SELENIUM LOCATORS COMMANDS

Lecture Notes

A COMPLETE SELENIUM GUIDE

Softech Solutions Inc.

www.softechsolutionsgroup.com

saney.alam@softechsolutionsgroup.com

Selenium Locators Commands

In the previous chapter of WebElement Commands, we learned different types to actions which can be performed on a ***WebElement object***. Thus the next thing to do is to interact with a web page to use ***WebElement Commands/Actions***. First thing to locate an element on the web page before interacting with it and locating elements can be done on the **WebDriver Instance (driver)** itself or on a **WebElement**. WebDriver gives us **Find Element and Find Elements** methods to locate element on the web page.

As in the previous chapters we learned that every method of the *WebDriver* either returns something or return void (***means return nothing***). The same way **findElement** method of **WebDriver** returns **WebElement**.

But the ***findElement()*** method accepts something as a Parameter/Argument and which is **By Object**. **By** is the mechanism used to locate elements within a document with the help of locator value. A normal syntax of **By** looks like this:

```
WebElement UserName = driver.findElement(By.id("UserName"));

// This can be also written as

By locator = By.id("UserName");
WebElement UserName = driver.findElement(locator);
```

Locating Element using By Strategy

Locating elements in WebDriver is done by using the **findElement (By.locator())** method. The **findElement** methods take a locator or query object called '**By'**. In the eclipse code window type **driver.findElement(By dot)**, Eclipse intelligence will populate the list of different locators. '**By'** strategies are listed below.

Browser tools for Element Inspector

- **Firefox*: Firebug add on. Right click on any element and select Inspect Element or F12***
- **Chrome*: Build in Page analyzing feature (right click –> Inspect Element / F12)***
- **IE: *Developers Tool (Tools –> Developers Tools/ F12)***

I would suggest to take a look at the small chapter of **Finding Elements using Browser Inspector** before moving on to this chapter**.**

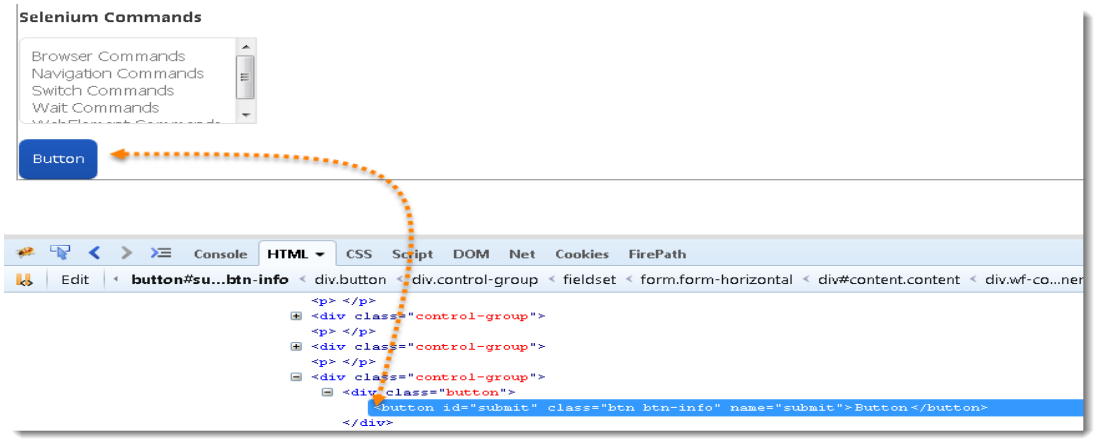
By ID

id(String id) : By – This is the most efficient and preferred way to locate an element, as most of the times IDs are unique. It takes a parameter of String which is a Value of ID attribute and it returns a BY object to findElement() method.

Command – **driver.findElement(By.id("Element ID"));**

With this strategy, If no element has a matching id attribute, a NoSuchElementException will be raised.

