Multiple Ways to Locate Elements

This section will guide you to:

- Locate elements in Multiple ways using selenium web driver

This guide has mainly seven subsections, namely :

Using ID as a Locator

Using class name as a Locator

Using name as a Locator

Using Link Text as a Locator

Using Xpath as a Locator

Using CSS Selector as a Locator

Using XPath for handling complex and dynamic elements

Step Using ID as a Locator

- Open Eclipse
- Find a web element using Locator **ID**
  a. Syntax: id = id of the element
  b. Example: driver.findElement(By.id("Email"));

Step Using class name as a Locator

- Find a web element using Locator **ClassName**
  a. Syntax: class = Class Name of the element
  b. Example: driver.findElement(By.class("classname"));

Step Using Name as a Locator

- Find a web element using Locator **Name**
  a. Syntax: name = Name of the element

b. Example: driver.findElement(By.name(“name”));

Step Using LinkText as a Locator

- Find a web element using Locator **Link Text**
  a. Syntax: link = partialLink of the element
  b. Example: driver.findElement(By.partialLinkText(“plink”));

Step Using Xpath as a Locator

- Find a web element using Locator **Xpath**
- Xpath can be created in two ways
  a. Relative Xpath
    - Syntax: relativeXpath : //*[@class=’relativexapath’]
    - Example: driver.findElement(By.xpath(“//*[@class=’relativexapath’]”));

  b. Absolute Xpath
    - Syntax: absoluteXpath : html/body/div[1]/div[1]/div/h4[1]/b
    - Example: driver.findElement(By.xpath(“html/body/div[1]/div[1]/div/h4[1]/b”));

Step Using Xpath as a **CSS Selector**

- CSS Selector have many formats, namely
  a. Tag and ID
    - Syntax:”css = tag#id”
    - Example: driver.findElement(By.cssSelector(“input#email”));

  b. Tag and Class
    - Syntax: “css = tag.class”
    - Example: driver.findElement(By.cssSelector(”input.inputtext”));

  c. Tag and Attribute
    - Syntax: “css = tag[attribute=value]”
    - Example: driver.findElement(By.cssSelector(“input[name=lastName]”));

  d. Tag, Class, and Attribute
    - Syntax: “tag.class[attribute=value]”
    - Example: driver.findElement(By.cssSelector(“input.inputtext[tabindex=1]”));

  e. Inner text
    - Syntax: ”css = tag.contains(“innertext”)”
    - Example: driver.findElement(By.cssSelector(font:contains(“Boston”)));

Step Using Xpath for handling complex and dynamic elements

- Dynamic Xpath has many formats, namely

a. **Contains();**
   - Syntax: “xpath = //*[contains(text(),’text’)]
   - Example: driver.findElement(By.xpath(”//*[contains(text(),’sub’]”));

b. **Using OR & AND**
   - Syntax: xpath=//*[@type=’submit’ or @name=’btnReset’]
   - Example:

   driver.findElement (By.xpath(”=//*[@type=’submit’ or @name=’btnReset’]”));

c. **Start-with function**
   - Syntax: xpath= //label[starts-with(@id,’message’)]
   - Example:

   driver.findElement (By.xpath(”//label[starts-with(@id,’message’)]”));

d. **Text();**
   - Syntax: xpath=//td[text()=’UserID’]
   - Example: : driver.findElement (By.xpath(”=//td[text()=’UserID’]”));

e. **Following**
   - Syntax: xpath=//*[@type=’text’]//following::input
   - Example: driver.findElement(By.xpath(”=//*[@type=’text’]//following::input”));

f. **Preceding**
   - Syntax: xpath=//*[@type=’text’]//preceding::input
   - Example: driver.findElement(By.xpath(”//*[@type=’text’]//preceding::input”));

g. **Following - sibling**
   - Syntax: xpath=//*[@type=’submit’]//preceding::input
   - Example:

   driver.findElement (By.xpath (”//*[@type=’text’]//following-sibling::input”));