Stale Exception

===============

StaleElementException

Most automation tools depend on the concept of the page has finished loading. With AJAX and Web 2.0 this has become a grey area. META tags can refresh the page and Javascript can update the DOM at regular intervals.

For Selenium this means that StaleElementException can occur. StaleElementException occurs if I find an element, the DOM gets updated then I try to interact with the element.

Actions like:

driver.findElement(By.id("foo")).click();

are not atomic. Just because it was all entered on one line, the code generated is no different than:

By fooID = By.id("foo");

WebElement foo = driver.findElement(fooID);

foo.click();

If Javascript updates the page between the findElement call and the click call then I'll get a StaleElementException. It is not uncommon for this to occur on modern web pages. It will not happen consistently however. The timing has to be just right for this bug to occur.

Generally speaking, if you know the page has Javascript which automatically updates the DOM, you should assume a StaleElementException will occur. It might not occur when you are writing the test or running it on your local machine but it will happen. Often it will happen after you have 5000 test cases and haven't touched this code for over a year. Like most developers, if it worked yesterday and stopped working today you'll look at what you changed recently and never find this bug.

So how do I handle it? I use the following click method:

public boolean retryingFindClick(By by) {

boolean result = false;

int attempts = 0;

while(attempts < 2) {

try {

driver.findElement(by).click();

result = true;

break;

} catch(StaleElementException e) {

}

attempts++;

}

return result;

}

This will attempt to find and click the element. If the DOM changes between the find and click, it will try again. The idea is that if it failed and I try again immediately the second attempt will succeed. If the DOM changes are very rapid then this will not work. At that point you need to get development to slow down the DOM change so this works or you need to make a custom solution for that particular project.

The method takes as input a locator for the element you want to click. If it is successful it will return true. Otherwise it returns false. If it makes it past the click call, it will return true. All other failures will return false.