Finding WebElements

Ram Sharma

Which method to use?

- Below are the methods:
  - WebElement findElement(By by)
  - List<WebElement> findElements(By by)
- We can use either WebDriver API or WebElement API to find WebElements
- Means, we can find element using driver and also with respect to another WebElement.
- If searched WebElement is not found.
  - findElement() gives NoSuchElementException
  - findElements() return a List of size 0.

Using className, id, name

- All the above are attribute of any tag.
- It is not compulsory that a tag have any of these mentioned attributes.
- In any professional webpage, id is made as unique.
- Methods used:
  - driver.findElement(By.id("id_Name");
  - driver.findElement(By.className("class_Name");
  - driver.findElement(By.name("name_Name");

Using tagName

- We can also find element using tag name.
- When a user wants to find all the elements with a particular tag name, it can be used.
- If a user want to find all the hyperlinks in an webpage.
  - driver.findElements(By.tagName("a"))
- General syntax:
  - driver.findElement(By.tagName("tag_Name"))

Using linkText and partialLinkText

- It is generally targeted to find the hyperlink
- Text enclosed inside ***a*** tag is found by this method.
- linkText is used to find exact match and partialLinkText is used when part of the searched text is known.
- Syntax:
  - driver.findElement(By.linkText("Hyperlink_name"))
  - driver.findElement(By.partialLinkText("partial_name"))

Using xpath

- xpath is also called as XML path.
- It assumes that webpage is a xml file and start searching for it. It is the slowest among locators.
- They are of two types:
  - Absolute xpath : Searches from root element
    - html/body/section[2]/div/div/div/form/h1
  - Relative xpath : Search wrt a particular element
    - //*[@id='SearchForm']/h1 ----- * means any
- xpath general form
  - //tag_Name[@attribute_Name='value']
  - //tag_Name[text()='searched_text']

xpath helpers

- contains method
- starts-with method
- not method
- and & or
- xpath axes : axes::tagName
  - child : child element.(/)
  - parent : parent element.(..)
  - following : elements following to it.
  - preceding : elements preceding to it.
  - following-sibling : sibling element following to it.
  - preceding-sibling : sibling element preceding it.
  - descendant : n-th child element in hierarchy.(//)
  - ancestor : n-th parent element in hierarchy.

xpath positions

- position for xpath and cssSelector starts from 1.
- If there are multiple matches of an xpath expression, we can distinguish them by their position.
- Syntax:
  - [1] or [position()=1] : first position
  - [5] or [position()=5] : fifth position
  - [last()] or [position()=last()] : last position
- Example:
  - "//div[@id='top']/span[1]" : this will return first span of div with specified id as top if multiple span are present.
  - "//div[@name='dress']/span][1]": this will return first span of all div whose name is dress and it may not be unique.
  - "(//div[@name='dress']/span])[1]": this will return the first span tag among all matching span, it will be unique.

Using cssSelector

- CSS is "Cascading Style Sheets“. Based on the css of an element, we will be able to identify it.
- General syntax:
  - tagName[attributename=attributeValue]
  - "input[name='field-keywords']"
- For id attribute:
  - "input[id='twotabsearchtextbox']"
  - "input#twotabsearchtextbox“
- For class attribute:
  - "input[class='nav-input']"
  - "input.nav-input"

cssSelector helpers

- For multiple and conditions – no space allowed
  - "input[class='nav-input'][id='twotabsearchtextbox']“
- value as substring
  - '^' - value starts with — input[id^='ema']
  - '$' - value ends with — input[id$='mail']
  - '*' - value contains — input[id*='mai']
- For child selectors
  - '>' direct child element
  - ' ' descendant child element. (space)
- Accessing sibling - Just the next sibling
  - "div[id=nav-xshop] script:nth-of-type(1) + a"

cssSelector positions

- When multiple elements are found, finding an element wrt to a tag
  - "div[id=nav-xshop] a:nth-child(1)“ ---- discouraged
  - "div[id=nav-xshop] a:nth-of-type(1)"
- When multiple element are found, finding an element irrespective of any tag
  - "div[id=nav-xshop] *:nth-child(1)“
  - "div[id=nav-xshop] *:nth-of-type(1)“ --- discouraged
- In finding element via css selector, tag name is skipped if applicable to any tag
  - "input[id='twotabsearchtextbox']“ ----- input element
  - "[id='twotabsearchtextbox']“ ---------- any element