**Handling Keyboard & Mouse Events**

Advanced user Interactions

Handling special keyboard and mouse events are done using the **Advanced User Interactions API**. It contains the **Actions** and the **Action** classes that are needed when executing these events. The following are the most commonly used keyboard and mouse events provided by the Actions class.

In order to handle advanced user interactions we have to use class as **Actions.**

**Action is a class and interface**

**Actions builder=new Actions(web driver object)**

**Actions builder=new Actions(driver)**

**moveToElement and** click

//inspect source web element where you will ose over

WebElement eleMouseOver = driver.findElementByXPath("//span[text()='TVs & Appliances']");

//inspect the taget to click

WebElement eletarget = driver.findElementByXPath("//span[text()='Thomson']");

//mouse over

Actions builder=**new** Actions(driver);

builder.moveToElement(eleMouseOver).pause(3000).click(eletarget).build().perform();

**moveToElement(eleMouseOver)🡪which element you want to mouse over**

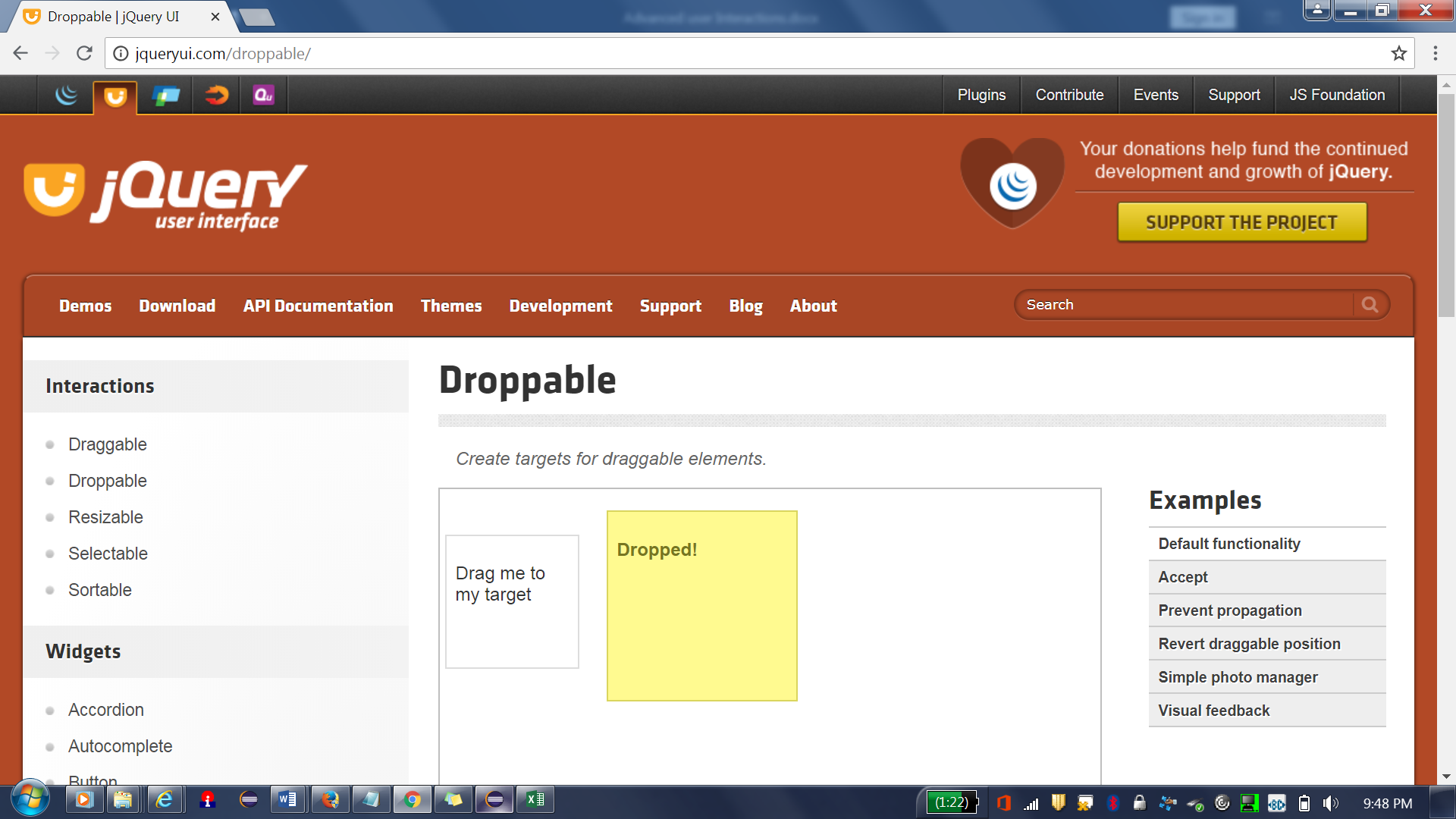
**click(eletarget)🡪 click on target element**

