**Windows and Frames**

**Task Description:**

**1) Write a Selenium script to automate the following task:**

1. Open a new instance of the Chrome/Firefox/Safari browser.
2. Navigate to the URL "https://the-internet.herokuapp.com/iframe".
3. Switch to the iframe using the iframe using css Selector or Xpath.
4. Locate the "p" tag inside the iframe and write the text "Hello People".
5. Close the browser instance

|  |
| --- |
| **package** trainingtaskcompletion;  **import** java.io.File;  **import** org.apache.commons.io.FileUtils;  **import** org.openqa.selenium.By;  **import** org.openqa.selenium.OutputType;  **import** org.openqa.selenium.TakesScreenshot;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.WebElement;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** IframeTest {  **public** **static** **void** main(String[] args) {  // Initialize ChromeDriver  WebDriver driver = **new** ChromeDriver();  // Navigate to the URL  driver.get("https://the-internet.herokuapp.com/iframe");  // Maximize the browser window  driver.manage().window().maximize();  // Switch to the iframe using css Selector or Xpath  driver.switchTo().frame(driver.findElement(By.*cssSelector*("#mce\_0\_ifr"))); // Assuming the iframe has  // id="mce\_0\_ifr"  // Locate the "p" tag inside the iframe and write the text "Hello People"  WebElement paragraph = driver.findElement(By.*tagName*("p"));  paragraph.clear(); // Clear existing text  paragraph.sendKeys("Hello People");  // Take screenshot  File screenshotFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.***FILE***);  **try** {  FileUtils.*copyFile*(screenshotFile, **new** File("D:\\New Bitmap image.bmp"));  } **catch** (Exception e) {  e.printStackTrace();  }  // Wait for a while (you might need to adjust this depending on your system  // speed)  **try** {  Thread.*sleep*(2000);  } **catch** (InterruptedException e) {  e.printStackTrace();  }  // Switch back to the main content  driver.switchTo().defaultContent();  // Close the browser instance  driver.quit();  }  } |

***Output:-***

|  |
| --- |
| C:\Users\Admin\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\New Bitmap image.jpeg |

**2. Write a Selenium script to automate the following task:**

1. Open a new instance of the Chrome/Firefox/Safari browser.
2. Navigate to the URL "https://the-internet.herokuapp.com/windows".
3. Click the "Click Here" button to open a new window.
4. Switch to the newly opened window.
5. Verify that the text "New Window" is present on the page.
6. Close the new window.
7. Verify that the original window is active.
8. Close the browser instance.

|  |
| --- |
| **package** trainingtaskcompletion;  **import** org.openqa.selenium.By;  **import** org.openqa.selenium.OutputType;  **import** org.openqa.selenium.TakesScreenshot;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.WebElement;  **import** org.openqa.selenium.chrome.ChromeDriver;  **import** java.io.File;  **import** java.io.IOException;  **import** java.util.Set;  **import** org.apache.commons.io.FileUtils;  **public** **class** WindowTest {  **public** **static** **void** main(String[] args) {  // Choose the browser to use (uncomment the appropriate line)  WebDriver driver = **new** ChromeDriver();  // Maximize the browser window  driver.manage().window().maximize();  // Navigate to the URL  driver.get("https://the-internet.herokuapp.com/windows");  // Click the "Click Here" button to open a new window  WebElement clickHereButton = driver.findElement(By.*linkText*("Click Here"));  clickHereButton.click();  // Get the handles of all windows  Set<String> windowHandles = driver.getWindowHandles();  // Switch to the newly opened window  **for** (String handle : windowHandles) {  driver.switchTo().window(handle);  }  // Verify that the text "New Window" is present on the page  String newWindowText = driver.findElement(By.*tagName*("h3")).getText();  **if** (newWindowText.equals("New Window")) {  System.***out***.println("New Window text verified");  } **else** {  System.***out***.println("New Window text verification failed");  }  // Take screenshot before closing the new window  *takeScreenshot*(driver, "D:\\Source.bmp");  // Close the new window  driver.close();  // Switch back to the original window  driver.switchTo().window((String) windowHandles.toArray()[0]);  // Verify that the original window is active  String originalWindowText = driver.findElement(By.*tagName*("h3")).getText();  **if** (originalWindowText.equals("Opening a new window")) {  System.***out***.println("Original Window is active");  } **else** {  System.***out***.println("Original Window is not active");  }  // Take screenshot before closing the browser instance  *takeScreenshot*(driver, "D:\\Destination.bmp");  // Close the browser instance  driver.quit();  }  **private** **static** **void** takeScreenshot(WebDriver driver, String fileName) {  File screenshotFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.***FILE***);  **try** {  FileUtils.*copyFile*(screenshotFile, **new** File(fileName + ".png"));  System.***out***.println("Screenshot taken before closing: " + fileName);  } **catch** (IOException e) {  e.printStackTrace();  }  }  } |

