What Is Robot Class in Selenium?

Robot Class facilitates Selenium to use actual mouse and keyboard events rather than simulated mouse. Robot Class can handle window based pop-pos like(alerts, download based pops, etc) or native applications like(Calculator, Notepad, etc). Robot Class in Selenium also helps in completing the task in specified time.

Robot Class is invoked using the following syntax:

```
Robot robot = new Robot()

robot.mouseMove(coordinates.getX(), coordinates.getY());
```

Robot Class in Java

Robot Class was introduced in Java in version 1.3. The Robot Class in Java AWT package is to generate native system input events for the purpose of automation testing, self running demos and other applications where control of mouse and keyboard is needed. The main purpose of this class is to provide automated testing of java platform implementations.

Methods of Robot class

Mouse Actions:

1. **mouseMove(X,Y):** Move mouse pointer to given screen coordinates.
2. **mouseRelease(buttons):** Releases one or more mouse buttons.
3. **mousePress(buttons):** Presses one or more mouse buttons.
4. **mouseWheel(wheelAmt):** Rotates the scroll wheel on wheel-equipped-mice.
5. **getPixelColor(X,Y):** Returns the color of the pixel at a given screen coordinates.

Keyboard Actions:

1. **keyPress(keycode):** Presses a given key.
2. **keyRelease(keycode):** Releases a given key.

For example, Robot.mouseMove will actually move the mouse cursor instead of just generating mouse move events.

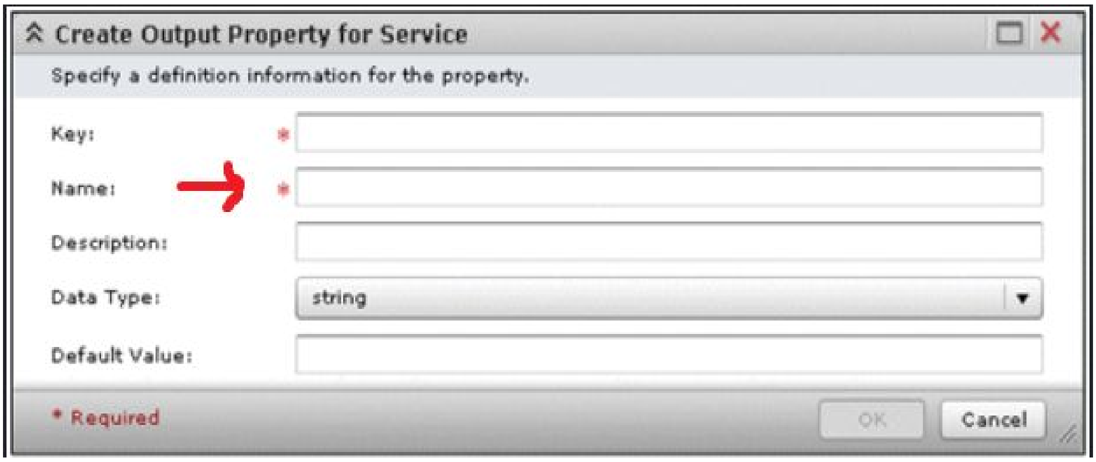
Difference between Action class and Robot class in Selenium

The difference between Action class and Robot class is mentioned below:

Action class	Robot class
Action class simulates a mouse and Keyboard.	Robot class enables the actual mouse and keyboard, because of this reason you can see movement of mouse cursor.
Action class doesn't affect parallel running.	Robot class affects parallel running as there is only one mouse connected to the system.
Action class uses the native browser commands.	Robot class doesn't use browser based commands.
Action class is limited to browser application.	Robot class can be used along with all applications.
Action class is from org.openqa .selenium.interactions.Actions.	Robot class is from java.awt.Robot.

Enter Text Using Robot class in Selenium

Test Scenario: If the mouse pointer(means control) is present in the text box the we can perform action using Robot class.



Create Output Property for Service

Specify a definition information for the property.