Build()

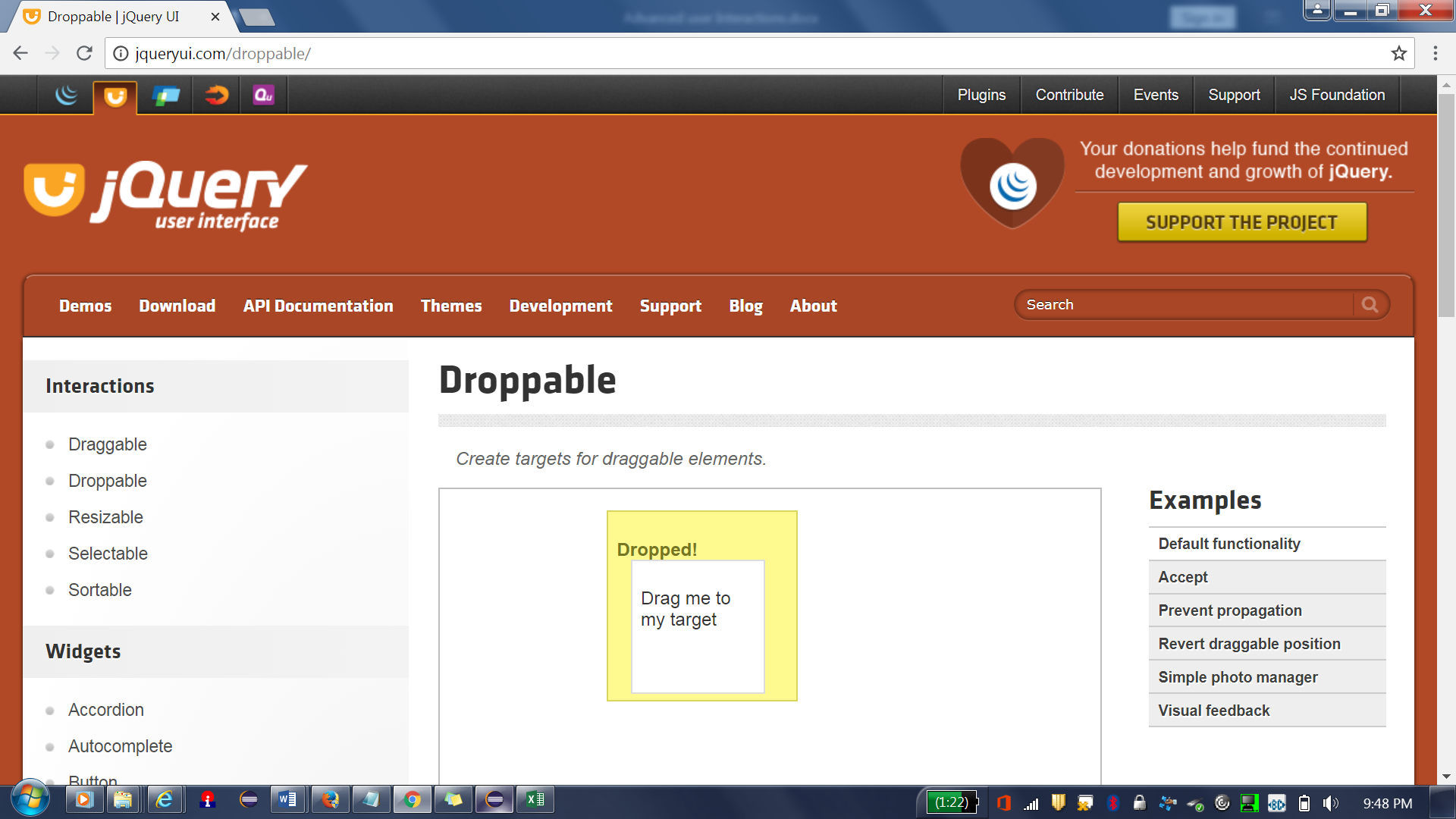
**Will add two actions and compile and it keep ready to perform**

