

# Difference Between findElement & findElements

## Table of Contents

Introduction .....	2
4 Common Selenium Exceptions.....	2
What Is An Exception.....	2
NoSuchElementException.....	2
NoSuchFrameException.....	2
NoSuchWindowException .....	3
ElementNotVisibleException .....	3

## Introduction

Hello and Welcome. My name is Rex Jones II. Feel free to connect with me on [LinkedIn](#) and subscribe to my [YouTube](#) channel. When you subscribe, click the bell icon to get notification of each video. In this video, we are going to discuss and demo 4 Common Selenium Exceptions.

## 4 Common Selenium Exceptions

The 4 Exceptions are [NoSuchElementException](#), [NoSuchFrameException](#), [NoSuchWindowException](#), and [ElementNotVisibleException](#). There are more exceptions in Selenium such as [NoAlertPresentException](#), [NoSuchSessionException](#), [StaleElementReferenceException](#), [TimeoutException](#), and many more. However, we will demo these 4 Common Exceptions.

## What Is An Exception

What Is An Exception? An Exception is an event that happens at run time. It stops our program from running and does not provide our expected result.

## NoSuchElementException

The first exception is [NoSuchElementException](#). This means, the element we are trying to locate does not exist. There are 3 reasons for a [NoSuchElementException](#). **First**, the element may not have been rendered on the application. This could be a problem with the application or the previous test step failed in our Test Script. **Second**, we could be searching for an element before it is rendered on the application. Selenium is very fast and it's trying to interact with an element before the element shows up. If that's the case, we can add a Wait Statement before searching for the element. **Third**, we are using an incorrect locator or locator value to find the element.

The demo illustrates Number 3. This Test Script has a name locator and q1 as the value. q1 is not correct. The correct value is q but I added 1. If I **execute**, then the program will not send Test into the Search box. Failed – [NoSuchElementException](#). Unable to locate element: {"method": "name", "selector": "q1"}

## NoSuchFrameException

Next, is the [NoSuchFrameException](#). We come across this exception with frames and iframes. There are also 3 possible reasons for a [NoSuchFrameException](#). **Number 1**: We Switch To A Frame That Does Not Exist. This scenario happens when we enter an incorrect index, WebElement, name or id for a frame **Number 2**: Switching To A Frame That's Inside Of Another Frame and **Number 3**: Not Going Back To The Parent Frame Before Switching To A Different Frame. Numbers 2 and 3 can happen if we have more than 1 frame. Imagine we switch from Frame 1 to Frame 2 but want to go back to Frame 1. The exception is thrown if we don't go back to the parent frame using the `switchTo().defaultContent()` method.

Our demo shows `driver.switchTo().frame(34)`; The correct value should be 2 and not 34. Therefore this frame does not exist. **Let's Run**. Failed - [NoSuchFrameException](#)

## NoSuchWindowException

The NoSuchWindowException is next. There are 2 possible reasons for this exception. **First**, switching to a window that no longer exist. This can happen if our automation code does not close a particular window without reloading the list of windows using the getWindowHandles method. **Second**, Switching To A Window Before Invoking getWindowHandles() Method. Let's look at Number 2.

Do you see line 66 the `driver.getWindowHandles()` method? This method gets all of the Window ID's. The code is not correct because the previous line 65 tries to switch to a different window before calling the `driver.getWindowHandles` method. We already have the Main Window ID at line 58 so this code will run without an exception if we remove line 65. Nevertheless, **Let's Run** and see the exception. The Main Window ID is good but our Test Script Failed with NoSuchWindowException

## ElementNotVisibleException

The last of exceptions is ElementNotVisibleException. This exception happens if we try to **Interact With An Element That's Not Visible** and our **Automation Code Missing A Dynamic Wait Statement**. Number 2 goes back to Selenium which runs the Test Script at a fast rate. Let's demo Number 1 on Amazon's site.

Do you see how there's a drop-down list for **Account and Lists**? More options show up after hovering the list. All of these options are elements not visible to Selenium unless we use Selenium's Action Class.

**In Eclipse**, I commented out lines 94 – 98 to hover the drop-down list. Therefore, **when I run**, we should see an ElementNotVisibleException. Our Test Result shows Failed – ElementNotVisibleException.

These 4 Exceptions (NoSuchElementException, NoSucFrameException, NoSuchWindowException, ElementNotVisibleException) are examples of what could happen when writing our Test Scripts for Selenium. Thank You for watching 4 Common Selenium Exceptions.