Rex Jones II

Drag & Drop Target

In this video, we are going to drag this element to this Drop Here element using Selenium's Actions Class. Inspect both WebElements. Right click the first WebElement and we see draggable as the value for id. Right click the second element and the value for id is droppable.

Let's go to Eclipse, we have our set up and tear down methods. The test method is dragAndDropTarget. Start by loading the application: driver.get("https://jqueryui.com/droppable/");

There are 2 WebElements. First is WebElement source = driver.findElement(By.*id*("draggable")). The value for id is draggable. Next is the target. WebElement target = driver.findElement(By.*id*("droppable")); The value for id is droppable. Import WebElement. Did you notice that both WebElements were located inside an iFrame? Close the DOM and right click again. View Frame Source is an indicator that the WebElements are within a frame. Recall from the Switch To Frame Videos 29 – 32, we must switch to the frame before performing a command on the WebElement. If not, we will get an Exception while executing our Test Script.

Let's write //iframe and we see there is only 1 iframe on this webpage. As a result, we can use the index to locate the only iFrame. Go back to Eclipse and add our switch statement. driver.switchTo().frame(0);

Now, we implement the Actions class with act as the object reference = new Actions driver in the parenthesis. Import the Actions class. Object Reference act dot dragAndDrop. Here we go, the drag and drop methods. Look at the description "A convenience method that performs click-and-hold at the location of the source element, moves to the location of the target element, then releases the mouse." The parameters are source and target. Next, we perform the action by writing perform.

Let's Run. That's It for Dragging Then Dropping A Target