**Perform() will perform that combined two actionss**

**DragAndDrop**



**draganddrop**



WebElement source = driver.findElementByXPath("//div[@id='draggable']/p");

//inspect target element

WebElement target = driver.findElementByXPath("//div[@id='droppable']/p");

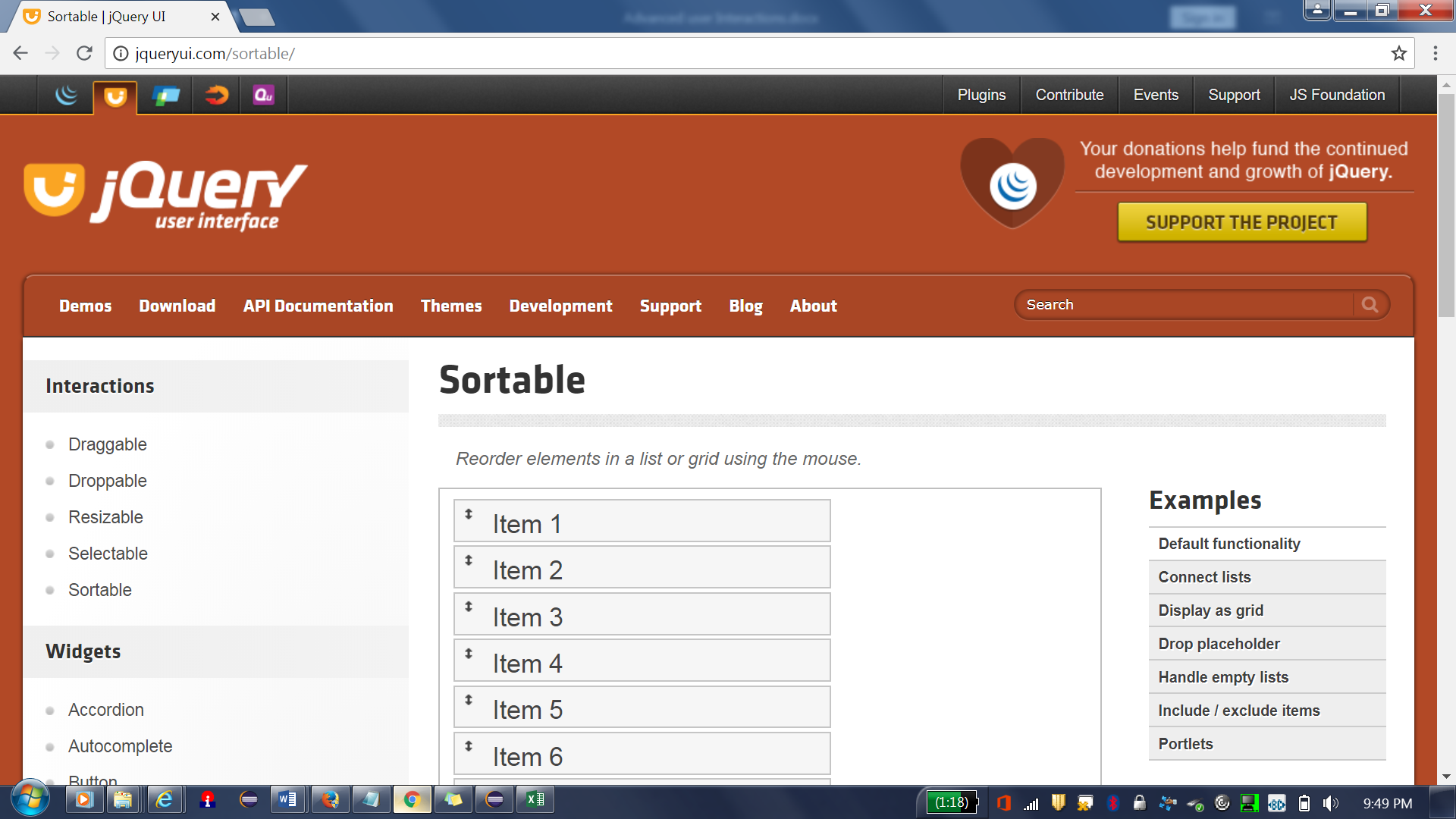
//drag and drop

Actions builder=**new** Actions(driver);

