

Selenium Sudhakar

[Java](#) [SQL](#) [Agile](#) [Scrum](#) [CUCUMBER](#) [Guide](#) [Wire](#)

Browser	Selenium Grid	Day_013_InvocationCount	TestNG Annotations	Day_018_Waits
Get_vs_Navigate	Selenium Limitations	Day_005_TestNG_DataProvider_Lab1	TestNG_Parameters1	Day_019_Frames
Day_004_Dropdown	Selenium Featues	Day_006_TestNG_DataProvider_Lab2	Day_014_015_Parallel_Methods_Tests	Day_020_Multiple_Windows
OR1_applicationProperties	Selenium Challanges	Day_007_Browser_Compatability_Testi	Day_016_ParallelClasses	Day_021_Java_Script_Executer
OR1_Excel	Web Table	Day_025_ExcelUtil_Apache_POI	Day_017_Multiple_Suites	Day_022_Action_class
Xpath_vs_cssSelector	Capcha	Proxy Setting	Day_032_TestNG_Groups	Day_027_Alerts
cssSelector	Assertions	Dimesions	Day_034_TestNG_Listeners	Day_026_TakeScreenShot
Day_024_Advanced_Xpaths	Page Object Model	Keyword Driven	TestNG_Priority	Day_023_Robot_Class
Day_41_Exceptions	Page Factory Model	Interfaces	dependsOnMethods	Day_025_ExcelUtil_Apache_POI
Web Element Border	Web Element Color	Classes in Selenium	BrokenLinks	Emailable Report
Locators in Selenium	FindElement vs Find Elements	Number of Hyperlinks	Extent Report1	SSL Certificate
Page Load Strategy	Relative Locators	Selenium Suite	Chrome Options	Preserve Order
Maven Life Cycle - Pom.xml	HeadLess Browser	is Displayed - is Enabled - is Selected	Navigate_Get	Quit_Close
			Tell me about your self	HeadLess Browser
is Displayed - is Enabled - is Selected	Navigate_Get	Quit_Close		

Exception Handling:

- TC001_NoSuchElementException.java
 - TC002_StaleElementException.java
 - TC003_ElementNotVisibleException.java
 - TC004_NoSuchWindowException.java
 - TC005_NoSuchFrameException.java
 - TC006_NoAlertPresentException.java
 - TC007_InvalidSelectorException.java
 - TC008_ElementNotSelectableException.java
 - TC009_TimeoutException.java
 - TC010_NoSuchSessionException.java
 - TC011_ElementNotInteractableException.java
-

org.openqa.selenium.StaleElementReferenceExcept

stale element reference: element is not attached to the page document

when?

whenever we perform ,getText(),sendKeys,click() operations on a webelement and if that webelement locator is not available in the HTML DOM page,Then system will throw an exception as Stale element Reference Exception

Solution - We have to navigate back to previous state or we have to refresh the page ,so that HTML DOM page will be loaded with the required webelement and we can perform required operation as sendKeys,click, or getText etc...

org.openqa.selenium.NoSuchElementException:

no such element: Unable to locate element:

```
{"method": "xpath", "selector": "//*[@id='txtUsername1']"}
```

if we give invalid locator id,name,xpath ,
System will throw exception as NoSuchElementException

ElementClickInterceptedException , Indicates that a click could not be properly executed

```
WebElement PaginAtion_Link = new WebDriverWait(driver, 20).until(ExpectedConditions.elementToBeClickable(By.xpath(str1)));  
js.executeScript("arguments[0].scrollIntoView();", PaginAtion_Link);  
Thread.sleep(1000);  
js.executeScript("arguments[0].click();", PaginAtion_Link);  
Thread.sleep(4000);
```

org.openqa.selenium.InvalidSelectorException:

invalid selector: Unable to locate an element with the xpath expression `//*[@id='txtUsername']` because of the following

When we will write invalid syntax for xpath

org.openqa.selenium.ElementNotInteractableException:

When the driver can't be scroll into the view , for a particular WebElement

2. Element `<div id="divUsername" class="textInputContainer">`
is not reachable by keyboard

When the locator is not an Input tag and if we perform sendKeys ,
Then System will throw an Exception as ElementNotInteractable Exception

org.openqa.selenium.NoSuchWindowException: no such window

if we are trying to perform , set of driver instructions on the previous window/tab, which was closed already .

```
//below line opens new tab
((JavascriptExecutor)driver).executeScript("window.open()");

// It returns no. of windows opened by WebDriver and will return Set of Strings
Set<String> set =driver.getWindowHandles();

ArrayList<String> tabs = new ArrayList<String> (set);

//Switch to New tab
driver.switchTo().window(tabs.get(1));
driver.get("http://google.com");
driver.close();

//back to MainWindow
driver.switchTo().window(tabs.get(0));
driver.findElement(By.name("txtUsername")).sendKeys("Admin");
driver.findElement(By.name("txtPassword")).sendKeys("admin123");

driver.switchTo().window(tabs.get(1));
```

org.openqa.selenium.NoSuchFrameException:

No frame element found by name or id Google_ContactForm1

When we give invalid Frame name,id System will throw an Exception as NoSuchFrameException

org.openqa.selenium.NoAlertPresentException:

no such alert

When there is no alert in the Webpage , during the execution

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

Timeout Exception :

If the WebElement is not loaded or displayed within the specified time , then System will throw an Exception as Timeout Exception

#7) org.openqa.selenium.ElementNotSelectableException

This exception comes under *InvalidElementStateException* class. *ElementNotSelectableException* indicates that the web element is present in the web page but cannot be selected.

