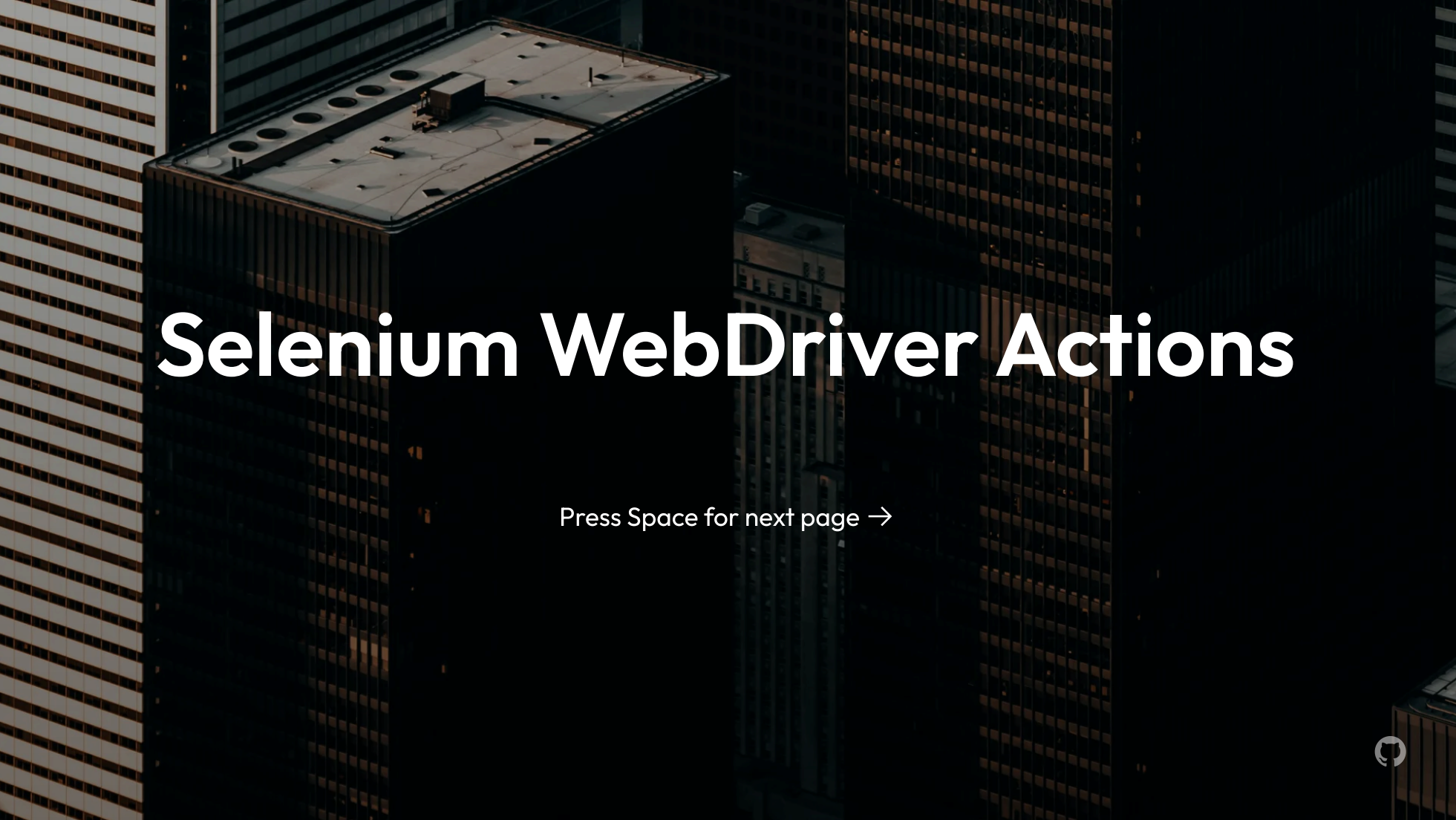

Selenium WebDriver Actions
Press Space for next page →

Agenda

1. WebDriver Actions in Selenium
2. Double Click: Code Example
3. Right Click: Code Example
4. Jquery - Draggable
5. Jquery - Droppable
6. Drag and Drop(Manual) : Code Example
7. Drag and Drop(Auto) : Code Example
8. Draggable: Code Example
9. Keyboard Actions
10. Keyboard Actions: Code Example
11. Keyboard Actions: Code Snippet
12. Summery
13. Mouse Hover
14. Mouse Hover: Code Example

WebDriver Actions in Selenium

The Actions class in Selenium WebDriver is a powerful feature used to automate complex user interactions on web applications. This class is part of the `org.openqa.selenium.interactions` package and allows you to perform advanced user interactions like mouse movements, keypresses, drag-and-drop operations, and other interactions that are not possible using the basic WebDriver methods.

Examples:

- Mouse Actions
- Keyboard Actions
- Drag & Drop actions

Documentation: Selenium WebDriver Actions

Double Click: Code Example

```
class DoubleClickExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

        driver.get("https://api.jquery.com/dblclick/");

        driver.switchTo().frame(driver.findElement(By.xpath("//iframe")));

        Actions actions = new Actions(driver);

        WebElement squareBox = driver.findElement(By.xpath("//span[text()='Double click the block']//parent::body/div"));
        actions.doubleClick(squareBox).perform();

        Thread.sleep(3000);

        System.out.println("Yeeee I double clicked the element !!! wowww !!!");
        driver.quit();
    }
}
```

Right Click: Code Example

```
class RightClickExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

        driver.get("https://swisnl.github.io/jQuery-contextMenu/demo.html");
        driver.manage().window().maximize();

