24-FEB-2020

-----------

AUTHENTICATION POPUPS

---------------------

Note: Some applications will provide authentication popups during navigating URL to handle authenticate popups we should provide userID & pwd values in get() method along with application URL.

Syntax: driver.get(http:username:password@url);

// To handle Authenticate popup

driver.get("http://admin:admin@the-internet.herokuapp.com/basic\_auth");

------------------------------------------------------------------------------------------------------------------------------------------------------------------

WORKING ON FRAMES

-----------------

Frame is a kind of html document which is embedded on another html doc (i.e. page in a page)

--> Advantage of Frames is without any impact on entire page code we can update that particular Frame in a page.

-----------------------------------

\*\*\* Frame tagname will be frame/iframe

------------------------------------

I.Q. How to know webElement present in a Frame or not?

Procedure-1

-----------

Open application in FF browser

--> right click on webelement, if we get "This frame" option then webelement present in a Frame

(or)

--> Procedure-2

------------

--> If that webElement exist under "frame" or "iframe" tage in DOM

--------------------------------------------------------------------------------------------------------------------------------------------------------------

Ex-32

Write script to find number of frames in a page and to read each frame name

// initialize browser

System.setProperty("webdriver.chrome.driver", "D:\\SelResource\\chromedriver.exe");

WebDriver driver= new ChromeDriver();

driver.get("https://www.selenium.dev/selenium/docs/api/java/index.html");

driver.manage().window().maximize();

//to find no. of frames in a page

List<WebElement> frames=driver.findElements(By.tagName("frame"));

System.out.println("Number of Frames are: "+frames.size());

for(WebElement frame: frames) {

System.out.println(frame.getAttribute("name"));

}

-----------------------------------------------------------------------------------------------------------------------------------------------------

--> Frames doesn't have visible text

--> Should not use gettext() method

--> Should use getAttribute() method

Note: In order to get no of count need to use "tagname"

-----------------------------------------------------------------------------------------------------------------------------------------------------

Q. How to work on WebElement which exist in a Frame?

A. First we need to switch to that Frame then we can specify operation on webElement within that frame.

Syntax: {to switch to Frame}

driver.switchTo().frame(name/ID/index);

where as index starts with zero

-----------------------------------------------------------------------------------------------------------------------------------------------------

Ex-33

write script to click on "org.openqa.selenium" link which is exist in a frame

// initialize browser

System.setProperty("webdriver.chrome.driver", "D:\\SelResource\\chromedriver.exe");

WebDriver driver= new ChromeDriver();

driver.get("https://www.selenium.dev/selenium/docs/api/java/index.html");

driver.manage().window().maximize();

//to switch to frame

driver.switchTo().frame("classFrame");

//to click on link "org.openqa.selenium"

driver.findElement(By.linkText("org.openqa.selenium")).click();

-----------------------------------------------------------------------------------------------------------------------------------------------------

Ex-34

Develop ATS to perform login transaction in hdfc bank application

Procedure:

step 1: launch FF browser

Step 2: navigate to url :https://netbanking.hdfcbank.com

Step 3: click on "Continue" button

Step 4: read msg from popup

step 5: close popup

Step 6: enter customer id as "123456"

// initialize browser

System.setProperty("webdriver.chrome.driver", "D:\\SelResource\\chromedriver.exe");

WebDriver driver= new ChromeDriver();

driver.get("https://netbanking.hdfcbank.com");

driver.manage().window().maximize();

driver.switchTo().frame("login\_page");

//to click on "continue"

driver.findElement(By.xpath("//img[@alt='continue']")).click();

//to read msg from popup

String eMsg=driver.switchTo().alert().getText();

System.out.println(eMsg);

//to close popup

driver.switchTo().alert().accept();

//to enter customer id

driver.findElement(By.name("fldLoginUserId")).sendKeys("123456");

-----------------------------------------------------------------------------------------------------------------------------------------------------

switchTo will be used in three concepts:

----------------------------------------

1. Alerts (Java)

2. Frames

3. Window

-----------------------------------------------------------------------------------------------------------------------------------------------------

WORKING WITH MULTIPLE WINDOWS

-----------------------------

To work on multiple windows we need to switch between windows by using window handles

Procedure:

---------

Step 1: First we need to get all the opened window handles using

"getWindowHandles" command

Syntax:

------

Set <String> pageHandles= driver.getWindowHandles();

It will return handles of all opened windows into "Set" class

Step 2:

------

Use Iterator class to read each window handle from "Set" class

Syntax:

------

Iterator <String> winHandles= pageHandles.iterator();

Step-3:

------

To read each window handle value into variable from Iterator class we can use next() method