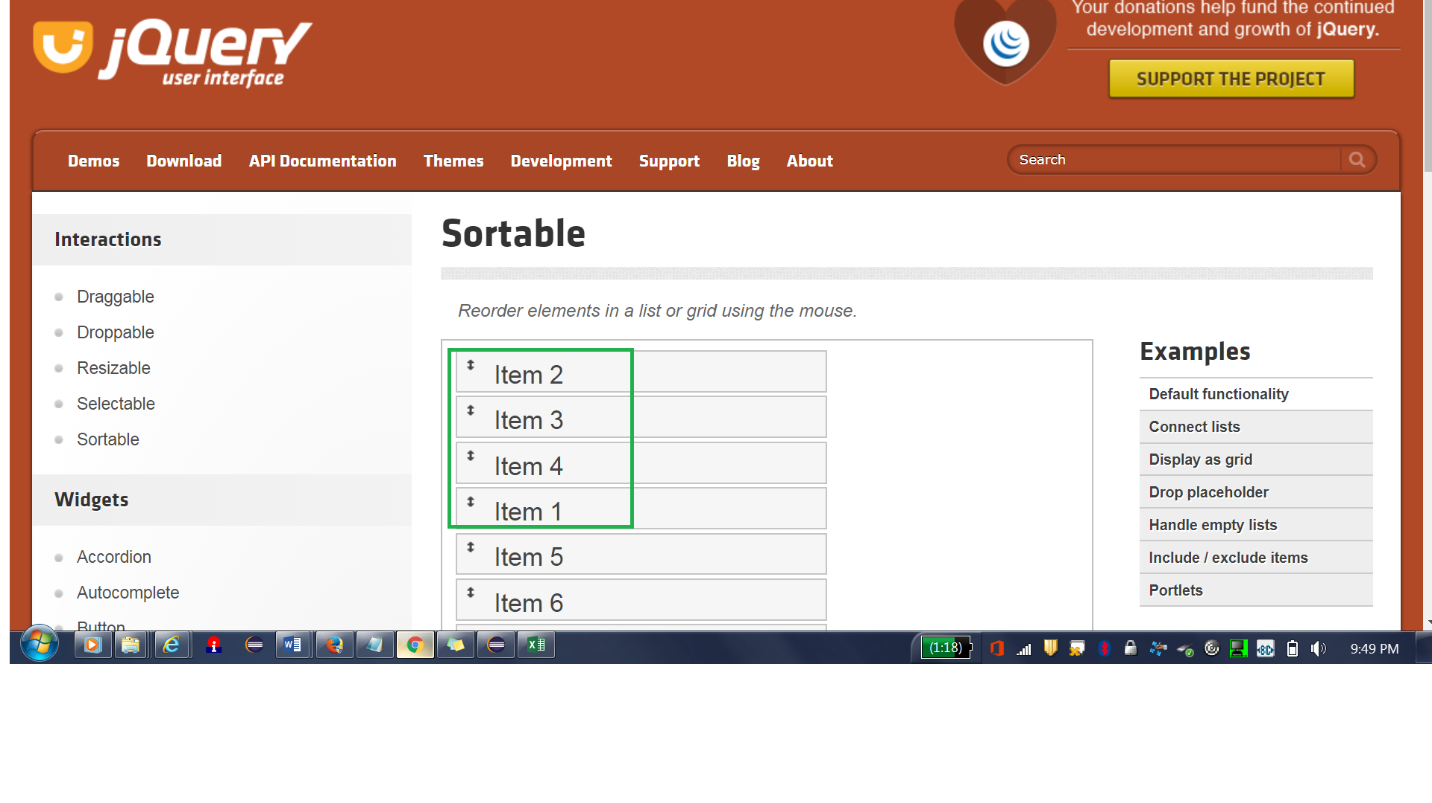
builder.dragAndDrop(source, target).perform();

dragAndDropBy

**Before sort**



**After sort**

****

WebElement source = driver.findElementByXPath("//ul[@id='sortable']/li[1]");

//inspect target element

WebElement target = driver.findElementByXPath("//ul[@id='sortable']/li[4]");