        Actions actions = new Actions(driver);

        actions.contextClick(driver.findElement(By.xpath("//span[contains(text(),'right click me')]"))).perform();

        driver.findElement(By.xpath("//li[@class='context-menu-item context-menu-icon context-menu-icon-copy']")).click();
        System.out.println(driver.switchTo().alert().getText());

        Thread.sleep(3000);

        driver.switchTo().alert().accept();

        Thread.sleep(3000);

        driver.quit();
    }
}
```

Jquery - Draggable

> Functionality: Allows elements to be clicked and dragged using the mouse.

Common Use Cases:

- Rearranging items on a page.
- Implementing custom sliders.
- Creating drag-and-drop interfaces.

Documentation: Jquery Draggable

Jquery - Droppable

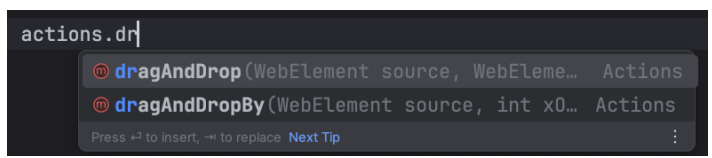
> Functionality: Designates elements as targets for draggable items.

Common Use Cases:

- Building drag-and-drop file uploaders.
- Creating interactive games.
- Developing intuitive UI components.

Playground

URL: 1
URL: 2



Drag and Drop(Manual) : Code Example

```
class DragAndDropManualExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

        driver.get("https://dhtmlx.com/docs/products/dhtmlxTree/");
        driver.manage().window().maximize();

        driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@class='js-iframe active']")));

        Actions actions = new Actions(driver);

        WebElement src = driver.findElement(By.xpath("//span[text()='Lawrence Block']"));
        WebElement dest = driver.findElement(By.xpath("//li[text()='Zend Framework in Action']"));

        actions.clickAndHold(src)
                .pause(Duration.ofSeconds(2))
                .moveToElement(dest)
                .pause(Duration.ofSeconds(2))
                .release()
                .build()
                .perform();

        driver.quit();
    }
}
```