Example: If an element is given like this:



Actual Command

```
WebElement element = driver.findElement(By.id("submit"));
// Action can be performed on Input Button element
element.submit();
```

Note: *Common pitfalls that UI developers make is having non-unique id's on a page or auto-generating the id, both should be avoided.*

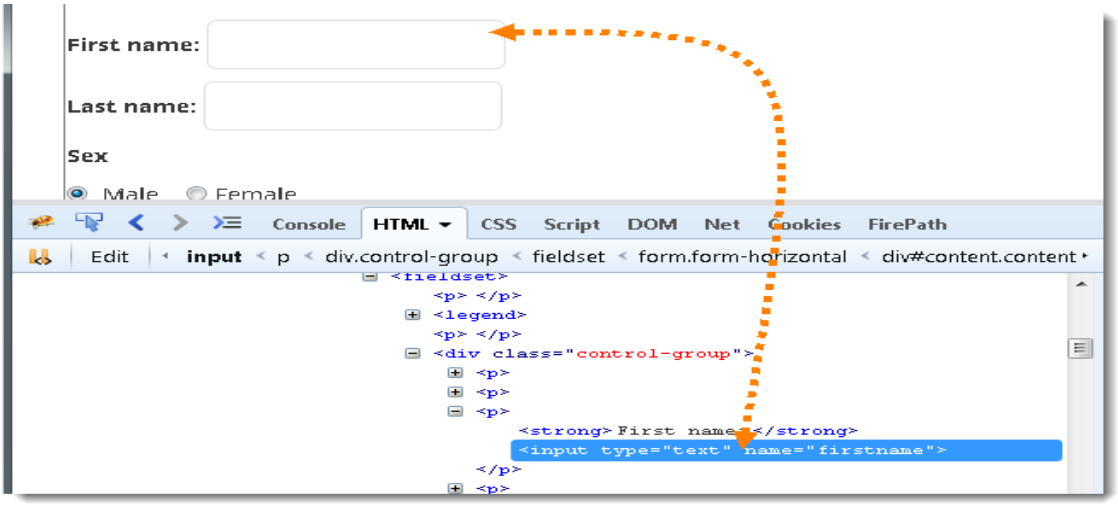
By Name

name(String name) : By – This is also an efficient way to locate an element but again the problem is same as with ID that UI developer make it having non-unique names on a page or auto-generating the names. It takes a parameter of String which is a ***Value of NAME attribute*** and it returns a **BY object** to **findElement()** method.

Command – **driver.findElement(By.name("Element NAME"));**

With this strategy, the first element with the name attribute value matching the location will be returned. If no element has a matching name attribute, a **NoSuchElementException** will be raised.

Example: If an element is given like this:



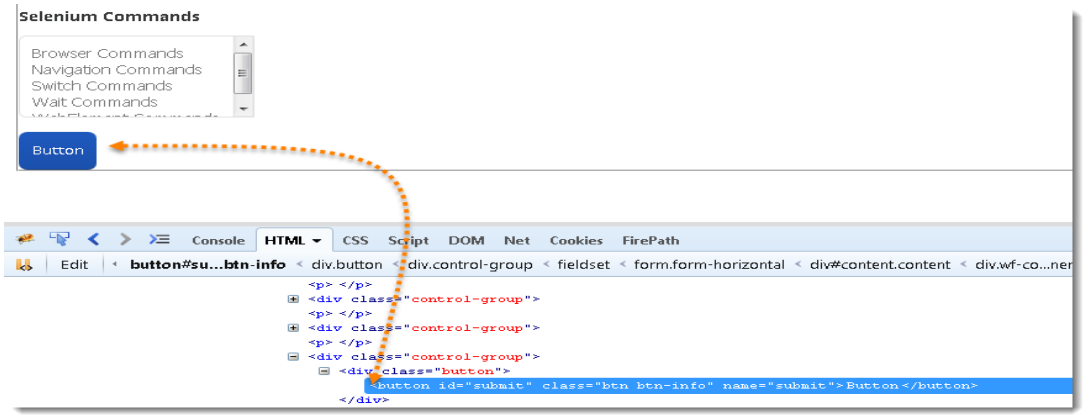
Actual Command

```
WebElement element = driver.findElement(By.name("firstname"));
// Action can be performed on Input Text element
element.sendKeys("Saney");
```

By ClassName

className(String className) : By – This finds elements based on the value of the ***CLASS*** attribute. It takes a parameter of String which is a ***Value of CLASS attribute*** and it returns a **BY object** to **findElement()** method.
Command – **driver.findElement(By.className("Element CLASSNAME"));**
If an element has many classes then this will match against each of them.