Syntax:

------

String win1= winHandles.next();

String win2= winHandles.next();

Step 4:

------

To work on specified window we can use following command:

driver.switchTo().window(window handle);

Note:

----

If we want to focus on first window/default frame, then we can use below command

driver.switchTo().defaultContent();

----------------------------------------------------------------------------------------------------------------------------------------------------

Ex-35

write script to read each window title while navigating multiple windows in Gmail

// initialize browser

System.setProperty("webdriver.chrome.driver", "D:\\SelResource\\chromedriver.exe");

WebDriver driver= new ChromeDriver();

driver.get("http://gmail.com");

driver.manage().window().maximize();

//to read page title

String pgTitle1=driver.getTitle();

System.out.println(pgTitle1);

//to click on "Help"

driver.findElement(By.linkText("Help")).click();

Thread.sleep(4000);

//to get all the opened window handles

Set<String> pgHandles= driver.getWindowHandles();

//to read values from Set class

Iterator<String> winHandles=pgHandles.iterator();

String w1=winHandles.next();

String w2=winHandles.next();

//to focus on 2nd window

driver.switchTo().window(w2);

String pgTitle2=driver.getTitle();

System.out.println(pgTitle2);

-----------------------------------------------------------------------------------------------------------------------------------------------------

WORKING ON KEYBOARD EVENTS USING SENDKEYS()

-------------------------------------------

Using sendKeys() method we can perform keyboard operations on specified webelement in a page.

Syntax:

------

obj.sendkeys(Keys.event);

Note:

----

In general sendKeys() will help to work on AJAX controls

-----------------------------------------------------------------------------------------------------------------------------------------------------

\*\*\*\* Ex-36

write script to select 3rd option in Google search after enter

"Selenium" in search box without using locator value for that option

--------------------

I.Q. How to write script without using locator?

Script:

// initialize browser

System.setProperty("webdriver.chrome.driver", "D:\\SelResource\\chromedriver.exe");

WebDriver driver= new ChromeDriver();

driver.get("http://google.co.in");

driver.manage().window().maximize();

//create reference object for Search editbox

WebElement txtSearch=driver.findElement(By.name("q"));

//to enter Selenium

txtSearch.sendKeys("selenium");

Thread.sleep(4000);

txtSearch.sendKeys(Keys.ARROW\_DOWN);

Thread.sleep(3000);

txtSearch.sendKeys(Keys.ARROW\_DOWN);

Thread.sleep(3000);

txtSearch.sendKeys(Keys.ARROW\_DOWN);

Thread.sleep(3000);

txtSearch.sendKeys(Keys.ENTER);

-------------------------------------------------------------------------------------------------------------------------------------------------

Note: Can't go for Robot class

Disadvantages of using Robot Class from Quora

---------------------------------------------

Here I am giving the drawback of using Robot Class. Which I personally experienced while coding.

When you use Robot class in your code then you have make sure that no other window is open while execution of the code. Because Robot class will work on active window. Your eclipse should be in the background to the active window.

When you use keyPress event of this class then you should use keyRelease event also to perform the short keys on to the browser. e.g you want to click Ctrl+V if you forget to add keyRelease event then it will remain pressed and consume memory in the background. Like this

You need to pay attention on the order of key release. Consider this example. And please try to understand that what I am trying to say

Robot robot = new Robot();

robot.keyPress(KeyEvent.VK\_CONTROL);

robot.keyPress(KeyEvent.VK\_V);

robot.delay(100);

robot.keyRelease(KeyEvent.VK\_V);

robot.keyRelease(KeyEvent.VK\_CONTROL);

Note: Disadvantage of Robot Class instead need to go with the above mentioned example

\*\*\*Keyword/mouse event will only works on current instance of Window. ...

Most of the methods like mouseMove is screen resolution dependent so there might be a chance that code working on one machine might not work on other.

-----------------------------------------------------------------------------------------------------------------------------------------------------

\*\*\* WORKING WITH ACTION CLASS\*\*\*

-------------------------

In Webdriver, to handle keyboard events, mouse events/mouse hover, Drap and Drop, click, double click and Right click operations to perform we can use "Actions"

Class.

Syntax: { to Configure Actions Class}

------

Actions obj= new Actions class;

Ex:

To click on webelement using Actions class:

Procedure:

---------

First we need to specify webelement location using moveToElement() method

--------------

and then specify operation which is needed to perform on that webelement.

Syntax:

------

obj.moveToElement(element).click().build().perform();

Note:

-----

--> The build() method is used to compile all the listed actions into a single step.

--> The perform() method is used to execute the action.

========================================================== END OF CLASS ===========================================================================