

<https://www.linkedin.com/in/gayatri-mishra-freelancecoach/>

NoSuchElementException in Selenium :

#SeleniumExceptions

In Selenium , the "NoSuchElementException" is a common exception which occurs when the WebDriver is unable to locate an element in the web page with which it is interacting with. Here are some generic situations where we may get this exception:

Incorrect locator:

If the locator strategy or the locator value used to find an element is incorrect, WebDriver won't be able to locate the element, resulting in a NoSuchElementException. Ensure that you are using the correct locator strategy (such as ID, class name, XPath, CSS selector, etc.) and the correct value for the locator.

Timing issues:

Sometimes, WebDriver may try to locate an element before it is available in the DOM. This can happen when the element is loaded dynamically or takes some time to appear after the page has loaded. In such cases, we may need to wait for the element to be present, visible, clickable, or enabled using explicit or implicit waits.

Element not present on the page:

If the element you are trying to locate is not present in the DOM of the page at all, WebDriver will throw a NoSuchElementException. We must make sure the element exists on the page and is not inside an iframe or a shadow DOM, which WebDriver cannot access directly.

Element not yet loaded:

If the page is still loading or elements are still being rendered when WebDriver tries to locate an element, it may not find it, resulting in a NoSuchElementException. We must ensure that the page has fully loaded before attempting to locate elements.

Dynamic content:

If the content of the page changes dynamically after it has loaded (e.g., through AJAX requests or JavaScript), the element we are trying to locate may no longer be available in the DOM, leading to a NoSuchElementException. We can handle such dynamic content by waiting for specific conditions or events to occur before attempting to locate elements.

Frame/iframe issues:

If the element we are trying to locate is inside an iframe, but WebDriver is not switched to that frame, it won't be able to locate the element, resulting in a NoSuchElementException.

<https://www.linkedin.com/in/gayatri-mishra-freelancecoach/>

We must make sure to switch to the correct frame using WebDriver's "switchTo().frame()" method before locating elements inside frames.

NoSuchElementException in Selenium :

#SeleniumExceptions

In Selenium , the "NoSuchElementException" is a common exception which occurs when the WebDriver is unable to locate an element in the web page with which it is interacting with. Here are some generic situations where we may get this exception:

Incorrect locator:

If the locator strategy or the locator value used to find an element is incorrect, WebDriver won't be able to locate the element, resulting in a NoSuchElementException. Ensure that you are using the correct locator strategy (such as ID, class name, XPath, CSS selector, etc.) and the correct value for the locator.

Timing issues:

Sometimes, WebDriver may try to locate an element before it is available in the DOM. This can happen when the element is loaded dynamically or takes some time to appear after the page has loaded. In such cases, we may need to wait for the element to be present, visible, clickable, or enabled using explicit or implicit waits.

Element not present on the page:

If the element you are trying to locate is not present in the DOM of the page at all, WebDriver will throw a NoSuchElementException. We must make sure the element exists on the page and is not inside an iframe or a shadow DOM, which WebDriver cannot access directly.

Element not yet loaded:

If the page is still loading or elements are still being rendered when WebDriver tries to locate an element, it may not find it, resulting in a NoSuchElementException. We must ensure that the page has fully loaded before attempting to locate elements.

Dynamic content:

If the content of the page changes dynamically after it has loaded (e.g., through AJAX requests or JavaScript), the element we are trying to locate may no longer be available in the DOM, leading to a NoSuchElementException. We can handle such dynamic content by waiting for specific conditions or events to occur before attempting to locate elements.

Frame/iframe issues:

<https://www.linkedin.com/in/gayatri-mishra-freelancecoach/>

If the element we are trying to locate is inside an iframe, but WebDriver is not switched to that frame, it won't be able to locate the element, resulting in a `NoSuchElementException`. We must make sure to switch to the correct frame using WebDriver's `switchTo().frame()` method before locating elements inside frames.

Page structure changes:

If the structure of the page changes (e.g., due to a website update or redesign), the locators we used previously may no longer work, resulting in a `NoSuchElementException`. We must make sure to regularly update your locators to match the current structure of the page.

Page structure changes :

If the structure of the page changes (e.g., due to a website update or redesign), the locators you used previously may no longer work, resulting in a `NoSuchElementException`. We must make sure to regularly update your locators to match the current structure of the page.

`org.openqa.selenium.NoSuchElementException`

Here I attached a sample example script with `NoSuchElementException`

Script :

```
1 package awesomeqa.TaskRegs;
2
3 import java.util.Scanner;
4
5 public class Testcase3 {
6
7     public static void main(String[] args) throws InterruptedException {
8         // TODO Auto-generated method stub
9         // WebDriver instance
10        Scanner c = new Scanner(System.in);
11        WebDriver wl = new FirefoxDriver();
12
13        // launch the application
14        wl.get("https://awesomeqa.com/ui/index.php?route=account/login");
15        //wl.findElement(By.cssSelector("#input-email")).sendKeys("*****");
16        wl.findElement(By.cssSelector("#input-email")).sendKeys(c.next());
17        Thread.sleep(2000);
18
19        wl.findElement(By.cssSelector("#input-password")).clear();
20
21        wl.findElement(By.cssSelector("#input-password")).sendKeys("*&*~*&*");
22        wl.findElement(By.cssSelector("input.btn")).click();
23    }
24 }
```

<https://www.linkedin.com/in/gayatri-mishra-freelancecoach/>

Output:

```
Exception in thread "main" org.openqa.selenium.NoSuchElementException: Unable to locate element: #input-email
For documentation on this error, please visit: https://www.selenium.dev/documentation/webdriver/troubleshooting/errors#no-such-element
```