Scroll Bar

What is Scroll Bar?

- Scroll bar is let's you move around screen in horizontal and vertical direction and it is used to move the window up and down
- In Selenium certain web pages will visible once the user have scrolled to the web element
- In such cases we have to use scrollbar.
- We can handle this scrollbar using JavaScript executor interface ,with the help of executeScript() abstract method.

How to use JavaScriptExecutor in Selenium

- Here is a step-by-step process on how to use JavaScriptExecutor in Selenium:

Step 1) Import the package.

import org.openqa.selenium.JavascriptExecutor;

Step 2) Create a Reference.(downcasting statement)

JavascriptExecutor js = (JavascriptExecutor) driver;

Step 3) Call the JavascriptExecutor method.

js.executeScript(script, args);

Handling scroll bar with the help of JavaScriptExecutor and getLocation() method

- First find the x and y co ordinates with the help of getLocation().

  //address of the element
- Webelement ele= driver.findElement(By.xpath(“ ”));

  //downcasting to javaScriptExecutor interface
- JavaScriptExecutor js=(JavaScriptExecutor)driver;
- Js.executrScript(“window.scrollBy(“+x+”,”+y+”)”);

- Window()----->available in options nested interface.

Without using getLocation()

  //address of the element
- Webelement ele= driver.findElement(By.xpath(“ ”));

  //downcasting to javaScriptExecutor interface
- JavaScriptExecutor js=(JavaScriptExecutor)driver;

- Js.executrScript(“arguments[0].scrollIntoView();”,ele);

Script:

```
package ScrollBy;

import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Test1 {

	public static void main(String[] args)
	{
		WebDriver driver = new ChromeDriver();
		driver.manage().window().maximize();
		driver.get("https://www.amazon.com/");

	driver.manage().timeouts().implicitlyWait(Duration
.ofSeconds(10));
```

```
		WebElement ele =
driver.findElement(By.xpath("//img[@alt='Shop
Laptops & Tablets']"));
		Point l = ele.getLocation();
		int x = l.getX();
		int y = l.getY();

		JavascriptExecutor js = (JavascriptExecutor)
driver;

	js.executeScript("window.scrollBy("+x+","+y+")");
		ele.click();
	}

}
```

Script:
Open browser enter the URL https://demoapp.skillrary.com/product.php?product=selenium-training scroll down till carrer and click on it and close.

```
package Scroll_Bar;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import io.github.bonigarcia.wdm.WebDriverManager;

public class Scroll_Skillrary_Carrers {

     public static void main(String[] args) throws Throwable {
          WebDriverManager.chromedriver().setup();
          WebDriver driver = new ChromeDriver();
          driver.manage().window().maximize();

     driver.get("https://demoapp.skillrary.com/product.php?pro
duct=selenium-training");
          driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);
          WebElement career =
driver.findElement(By.xpath("//a[text()='Career']"));
          Point ele = career.getLocation();
          int x = ele.getX();
          int y = ele.getY();

          JavascriptExecutor js = (JavascriptExecutor) driver;
          js.executeScript("window.scrollBy(" + x + "," + y +
")");
          career.click();

          Thread.sleep(5000);
          driver.close();

     }

}
```

ScreenShots:

- Inorder to take screen shot selenium takes help of 3rd party tool called Apache commons.Io
- We have to do Type casting from WebDriver interface to TakesScreenshot interface
- So that, we can access getScreenshotAS method from TakesScreenshot interface .
- This will take the photo and stores in Ram .
- So; we will specify required position to store the screenshot

- Inorder to perform this task we have to call copyFile() from FileUtils class.

```
                    //Typcasting
TakesScreenshot ts=(TakesScreenshot) driver;

 //we access the method and that stores SS in RAM
 File src = ts.getScreenshotAs(OutputType.FILE);
 //i have specify the location
 File dest = new File("./Screen-Shots/amazon.png");
//copy paste from RAM to required location
FileUtils.copyFile(src, dest);
```