Key: *

Name: *

Description:

Data Type: string

Default Value:

* Required

Code Snippet:

```
import java.awt.AWTException;
import java.awt.Robot;
import java.awt.event.KeyEvent;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium..chrome.ChromeDriver;
public class Check {
    public static void main(String[] arg) throws AWTException, InterruptedException
    {
        WebDriver driver = new ChromeDriver();
         driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
         driver.get(&quot;http://atul-pc/login.do&quot;);
        Thread.sleep(3000);
         //used to move cursor on particular element and passing control
        WebElement un = driver.findElement(By.name(&quot;name&quot;));
        int xaxis = un.getLocation().x;
        int yaxis = un.getLocation().y;
        int width = un.getSize().width;
        int height = un.getSize().height;
        Robot r = new Robot();
        r.mouseMove(xaxis+width/2, yaxis+height/2 + 70);
        r.mousePress(KeyEvent.BUTTON1_MASK);  // Click function
        r.mouseRelease(KeyEvent.BUTTON1_MASK);
        //type text in text box one by one
        r.KeyPress(KeyEvent.VK_A);
        r.KeyPress(KeyEvent.VK_D);
        r.KeyPress(KeyEvent.VK_M);
        r.KeyPress(KeyEvent.VK_I);
        r.KeyPress(KeyEvent.VK_N);
        r.keyRelease(KeyEvent.VN_A);
        r.keyRelease(KeyEvent.VN_D);
        r.keyRelease(KeyEvent.VN_M);
        r.keyRelease(KeyEvent.VN_I);
        r.keyRelease(KeyEvent.VN_N);
    }
}
```

File Upload Using Robot class in Selenium

Test Scenario: There are different methods that Robot class uses. Here is the below example we have used ‘keyPress’ and ‘keyRelease’ methods.

Code Snippet:

```
import java.awt.Robot;
import java.awt.Toolkit;
import java.awt.datatransfer.StringSelection;
import java.awt.event.KeyEvent;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class UploadFileRobot {
        String URL = "application URL";
        @Test
        public void testUpload() throws InterruptedException
        {
                WebDriver  driver = new ChromeDriver();
                driver.get(URL);
                WebElement element = driver.findElement(By.name("uploadfile"));
                element.click();
                uploadFile("path to the file");
                Thread.sleep(2000);
        }
 // This method will set any parameter string to the system's clipboard.
        public static void setClipboardData(String string) {
                //StringSelection is a class that can be used for copy and paste operations.
                StringSelection stringSelection = new StringSelection(string);
                Toolkit.getDefaultToolkit().getSystemClipboard().setContents(stringSelection,
null);
                }

        public static void uploadFile(String fileLocation) {
        try {
```

```
        //Setting clipboard with file location
        setClipboardData(fileLocation);
        //native key strokes for CTRL, V and ENTER keys
        Robot robot = new Robot();
        robot.keyPress(KeyEvent.VK_CONTROL);
        robot.keyPress(KeyEvent.VK_V);
        robot.keyRelease(KeyEvent.VK_V);
        robot.keyRelease(KeyEvent.VK_CONTROL);
        robot.keyPress(KeyEvent.VK_ENTER);
        robot.keyRelease(KeyEvent.VK_ENTER);
        } catch (Exception exp) {
                exp.printStackTrace();
        }
        }
}
```

Limitations of Robot class in Selenium

Though robot class helps to manage window based pop-up, Robot framework has few disadvantages mentioned below:

1. Most of the methods like mouseMove are dependent on screen resolution, so the method might not perform the same on different machines.

2. Parallel running should be avoided because Robot class facilitates actual mouse commands so a machine cannot have two mice.

3. This class works only on the current instance of window, so behaviour might be different when multiple instances are open.

4. If a key is pressed using keyPress and not released using keyRelease then it will remain pressed and will consume memory in the background.

5. It may raise security exception if createRobot permission is not granted.

References

1. https://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html

2. https://arunseleniumhelp.wordpress.com/2016/08/06/how-to-type-text-using-robot-class-in-selenium/

3. https://www.seleniumeasy.com/selenium-tutorials/webdriver-file-upload-using-robots