**How to handle Stale Element Reference exception in Selenium Webdriver**

October 19, 2018 [0 Comments](https://automatorsworld.com/2018/10/19/handle-stale-element-reference-exception-selenium-webdriver/#comments)

In this post, we will see how to handle stale element reference [**exception**](https://automatorsworld.com/2019/09/25/selenium-web-driver-how-to-handle-exceptions/). First of all. Before fixing any error it is important to [**read console error/exception message**](https://automatorsworld.com/2019/09/23/how-to-read-eclipse-console-error-messages-in-selenium-webdriver/)

First, we should understand what is stale element –

Possible causes for a stale element exception-

**1 -The element has been deleted entirely.**

**2- The element is no longer attached to the DOM.**

How to fix or handle it

**Solution 1**–

[The Most way to handle this](https://automatorsworld.com/2017/10/19/test-automation-some-useful-selenium-hacks/) is to refresh the page, On refreshing it, most of the time driver found the element, But it’s not the perfect solution-

Driver.navigate().refersh();

Driver.findElement(By.id(“property”)).click();

Driver.navigate().refersh();

Driver.findElement(By.id(“property”)).click();

**Solution 2-**

The second option you can try is to place your driver interaction command inside some loop and to repeat that for a couple of times, 2-3 times, refer below code.

however this method is also but hardcoded and not generic, it can also fail, if even after 3 attempt element is not found however its best to use the third option which is placing Do-While loop

for(int i=0;i<=3;i++)

{

try{

driver.findElement("Property").click();

break;

}

}

catch(Exception e)

{

System.out.println("Exception-Element not found");

}

}

**Solution 3-**

Use a do-While loop on an element which is causing stale element exception-Initialize a counter with Zero and iterate and identify webelement, once its enabled and displayed perform action and exit from the loop.

int Counter=0

do

{

try

{

String str ;

if(driver.findelement(By.id()).isenabled() && driver.findelement(By.id()).isdisplayed() )

{

Counter=Counter+1;

driver.findelement(By.id()).click();

break;

}

}

catch(Exception ex)

{

}

)

while(Counter == 0)

Using a do-while loop is the most generic and perfect solution for such stale element [exceptions](https://automatorsworld.com/2019/09/25/selenium-web-driver-how-to-handle-exceptions/), it should be able to find element even if it appears a bit late after loading a web page.