For example, the below code can throw a *ElementNotSelectableException* if the id “swift” is disabled.

```
Select dropdown = new Select(driver.findElement(By.id("swift")));
```

Exception Handling:

```
try {  
    Select dropdown = new Select(driver.findElement(By.id("swift")));  
} catch (ElementNotSelectableException e)
```

In this case, exception is thrown even if the element becomes enabled after a while.

Avoiding-And-Handling: We can add a wait command to wait until the element becomes clickable. If there is still an exception, it is caught.

```
try {  
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));  
    wait.Until(ExpectedConditions.elementToBeClickable(By.id("swift")));  
    try {  
        Select dropdown = new Select(driver.findElement(By.id("swift")));  
    } catch (WebDriverException e) {  
        System.out.println("Exceptional case");  
    }  
    } catch (TimeoutException e)  
    System.out.println("WebDriver found that this element was not selectable.");  
}
```

org.openqa.selenium.NoSuchSessionException: **invalid session id**

```
@Test
public void login_test() throws Exception
{

System.setProperty("webdriver.chrome.driver", "C:\\\\chromedriver_win32\\\\chromedriver.exe");
driver =new ChromeDriver();
driver.manage().window().maximize() ;
driver.get("https://opensource-demo.orangehrmlive.com/");

driver.findElement(By.name("txtUsername")).sendKeys("Admin");
//driver.findElement(By.id("txtUsername")).sendKeys("Admin");
driver.findElement(By.id("txtPassword")).sendKeys("admin123");
driver.findElement(By.id("btnLogin")).click();

driver.close();
driver.findElement(By.name("txtUsername")).sendKeys("Admin");

}
```

After driver session is closed and when we try to execute set of driver instructions , System will throw an Exception as NoSuchSessionException

#6) org.openqa.selenium.ElementNotVisibleException

ElementNotVisibleException class is a subclass of *ElementNotInteractableException* class. This exception is thrown when WebDriver tries to perform an action on an invisible web element, which cannot be interacted with. That is, the web element is in a hidden state.

For example, in the below code, if the type of button with id 'submit' is 'hidden' in HTML, *org.openqa.selenium.ElementNotVisibleException* will be thrown.

```
driver.findElement(By.id("submit")).click();  
Exception Handling:  
try {  
    driver.findElement(By.id("submit")).click();  
} catch (ElementNotVisibleException e)
```

In this case, the exception is thrown even if the page has not loaded completely.

Avoiding-And-Handling: *There are two ways to do this. We can either use wait for the element to get completely.*

The below code waits 10 seconds for the element. If the element is visible and still exception is thrown, it is caught.

```
try {  
    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));  
    wait.Until(ExpectedConditions.visibilityOfElementLocated(By.id("submit")));  
    try {  
        driver.findElement(By.id("submit")).click();  
    } catch (WebDriverException e) {  
        System.out.println("Exceptional case");  
    }  
} catch (TimeoutException e)  
    System.out.println("WebDriver couldn't find this element visible");  
}
```

```
import org.openqa.selenium.ElementClickInterceptedException;
import org.openqa.selenium.ElementNotInteractableException;
import org.openqa.selenium.JavascriptException;
import org.openqa.selenium.InvalidSelectorException;
import org.openqa.selenium.InvalidElementStateException;
import org.openqa.selenium.InvalidCookieDomainException;
import org.openqa.selenium.WebDriverException;
import org.openqa.selenium.ElementNotSelectableException;
import org.openqa.selenium.ElementNotVisibleException;
import org.openqa.selenium.ImeActivationFailedException;
import org.openqa.selenium.ImeNotAvailableException;
import org.openqa.selenium.InvalidArgumentException;

import org.openqa.selenium.NoAlertPresentException;
import org.openqa.selenium.NoSuchContextException;
import org.openqa.selenium.NoSuchCookieException;
import org.openqa.selenium.NoSuchElementException;
import org.openqa.selenium.NotFoundException;
import org.openqa.selenium.NoSuchFrameException;
import org.openqa.selenium.NoSuchSessionException;
import org.openqa.selenium.NoSuchWindowException;

import org.openqa.selenium.ScriptTimeoutException;
import org.openqa.selenium.SessionNotCreatedException;
import org.openqa.selenium.StaleElementReferenceException;

import org.openqa.selenium.TimeoutException;
import org.openqa.selenium.UnableToSetCookieException;
import org.openqa.selenium.UnexpectedAlertBehaviour;
import org.openqa.selenium.UnhandledAlertException;
import org.openqa.selenium.UnsupportedCommandException;
```