***Output:-***

|  |
| --- |
| C:\Users\Admin\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Source.bmp.jpeg  C:\Users\Admin\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Destination.bmp.jpeg |

**3. Write a Selenium Java code to automate the following scenario:**

* Open the URL http://the-internet.herokuapp.com/nested\_frames
* Use only css selector or Xpath.
* Switch to the top frame.
* Verify that there are three frames on the page.
* Switch to the left frame.
* Verify that the left frame has a text "LEFT".
* Switch back to the top frame.
* Switch to the middle frame.
* Verify that the middle frame has a text "MIDDLE".
* Switch back to the top frame.
* Switch to the right frame.
* Verify that the right frame has a text "RIGHT".
* Switch back to the top frame.
* Switch to the bottom frame.
* Verify that the bottom frame has a text "BOTTOM".
* Switch back to the top frame.
* Verify that the page title is "Frames".

|  |
| --- |
| **package** trainingtaskcompletion;  **import** org.openqa.selenium.By;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** NestedFrameTest {  **public** **static** **void** main(String[] args) {  WebDriver driver = **new** ChromeDriver();  driver.manage().window().maximize();  driver.get("http://the-internet.herokuapp.com/nested\_frames");  // Switch to the top frame  driver.switchTo().frame(driver.findElement(By.*cssSelector*("frame[name='frame-top']")));  // Verify that there are three frames on the page  **int** frameCount = driver.findElements(By.*cssSelector*("frame")).size();  **if** (frameCount == 3) {  System.***out***.println("There are three frames on the page.");  } **else** {  System.***out***.println("Frame count verification failed.");  }  // Switch to the left frame  driver.switchTo().frame(driver.findElement(By.*cssSelector*("frame[name='frame-left']")));  // Verify that the left frame has a text "LEFT"  String leftFrameText = driver.findElement(By.*tagName*("body")).getText();  **if** (leftFrameText.equals("LEFT")) {  System.***out***.println("LEFT frame text verified.");  } **else** {  System.***out***.println("LEFT frame text verification failed.");  }  // Switch back to the top frame  driver.switchTo().defaultContent();  // Switch to the middle frame  driver.switchTo().frame(driver.findElement(By.*cssSelector*("frame[name='frame-top']")));  // Switch to the middle frame  driver.switchTo().frame(driver.findElement(By.*cssSelector*("frame[name='frame-middle']")));  // Verify that the middle frame has a text "MIDDLE"  String middleFrameText = driver.findElement(By.*tagName*("body")).getText();  **if** (middleFrameText.equals("MIDDLE")) {  System.***out***.println("MIDDLE frame text verified.");  } **else** {  System.***out***.println("MIDDLE frame text verification failed.");  }  // Switch back to the top frame  driver.switchTo().defaultContent();  // Switch to the right frame  driver.switchTo().frame(driver.findElement(By.*cssSelector*("frame[name='frame-top']")));  // Switch to the right frame  driver.switchTo().frame(driver.findElement(By.*cssSelector*("frame[name='frame-right']")));  // Verify that the right frame has a text "RIGHT"  String rightFrameText = driver.findElement(By.*tagName*("body")).getText();  **if** (rightFrameText.equals("RIGHT")) {  System.***out***.println("RIGHT frame text verified.");  } **else** {  System.***out***.println("RIGHT frame text verification failed.");  }  // Switch back to the top frame  driver.switchTo().defaultContent();  // Switch to the bottom frame  driver.switchTo().frame(driver.findElement(By.*cssSelector*("frame[name='frame-bottom']")));  // Verify that the bottom frame has a text "BOTTOM"  String bottomFrameText = driver.findElement(By.*tagName*("body")).getText();  **if** (bottomFrameText.equals("BOTTOM")) {  System.***out***.println("BOTTOM frame text verified.");  } **else** {  System.***out***.println("BOTTOM frame text verification failed.");  }  // Switch back to the top frame  driver.switchTo().defaultContent();  // Verify that the page title is "Frames"  String pageTitle = driver.getTitle();  **if** (pageTitle.equals("Frames")) {  System.***out***.println("Page title verification successful: Frames");  } **else** {  System.***out***.println("Page title verification failed.");  }  driver.quit();  }  } |

***Output:-***

|  |
| --- |
|  |