Example: If an element is given like this:



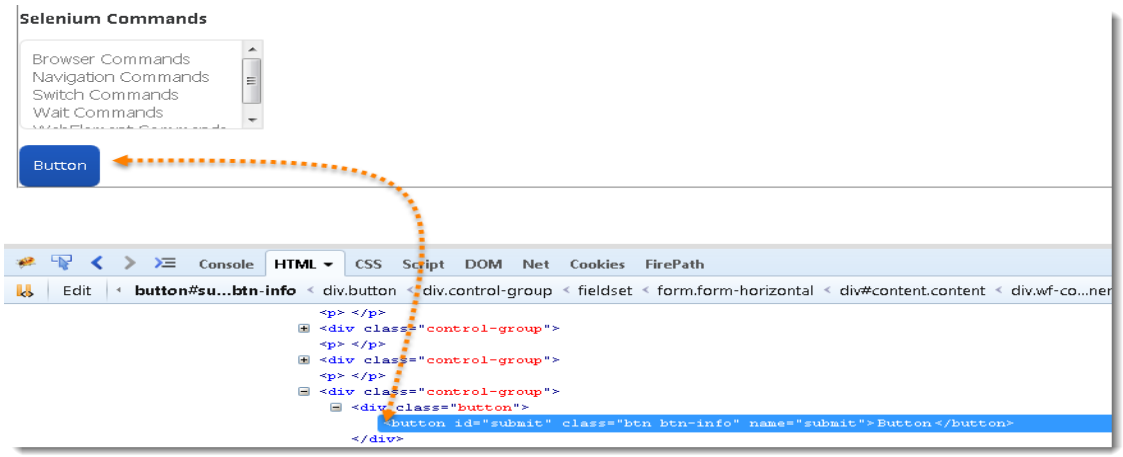
Actual Command

```
WebElement parentElement = driver.findElement(By.className("button"));
WebElement childElement = parentElement.findElement(By.id("submit"));
childElement.submit();
```

Note*: This method is a life saver. As said, class can contain many elements, many times when you end up with duplicate IDs and Names, just go for the ClassName first and try to locate the element with ID. That will work fine, as the selenium will look for the ID which is in the mentioned class.*

By TagName

tagName(String name) : By – With this you can find elements by their ***TAGNAMES***. It takes a parameter of String which is a ***Value of TAG attribute*** and it returns a **BY object** to **findElement()**method.
Command – **driver.findElement(By.tagName("Element TAGNAME"));**
Locating Element By Tag Name is not too much popular because in most of cases, we will have other alternatives of element locators. But yes if there is not any alternative then you can use element's DOM Tag Name to locate that element in ***WebDriver***.
Example: If an element is given like this:



Actual Command

```
WebElement element = driver.findElement(By.tagName("button"));
// Action can be performed on Input Button element
element.submit();
```

By LinkText & PartialLinkText

linkText(String linkText) : By – With this you can find elements of ***"a" tags(*Link*)*** with the link names. Use this when you know link text used within an anchor tag. It takes a parameter of String which is a ***Value of LINKTEXT attribute*** and it returns a **BY object** to **findElement()** method.

