# File upload

## Sendkeys

public void fileupload() throws Exception {

w.get("https://demoqa.com/automation-practice-form");

WebElement choosefile = w.findElement(By.xpath("//input[@id='uploadPicture']"));

choosefile.sendKeys("C:\\Users\\koush\\OneDrive\\Desktop\\temp.txt");

Thread.sleep(3000);

**}**

**Notes:**

* **sendKeys() works only with <input type="file"> elements.**
* **No need to interact with the system file dialog — Selenium directly sets the file path.**
* **Make sure the file path is absolute and exists.**

## ROBOT Class

**If the site uses a custom button that opens a system dialog (non-HTML <input type="file">), you cannot automate it directly using Selenium — instead, use tools like:**

* **AutoIt (Windows)**
* **Robot class (Java AWT)**
* **PyAutoGUI (for Python users)**

public void robotclassdemo() throws Exception {

// TODO Auto-generated method stub

w.get("https://demoqa.com/automation-practice-form");

JavascriptExecutor js = (JavascriptExecutor) w; // to perform scroll

down action

js.executeScript("scroll(0,500)");

// Click on 'Choose File' button

WebElement choosefile = w.findElement(By.xpath("//input[@id='uploadPicture']"));

Actions a = new Actions(w);

a.moveToElement(choosefile).click().build().perform();

// Copy file path to clipboard

StringSelection filePath = new StringSelection("C:\\Users\\koush\\OneDrive\\Desktop\\temp.txt");

Toolkit.getDefaultToolkit().getSystemClipboard().setContents(filePath, null);

// Use Robot to paste path and press Enter

Robot robot = new Robot();

Thread.sleep(1000);

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);

}

**Summary:**

* Uses Robot to paste a file path into system dialog.
* Works when file input can't be automated via sendKeys().

# Drag and drop

public void drag\_n\_drop() throws Exception {

w.get("https://demoqa.com/droppable");

WebElement dragg = w.findElement(By.xpath("//div[@id='draggable']"));

WebElement drop = w.findElement(By.xpath("//div[@id='droppable']"));

Actions a = new Actions(w);

a.dragAndDrop(dragg, drop).build().perform();

}

**Explanation:**

* driver.switchTo().frame(0); → because the drag/drop elements are inside an iframe.
* Actions actions = new Actions(driver); → for mouse interactions.
* dragAndDrop(source, target) → performs drag from source and drop into target.

# New tab / Multiple Window handle

**public** **void** multipletabs\_windows() {

*w*.get("https://rahulshettyacademy.com/AutomationPractice/");

// Store the original tab's handle

String originalTab = *w*.getWindowHandle();

System.***out***.println("Original Tab Handle: " + originalTab);

// Click the link that opens a new tab

// w.findElement(By.xpath("//a[@id='opentab']")).click(); // tab

*w*.findElement(By.*xpath*("//button[@id='openwindow']")).click(); // window

// Get all window handles. It will now be a set of two handles.

Set<String> allTabs = *w*.getWindowHandles();

System.***out***.println("Number of tabs open: " + allTabs.size());

// Iterate through the handles to find the new tab

Iterator<String> iterator = allTabs.iterator();

String newTab = "";

**while** (iterator.hasNext()) {

String currentTab = iterator.next();

**if** (!currentTab.equalsIgnoreCase(originalTab)) {

newTab = currentTab;

}

}

// Switch to the new tab

*w*.switchTo().window(newTab);

System.***out***.println("Title of the new tab: " + *w*.getTitle());

}

# List / Tabel

// Create list of all element and click on one specific element

Steps

1. Get all elements using findelements
2. Create list of all elements List<WebElement>
3. Uer For each loop to itrate each element
4. getText() from each element and verify the name
5. in if condition perform click() and break the loop

**public** **void** listall(String prod) {

*w*.get("https://vinothqaacademy.com/ecommerce-demo/home.html");

List<WebElement> list = *w*.findElements(By.*xpath*("//h3"));

System.***out***.println(list);

**for** (WebElement ele : list) {

String productname = ele.getText();