Drag and Drop(Auto) : Code Example

```
class DragAndDropAutoExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

        driver.get("https://dhtmlx.com/docs/products/dhtmlxTree/");
        driver.manage().window().maximize();

        driver.switchTo().frame(driver.findElement(By.xpath("//iframe[@class='js-iframe active']")));

        WebElement src = driver.findElement(By.xpath("//span[text()='Lawrence Block']"));
        WebElement dest = driver.findElement(By.xpath("//li[text()='Zend Framework in Action']"));

        Actions actions = new Actions(driver);
        actions.dragAndDrop(src, dest).perform();

        driver.quit();
    }
}
```

Draggable: Code Example

```
class DraggableExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

        driver.get("https://jqueryui.com/draggable/");
        driver.manage().window().maximize();
        driver.switchTo().frame(driver.findElement(By.tagName("iframe")));

        Actions actions = new Actions(driver);
        WebElement src = driver.findElement(By.xpath("//div[@id='draggable']"));

        actions.dragAndDropBy(src,394, 16).perform();
    }
}
```

Keyboard Actions

> Keyboard Actions: Enables simulation of keyboard events like key presses and releases.

Common Keyboard Actions:

- sendKeys(): Sends a sequence of keystrokes to an element.
- keyDown(): Simulates pressing a key down.
- keyUp(): Simulates releasing a key.

Keyboard Actions: Code Example

```
class KeyboardActionsExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

        driver.get("http://www.google.com");

        driver.manage().window().maximize();

        Actions actions =new Actions(driver);

        actions.click(driver.findElement(By.xpath("//textarea[@name='q']")))
                .sendKeys("Nexxvali")
                .pause(Duration.ofSeconds(2))
                .sendKeys(Keys.ARROW_DOWN)
                .pause(Duration.ofSeconds(2))
                .sendKeys(Keys.ENTER)
                .build()
                .perform();

    }
}
```

Keyboard Actions: Code Snippet

Selecting all text

```
Actions actions = new Actions(driver);
actions.click(inputField)
        .keyDown(Keys.CONTROL)
        .sendKeys("a")
        .keyUp(Keys.CONTROL)
        .build()
        .perform();
```

Copy-Paste Text

```
Actions actions = new Actions(driver);
actions.click(inputField)
        .keyDown(Keys.CONTROL)
        .sendKeys("a")
        .sendKeys("c")
        .keyUp(Keys.CONTROL)
        .sendKeys(Keys.TAB)
        .keyDown(Keys.CONTROL)
        .sendKeys("v")
        .keyUp(Keys.CONTROL)
        .build()
        .perform();
```

Summery

- Action Class: Essential for handling complex keyboard interactions.
- Key Methods: sendKeys(), keyDown(), keyUp().
- Practical Use: Automating tasks like text input, selection, and copy-paste.

Playground: URL

Mouse Hover

- Mouse Hover: Simulates mouse movement to an element without clicking it.
- Common Use Case: To reveal hidden elements or trigger hover-dependent events (like dropdown menus).
- Action Class
- Purpose: Provides methods to handle complex user gestures.
- Key Method: moveToElement() – moves the mouse to the middle of the element.

Mouse Hover: Code Example

```
class MouseHoverExample {
    public static void main(String[] args) {
        WebDriver driver = new ChromeDriver();

        driver.get("http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html");

        driver.manage().window().maximize();

        WebElement hoverButton = driver.findElement(By.xpath("//button [text()='Automation Tools']"));

        Actions act = new Actions(driver);
        act.moveToElement(hoverButton).perform();

        List<WebElement> links = driver.findElements(By.xpath("//div[@class='dropdown-content']//a"));
        int total_count = links.size();

        for (int i = 0; i < total_count; i++) {

            WebElement element = links.get(i);
            String text = element.getAttribute("href");
            System.out.println("Link: " + text);
        }
    }
}
```

Thank you ❤️

qa-june-2024-automation-with-java-slides