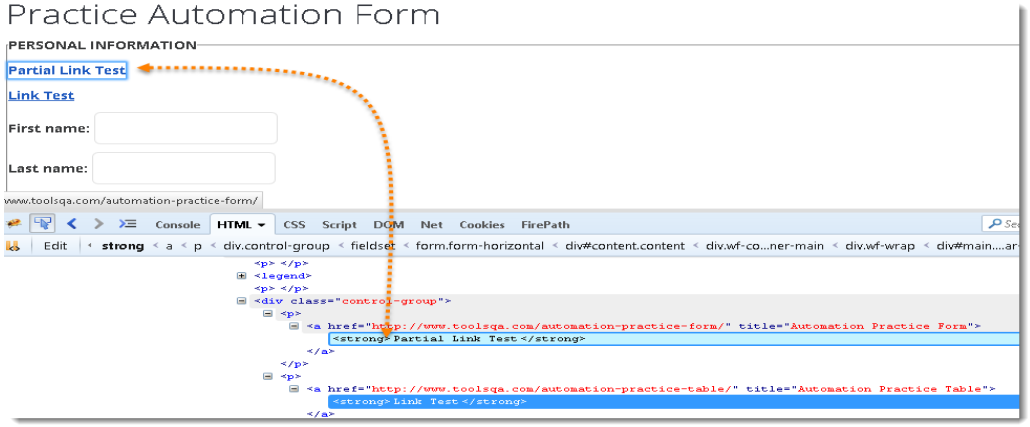
partialLinkText(String linkText) : By – With this you can find elements of ***"a" tags(*Link*)*** with the partial link names.

Command – **driver.findElement(By.linkText("Element LINKTEXT"));**

Command – **driver.findElement(By.partialLinkText("Element LINKTEXT"));**

If your targeted element is link text then you can use by link text element locator to locate that element. Partial Link Text is also same as Link text, but in this we can locate element by partial link text too. In that case we need to use **By.partialLinkText** at place of **By.linkText.**

Example: If an element is given like this:



Actual Command

```
WebElement element = driver.findElement(By.linkText("Partial Link Test"));
element.clear();

//Or can be identified as
WebElement element = driver.findElement(By.partialLinkText("Partial");
element.clear();
```

By XPath

xpath(String xpathexpression) : By – It is most popular and majorly used locating element technique or the easiest way to locate element in WebDriver. It takes a parameter of String which is a ***XPATHEXPRESSION*** and it returns a **BY object** to **findElement()** method.

Command – **driver.findElement(By.xpath("Element XPATHEXPRESSION"));**

The best thing in xpath is that it provides many different techniques to locate elements. It gives you feature to locate single element in many ways.

Difference between FindElement & FindElements Commands

The difference between findElement() **and** findElements()**method is the first returns a** *WebElement* **object otherwise it throws an exception and the latter returns a** *List of WebElements*, **it can return an empty list if no DOM elements match the query.**

findElement()

On Zero Match *: throws NoSuchElementException*

On One Match *: returns WebElement*

On One+ Match *: returns the first appearance in DOM* findElements()

On Zero Match *: return an empty list*

On One Match *: returns list of one WebElement only*

***On One+ Match**: returns list with all matching instance*

Summary

Variation	Description	Sample
*By.**className***	finds elements based on the value of the "class" attribute	*driver.findElement(By.className("Element_ClassName"))*
*By.**cssSelector***	finds elements based on the driver's underlying CSS Selector engine	*driver.findElement(By.cssSelector("Element_CssSelecetor"))*
*By.**id***	locates elements by the value of their id attribute	*driver.findElement(By.id("Element_ID"))*
*By.**linkText***	finds a link element by the exact text it displays	*driver.findElement(By.linkText("Element_LinkText"))*
*By.**name***	locates elements by the value of the "name" attribute	*driver.findElement(By.name("Element_Name"))*
*By.**partialLinkText***	locates elements that contain the given link text	*driver.findElement(By.partialLinkText("Element_PartialLinkText"))*
*By.**tagName***	locates elements by their tag name	*driver.findElement(By.tagName("Element_TagName"))*
*By.**xpath***	locates elements by their XPath	*driver.findElement(By.xpath("Element_Xpath"));*