

Selenium WebDriver JavaScript Executor & Elements

Press Space for next page →



Agenda

1. JavaScript Executor in Selenium WebDriver
2. JavaScript Executor Use Cases
3. Code Example: JavaScript Executor
4. Selenium WebDriver Elements Not Clickable Exception
5. Handling Hidden Elements
6. Code Example: Handling Hidden Elements
7. Code Example: Handling Calendar

JavaScript Executor in Selenium WebDriver

JavaScriptExecutor: An interface in Selenium WebDriver used to execute JavaScript code within the context of the browser.

Purpose: Allows direct interaction with the DOM and browser's JavaScript engine, enabling advanced operations not directly supported by WebDriver.

Key Methods: executeScript(): Executes JavaScript code synchronously

```
1 JavascriptExecutor js = (JavascriptExecutor) driver;  
2 js.executeScript("return document.title;");
```

JavaScript Executor Use Cases

- **Scrolling:** `js.executeScript("window.scrollBy(0,1000);");`
- **Handling Hidden Elements:** `js.executeScript("arguments[0].click();", hiddenElement);`
- **Scroll to an Element:** `js.executeScript("arguments[0].scrollIntoView(true);", element);`
- **Highlight Element:** `js.executeScript("arguments[0].style.border='3px solid red';", element);`

Code Example: JavaScript Executor

```
1  class JavaScriptExecutorExample {  
2      public static void main(String[] args) {  
3          WebDriver driver = new ChromeDriver();  
4          driver.get("http://www.facebook.com");  
5          JavascriptExecutor js = (JavascriptExecutor) driver;  
6  
7          String title = (String) js.executeScript("return document.title;");  
8          System.out.println("Title: " + title);  
9  
10         WebElement username= driver.findElement(By.id("email"));  
11  
12         js.executeScript("arguments[0].setAttribute('style', 'background: yellow; border: 2px solid red;')", username);  
13         Thread.sleep(1000);  
14  
15         driver.quit();  
16     }  
17 }
```

Selenium WebDriver Elements Not Clickable Exception

Problem: Selenium WebDriver throws `ElementNotInteractableException` when trying to interact with an element that is not clickable.

Reasons: Element is present in the DOM but cannot be interacted with. The element is not interactable in the sense it cannot be clicked or send keys cannot work as the element is:

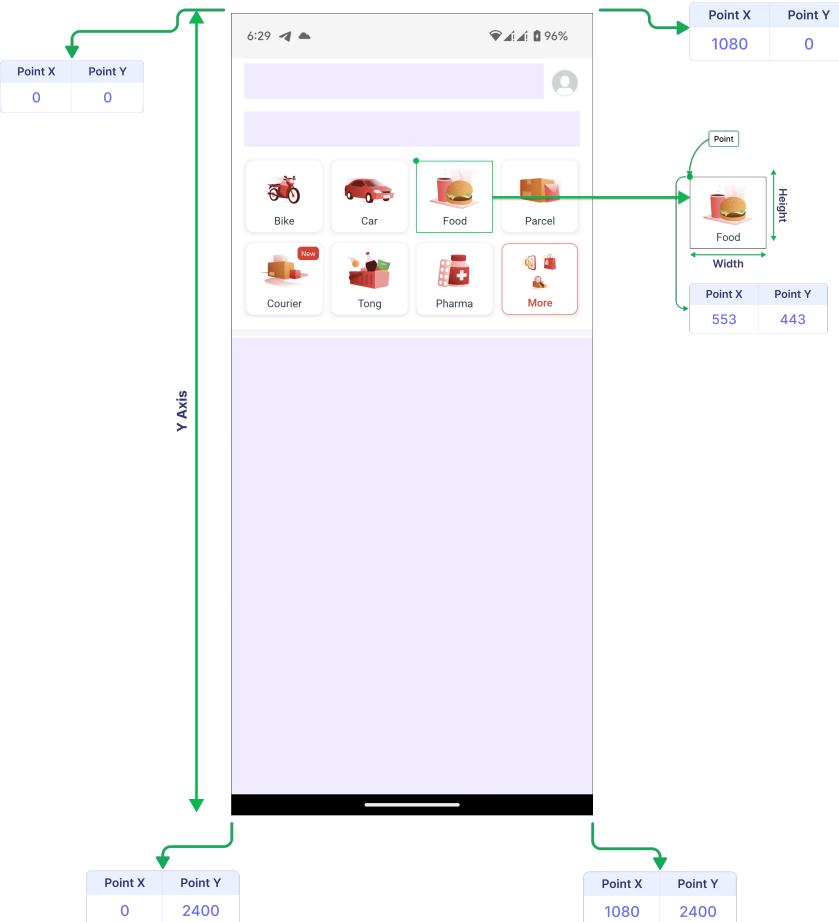
- not visible or hidden
- disabled
- outside the viewport and inaccessible
- overlapped by another element
- not completely rendered on the page

Handling Hidden Elements

Using:

- `getLocation().getX();`
- `getLocation().getY();`

Because Hidden Elements doesn't have X,Y coordinates



Code Example: Handling Hidden Elements

```
1  class HiddenElementExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4          driver.get("http://seleniumpractise.blogspot.in/2016/08/how-to-automate-radio-button-in.html");
5
6          List<WebElement> radio = driver.findElements(By.id("male"));
7          int count = radio.size();
8          System.out.println("Total radio buttons " + count);
9
10         for (int i = 0; i < count; i++) {
11             int x = radio.get(i).getLocation().getX();
12             int y = radio.get(i).getLocation().getY();
13             System.out.println("X:" + x);
14             System.out.println("Y:" + y);
15             if (x != 0) {
16                 radio.get(i).click();
17                 break;
18             }
19         }
20
21         driver.quit();
22     }
23 }
```

Code Example: Handling Calendar

```
1  class CalendarExample {
2      public static void main(String[] args) {
3          WebDriver driver = new ChromeDriver();
4          driver.get("https://www.globalsqa.com/demo-site/datepicker/");
5
6          driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@class='demo-frame lazyloaded']")));
7          driver.findElement(By.xpath("//input[@class='hasDatepicker']")).click();
8          List<WebElement> dates = driver.findElements(By.xpath("//table[@class='ui-datepicker-calendar']//td"));
9
10         int total_nodes = dates.size();
11         System.out.println("Total nodes: " + total_nodes);
12         for(int i=0; i<total_nodes; i++) {
13             String date = dates.get(i).getText();
14             if(date.equalsIgnoreCase("20")) {
15                 dates.get(i).click();
16                 break;
17             }
18         }
19
20         driver.quit();
21     }
22 }
```

Thank you ❤

QA June 2024 Automation with Java Slides