

SELENIUM Cheat Sheet

Creating Driver:

Action	Selenium code	Description
Firefox driver	<code>WebDriver Driver = new FirefoxDriver ();</code>	Driver is an object
IE driver	<code>System.setProperty("webdriver.ie. driver", PATH);</code> <code>WebDriver driver = new InternetExplorerDriver ();</code>	PATH = path of IEDriver exe file;
chrome Driver	<code>System.setProperty("webdriver.ie. driver",PATH);</code> <code>WebDriver driver = new ChromeDriver ();</code>	PATH = path of chrome exe file;

Identify Elements:

Action	Code	Description
Find single Element	<code>driver.findElement(locator)</code>	Locator is a location of element
Find multiple Elements	<code>driver.findElements(locator)</code>	Locator is a location of element

Locating UI Elements:

Action	Code	Description
By ID	<code>driver.findElement (By.id(str));</code>	Str is id of element
By Name	<code>driver.findElement(By.name(str));</code>	Str is name of element
By class name	<code>driver.findElement (By.className(str));</code>	Str is class value of element
By CSS selector	<code>driver.findElement(By.cssSelector(str));</code>	Str is cssSelector of element
By link text	<code>driver.findElement(By.linkText(str));</code>	Str is link text of element
By partial link text	<code>driver.findElement(By.partialLinkText(str));</code>	Str is partial text of element
By tag name	<code>driver.findElement(By.tagName(str));</code>	Str is tag name of element
By XPath	<code>driver.findElement(By.xpath(xpath));</code>	Str is xpath of element

Handling Java Script Alerts:

To handle alert first we need to switch to alert.

Alert al=driver.switchTo().alert();

The Actions list.

Action	code
Click on ok in alert	<code>al.accept();</code>
Click on cancel.	<code>al.dismiss()</code>
Type in alert box.	<code>al.sendKeys("text");</code>
Get text from alert box.	<code>al.getText();</code>

Capture Screen Shot of Browser:

Action	Code	Description
Capture screen	<code>File scrFile1 = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);</code>	It captures screen shot of particular page and stores it in variable
Save to disk	<code>FileUtils.copyFile(scrFile1, new File("c:\\tmp\\k2.png"));</code>	Save screen shot as k2.png

User Actions:

Action	Code	Description
Write in text fields	<code>driver.findElement(locator).sendKeys(text);</code>	Text: what u want to write locator is a location element
Click button or click radio button or check box	<code>driver.findElement(locator).click();</code>	locator is a location element
Clear text in text field	<code>driver.findElement(locator).clear();</code>	locator is a location element
Navigate back and forward in browser	<code>driver.navigate().back();</code> <code>driver.navigate().forward();</code>	
Navigate to frame	<code>driver.switchTo().frame(frame);</code>	frame can be integer value represents position of frame or string represents id of frame or WebElement represents frame of frame.
Navigate to next window or pop up window	<code>driver.switchTo().window(hashCode);</code>	hashCode is hash code of window
Get inner text of element or inner text of table	<code>driver.findElement(locator).getText();</code>	locator is a location element
Working on auto complete/suggestions Or Calendar pop up	<code>driver.findElement(locator).click();</code>	Get the locator of hidden division or element and perform required operation.

select drop down list:

Using Select class we can work on select drop down. Create select object for specific select drop down.

//creating webelement for select dropdown

```
WebElement usrs=driver.findElement(By.name("users"));
```

```
Select usr=new Select(usrs);
```

We can select options of drop down in 3 different ways as explained below:

Action	code	description
Select by using id of option tag	<code>usr.selectById(ID);</code>	ID is string, value of id attribute of option.
Select by using index of option tag	<code>usr.selectByIndex(i);</code>	i is the position of option
Select by using visible text in option tag	<code>usr.selectByVisibleText(str)</code>	str is the text inside option tag.

Working on excel sheet:

Before working on excel first we need to read excel in input stream using file io stream.

```
FileInputStream fis=new FileInputStream("Path of .xlsx file");
```

Action	Code	Description
Convert file io into workbook	<code>Workbook wb = WorkbookFactory.create(fis);</code>	Create function creates work book.
Get into specified sheet	<code>Sheet s = wb.getSheet(sheetName);</code> Or <code>Sheet s = wb.getSheetAt(sheetNum);</code>	sheetName is name of the sheet sheetNum is index of sheet
Get into specified row	<code>Row r = s.getRow(rowNum);</code>	
Get into specified column	<code>Cell c = r.getCell(colNum);</code>	
Get cell value	<code>String cellVal = c.getStringCellValue();</code> Or <code>boolean b = c.getBooleanCellValue();</code> or <code>Date d = c.getDateCellValue();</code> Or <code>int l = c.getNumericCellValue();</code>	Get cell value based on value in excel cell
Get row count	<code>int l = s.getLastRowNum();</code>	
Get Column count	<code>int j = r.getLastCellNum();</code>	
Write back to excel	<code>c.setCellValue("PASS1");</code> <code>FileOutputStream fos = new</code> <code>FileOutputStream("Path of .xlsx file ");</code> <code>wb.write(fos);</code> <code>fos.close();</code>	

Drag, Drop and Mouse Over, Mouse Events:

We use Actions Class for drag and drop

Create an object to action class

Actions a=new Actions(driver);

Action	code	description
Drag and Drop using source and destination	<code>a.dragAndDrop(src, des).build().perform();</code>	Src and dest is the WebElement object of source and destination of drag and drop element.
Drag and drop to specific position	<code>a.dragAndDrop(src, x,y).build().perform();</code>	x and y are integer values for specific position.
Mouse over on specific element.	<code>a.moveToElement(element).build().perform();</code>	Element is an object of WebElement which points to required element.
Mouse right click	<code>a.contextClick(element).build().perform();</code>	Element is an object of WebElement which points to required element.
Mouse movement after right click	<code>a.sendKeys(Keys.<keyboardstrokes>).build().perform();</code>	Keys is a class contains all key strokes such as right left, enter, back button.