reporting related capabilities have to depend on third party tools
- No vendor support for tool compared to commercial tools like HP UFT
- As there is no object repository concept in Selenium, maintainability of objects becomes difficult

4. What are the features of TestNG and list some of the functionality in TestNG which makes it more effective?

TestNG is a testing framework based on JUnit and NUnit to simplify a broad range of testing needs, from unit testing to integration testing. And the functionality which makes it efficient testing framework are

- Support for annotations
- Support for data-driven testing
- Flexible test configuration
- Ability to re-execute failed test cases

5. Explain what is the difference between find elements () and find element () ?

find element ():

It finds the first element within the current page using the given "locating mechanism". It returns a single WebElement

findElements () : Using the given "locating mechanism" find all the elements within the current page. It returns a list of web elements.

6. How do I launch the browser using WebDriver?

The following syntax can be used to launch Browser:

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

7. What are the different types of waits available in WebDriver?

There are two types of waits available in WebDriver:

1. Implicit Wait
2. Explicit Wait

Implicit Wait: Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

Explicit Wait: Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

8. How can you find if an element in displayed on the screen?
WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.
1. isDisplayed()
2. isSelected()
3. isEnabled()
Syntax:
isDisplayed():
boolean buttonPresence = driver.findElement(By.id("gbqfba")).isDisplayed();
isSelected():
boolean buttonSelected = driver.findElement(By.id("gbqfba")).isDisplayed();
isEnabled():
boolean searchIconEnabled = driver.findElement(By.id("gbqfb")).isEnabled();

9. How to select value in a dropdown?
Value in the drop down can be selected using WebDriver's Select class.
Syntax:
selectByValue:
Select selectByValue = newSelect(driver.findElement(By.id("SelectID_One")));
selectByValue.selectByValue("greenvalue");
selectByVisibleText:
Select selectByVisibleText = new Select (driver.findElement(By.id("SelectID_Two")));
selectByVisibleText.selectByVisibleText("Lime");
selectByIndex:
Select selectByIndex = newSelect(driver.findElement(By.id("SelectID_Three")));
selectByIndex.selectByIndex(2);

10. What are the different types of navigation commands?
Following are the navigation commands:
navigate().back() – The above command requires no parameters and takes back the user to the previous webpage in the web browser's history.
Sample code:

driver.navigate().back();
navigate().forward() – This command lets the user to navigate to the next web page with reference to the browser's history.
Sample code:
driver.navigate().forward();
navigate().refresh() – This command lets the user to refresh the current web page there by reloading all the web elements.
Sample code:
driver.navigate().refresh();
navigate().to() – This command lets the user to launch a new web browser window and navigate to the specified URL.
Sample code:
driver.navigate().to("https://google.com");

http://essentialsoftwaretesting.blogspot.com

By Tushar Singhal at June 15, 2016 4 comments:

TUESDAY, 10 MAY 2016

How to Open a New Tab in Browser Using Selenium WebDriver with Java

Quite often you may want to open a new tab in the same browser window that is running your Selenium WebDriver tests. Instead of opening a new browser, you can simply use the code below to open a new tab in the same browser:

driver.findElement(By.cssSelector("body")).sendKeys(Keys.CONTROL +"t");
Then when you open a new tab, you have to switch to it to be able to work with the newly opened tab:

ArrayList<String> tabs = new ArrayList<String>(driver.getWindowHandles());
driver.switchTo().window(tabs.get(0));

The above code works for Firefox Browser.

By Tushar Singhal at May 10, 2016 1 comment:

Eclipse IDE Shortcut Keys

I am sharing some Eclipse IDE shortcut Keys which really helpful when we do code.

Keys mentioned below :

1. **CTRL + SHIFT + O** : This key automatically imports all statement which required to Run Code successfully.

2. **CTRL + SHIFT + X** : This key convert selected string into UPPER CASE.

3. **CTRL + SHIFT + Y** : This key convert selected string into lower case.

4. **CTRL + L** : It help to jump on any line in selected code file.

5. **CTRL + O** : It will give you list of all constructors, variables and methods in selected file, you can directly jump on it by select.

6. **CTRL + =** : Zoom In

7. **CTRL + -** : Zoom out

8. **CTRL + ALT + S** : Synchronize with Repository

9. **CTRL + ALT + D** : Update to Revision

10. Most Used Key : If we open multiple tabs in eclipse and we frequently want to open different tabs then generally we click on tab by mouse but It's time-consuming.
You can create your own short cut key like "**CTRL + TAB**" which same works like as windows.

Steps to implement it(Only one time) :

1. Open Eclipse
2. Click on Preferences under window
3. Click on General
4. Click on Keys
5. Type Next tab on Blank Text Box
6. Automatically, System shows Next Tab
7. Click on it then after Click on Binding
8. then Click "CTRL + TAB"
9. Click on Apply and Save

Note : These Keys will not work on all versions of Eclipse

By Tushar Singhal at May 10, 2016 7 comments:

Newer Posts Home

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