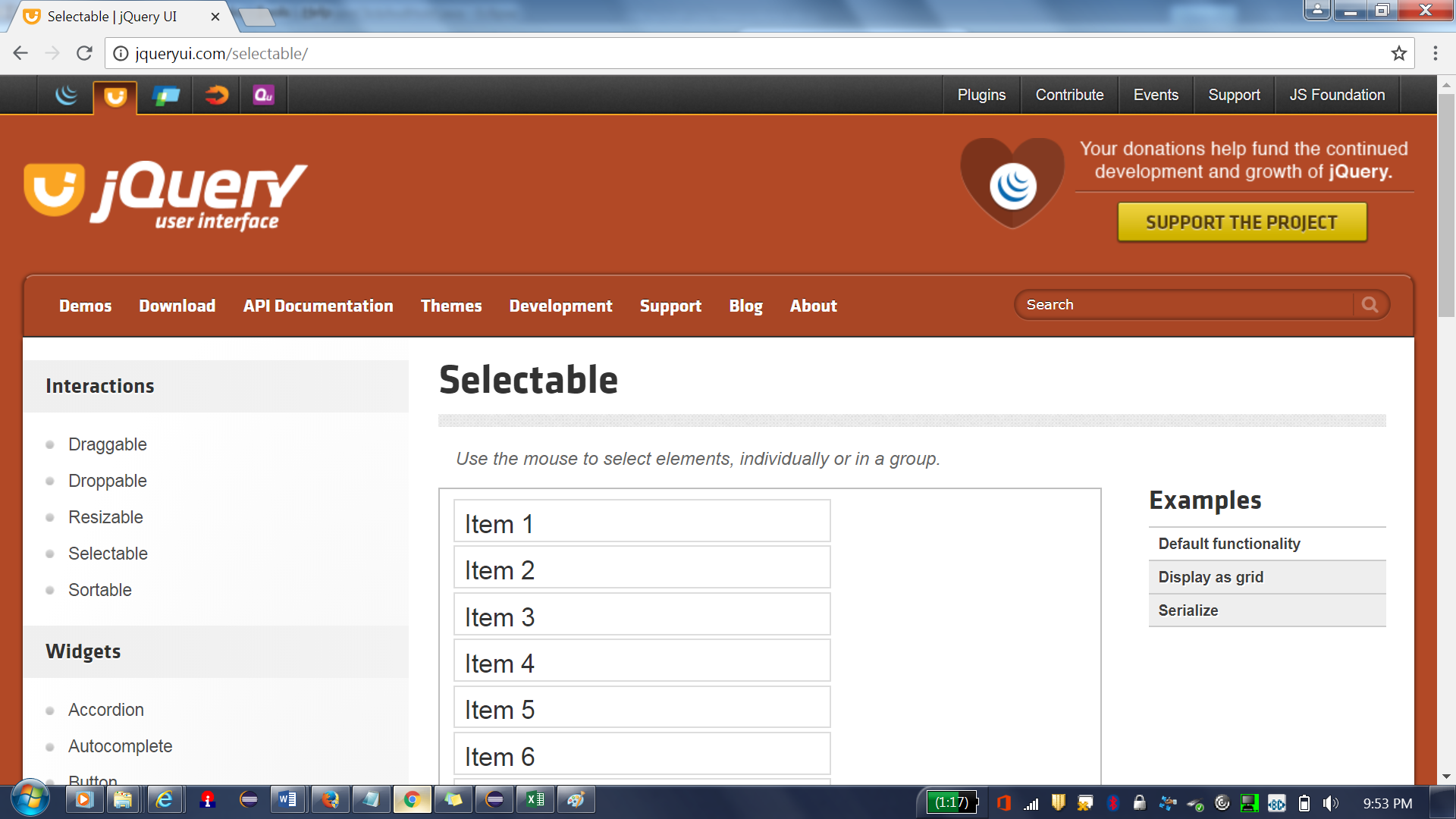
//drag and drop

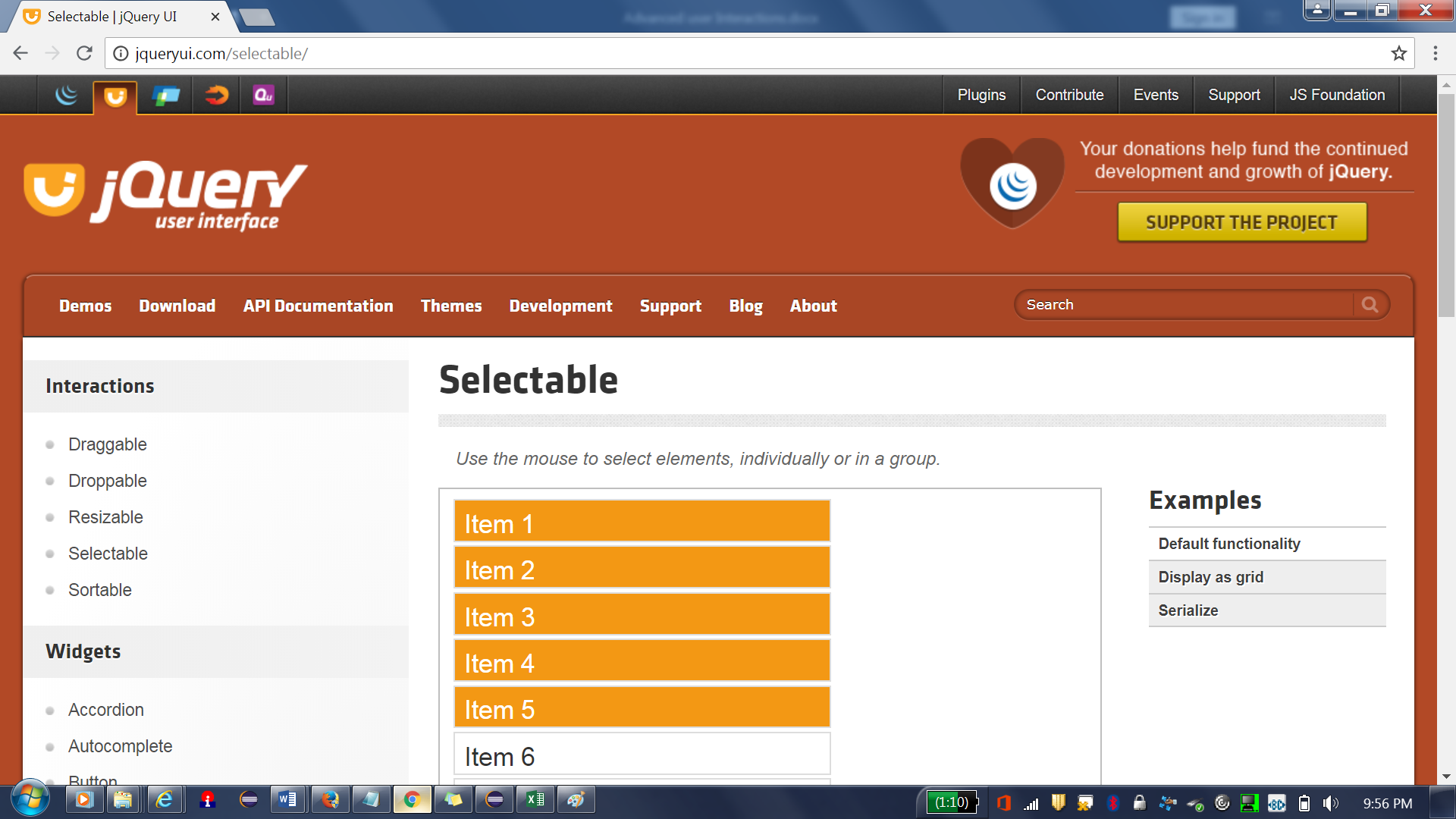
Actions builder=**new** Actions(driver);

builder.dragAndDropBy(source, target.getLocation().getX(), target.getLocation().getY()).perform();

**Select from item 1 to item 4**

clickAndHold and release





WebElement item1 = driver.findElementByXPath("//ol[@id='selectable']/li[1]");

WebElement item5 = driver.findElementByXPath("//ol[@id='selectable']/li[5]");

//click and hold

Actions builder=**new** Actions(driver);

driver.getKeyboard().sendKeys();

builder.clickAndHold(item1).release(item5).perform();

System.***out***.println("click and hold success");

**Summary**

**Mouse over and click--------moveToElement and click**

builder.moveToElement(eleMouseOver).pause(3000).click(eletarget).build().perform();

dragAndDrop

drag from source element and drop on target element

builder.dragAndDrop(source, target).perform();

select from item 1 to item 4---clickAndHold and release

builder.clickAndHold(item1).release(item5).perform();

to swap from item source to target

dragAndDropBy--sort

builder.dragAndDropBy(source, target.getLocation().getX(), target.getLocation().getY()).perform();

**WebElement source = driver.findElementByXPath("//ul[@id='sortable']/li[1]");**

**source.sendKeys("mounis",Keys.TAB);**

**webElement.sendKeys(Keys.TAB);**

**webElement.sendKeys(Keys.ENTER);**

Double click functionality(Actions)

Actions action = new Actions(driver);

action. doubleClick(element). build().perform();

how Right click functionality in selenium

Actions action = new Actions(driver);

action. contextClick(element).build().perform();