**Working with Frames in Selenium Webdriver**

What is iFrame? An iFrame (Inline Frame) is an HTML document embedded inside the current HTML document on a website. iFrame HTML element is used to insert content from another source, such as an advertisement, into a Web page. A Web designer can change an iFrame's content without making them reload the complete website. A website can have multiple frames on a single page. And a frame can also have inner frames (Frame in side a Frame)

---------------------------------------

| | |

| | |

| Frame 1 | |

| | |

| | |

|--------------| |

| | Frame 3 |

| | |

| | |

| | |

| Frame 2 | |

| | |

| | |

| | |

| | |

--------------------------------------------

In Selenium to work with iFrames, we have different ways to handle frame depending on the need. Please look at the below ways of handling frames

**driver.switchTo().frame(int arg0);**

Select a frame by its (zero-based) index. That is, if a page has multiple frames (more than 1), the first frame would be at index "0", the second at index "1" and so on.  
Once the frame is selected or navigated , all subsequent calls on the WebDriver interface are made to that frame. i.e the driver focus will be now on the frame. What ever operations we try to perform on pages will not work and throws element not found as we navigated / switched to Frame.

*Parameters: Index - (zero-based) index*  
*Returns: driver focused on the given frame (current frame)*  
*Throws: NoSuchFrameException - If the frame is not found.*

*Example: if iframe id=webklipper-publisher-widget-container-frame*, it can be written as driver.switchTo().frame("webklipper-publisher-widget-container-frame"); Below is the code snippet to work with switchToFrame using frame id.

**public** **void** **switchToFrame**(**int** frame) {

**try** {

driver.switchTo().frame(frame);

System.**out**.println("Navigated to frame with id " + frame);

} **catch** (NoSuchFrameException e) {

System.**out**.println("Unable to locate frame with id " + frame

+ e.getStackTrace());

} **catch** (Exception e) {

System.**out**.println("Unable to navigate to frame with id " + frame

+ e.getStackTrace());

}

}

**driver.switchTo().frame(String arg0);**

Select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID.  
*Parameters: name Or Id - the name of the frame or the id of the frame element.*  
*Returns: driver focused on the given frame (current frame)*  
*Throws: NoSuchFrameException - If the frame is not found*

Below is the example code snippet using frame name.

public void switchToFrame(String frame) {

try {

driver.switchTo().frame(frame);

System.out.println("Navigated to frame with name " + frame);

} catch (NoSuchFrameException e) {

System.out.println("Unable to locate frame with id " + frame

+ e.getStackTrace());

} catch (Exception e) {

System.out.println("Unable to navigate to frame with id " + frame

+ e.getStackTrace());

}

}

**driver.switchTo().frame(WebElement frameElement);**

Select a frame using its previously located WebElement.  
*Parameters: frameElement - The frame element to switch to.*  
*Returns: driver focused on the given frame (current frame).*  
*Throws: NoSuchFrameException - If the given element is neither an iframe nor a frame element. And StaleElementReferenceException - If the WebElement has gone stale.*

Below is the example code to send an Element to the and switch.

**public** **void** **switchToFrame**(WebElement frameElement) {

**try** {

**if** (isElementPresent(frameElement)) {

driver.switchTo().frame(frameElement);

System.**out**.println("Navigated to frame with element "+ frameElement);

} **else** {

System.**out**.println("Unable to navigate to frame with element "+ frameElement);

}

} **catch** (NoSuchFrameException e) {

System.**out**.println("Unable to locate frame with element " + frameElement + e.getStackTrace());

} **catch** (StaleElementReferenceException e) {

System.**out**.println("Element with " + frameElement + "is not attached to the page document" + e.getStackTrace());

} **catch** (Exception e) {

System.**out**.println("Unable to navigate to frame with element " + frameElement + e.getStackTrace());

}

}

Some times when there are multiple Frames (Frame in side a frame), we need to first switch to the parent frame and then we need to switch to the child frame. below is the code snippet to work with multiple frames.

public void switchToFrame(String ParentFrame, String ChildFrame) {

try {

driver.switchTo().frame(ParentFrame).switchTo().frame(ChildFrame);

System.out.println("Navigated to innerframe with id " + ChildFrame

+ "which is present on frame with id" + ParentFrame);

} catch (NoSuchFrameException e) {

System.out.println("Unable to locate frame with id " + ParentFrame

+ " or " + ChildFrame + e.getStackTrace());

} catch (Exception e) {

System.out.println("Unable to navigate to innerframe with id "

+ ChildFrame + "which is present on frame with id"

+ ParentFrame + e.getStackTrace());

}

}

After working with the frames, main important is to come back to the web page. if we don't switch back to the default page, driver will throw an exception. Below is the code snippet to switch back to the default content.

public void switchtoDefaultFrame() {

try {

driver.switchTo().defaultContent();

System.out.println("Navigated back to webpage from frame");

} catch (Exception e) {

System.out

.println("unable to navigate back to main webpage from frame"

+ e.getStackTrace());

}

}