System.***out***.println(productname);

//

**if** (productname.equalsIgnoreCase(prod)) {

ele.click();

**break**;

}

}

System.***out***.println("list over");

}

**public** **void** initialize() {

ChromeOptions op = **new** ChromeOptions();

op.addArguments("--incognito");

*w* = **new** ChromeDriver(op);

*w*.manage().window().maximize();

*w*.manage().deleteAllCookies();

*w*.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(10));

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

links\_tabel tb = **new** links\_tabel();

tb.initialize();

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("tell me product name");

String prod = sc.nextLine(); // Google Pixel 8 Pro

tb.listall(prod);

}

# Frame

**Why Switch to Frame?**

Because elements inside a frame are isolated from the main page. So you must tell Selenium to "enter" the frame first.

**✅ Ways to Switch to a Frame in Selenium Java:**

Selenium provides 3 main ways to switch to a frame:

| **Method** | **Usage** |
| --- | --- |
| driver.switchTo().frame(index) | Switch using frame **index** (starting from 0) |
| driver.switchTo().frame("nameOrId") | Switch using frame's **name or id** |
| driver.switchTo().frame(WebElement) | Switch using the **frame WebElement** |

**public** **void** frame() {

*w*.get("https://rahulshettyacademy.com/AutomationPractice/");

WebElement frame1 = *w*.findElement(By.*id*("courses-iframe"));

*w*.switchTo().frame(frame1);

System.***out***.println(*w*.getTitle());

List<WebElement> iframe\_element = *w*.findElements(By.*xpath*("//ul[@class='navigation clearfix']/li/a"));

**for** (WebElement ele : iframe\_element) {

String menu = ele.getText();

System.***out***.println(menu); // print list present under frame.

}

System.***out***.println("list over");

*w*.switchTo().frame(0);

System.***out***.println(*w*.getTitle());

}

# TakescreenShot

**public** **void** takescreenshot() **throws** Exception {

TakesScreenshot screenshot = (TakesScreenshot) *w*; // casting webdriver to takescreenshot

File src = screenshot.getScreenshotAs(OutputType.***FILE***); //capturing SS

File dest = **new** File("./Screenshot/basket.png"); // creating distination path

Files.*copy*(src, dest); //storing to that path

}

# Table

**public** **void** handletabel() {

// Print price written against the specific course

*w*.get("https://rahulshettyacademy.com/AutomationPractice/");

List<WebElement> cousList = *w*.findElements(By.*xpath*("//table[@id='product'][1]/tbody/tr/td[2]"));

**for** (WebElement ele : cousList) {

String courseName = ele.getText();

**if** (courseName.equalsIgnoreCase("WebServices / REST API Testing with SoapUI")) {

String price = ele.findElement(By.*xpath*("following-sibling::td")).getText();

System.***out***.println("Price of cource is " + price);

**break**;

}

}

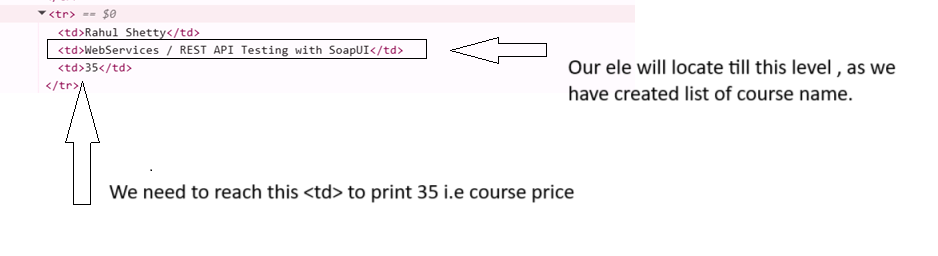
}

ele.findElement(By.*xpath*("following-sibling::td")).getText();

* This line represent, we are finding the element within ele element.
* As we are finding within we can continue from current location.
* We can use following-sibling in XPath: xpath

//tagname[condition]/following-sibling::tagname

* It selects the **next siblings** (with same parent) of the current node.



This will conclude basic