

How to Perform Mouse Actions in Playwright

When we talk about mouse actions in Playwright, we mostly perform the following actions:

1. Mouse Hover
 2. Right Click
 3. Double Click
 4. Drag and Drop
-

1. Mouse Hover

Description:

Mouse hover is when the cursor is placed over a particular element, and that element gets highlighted. In Playwright, handling the mouse hover action is very simple. To hover over two elements, find the locators for both, and then use the `hover()` function to perform the hover action.

Example DOM:

```
<div id="elementOne">Element One</div>
<div id="elementTwo">Element Two</div>
```

Code:

```
import { test, expect } from '@playwright/test';

test('Mouse Hover', async ({ page }) => {
    await page.goto('https://yourwebsite.com');

    const elementOne = await page.locator('//div[@id="elementOne"]');
    const elementTwo = await page.locator('//div[@id="elementTwo"]');

    await elementOne.hover();
    await elementTwo.hover().click();
});
```

2. Right Click

Description:

To right-click on an element, such as a button, in Playwright, find the element as usual and then use the `click()` function, passing `button: 'right'` inside the {} options.

Example DOM:

```
<button id="buttonRight">Right Click Me</button>
```

Code:

```
import { test, expect } from '@playwright/test';

test('Right Click', async ({ page }) => {
  await page.goto('https://yourwebsite.com');

  const button = await page.locator('//button[@id="buttonRight"]');
  await button.click({ button: 'right' });
});
```

3. Double Click

Description:

A double-click is when the mouse button is pressed twice quickly on the same element. In Playwright, you can use the `dblclick()` function to simulate a double-click on an element.

Example DOM:

```
<button id="buttonDoubleClick">Double Click Me</button>
```

Code:

```
import { test, expect } from '@playwright/test';

test('Double Click', async ({ page }) => {
  await page.goto('https://yourwebsite.com');

  const button = await page.locator('//button[@id="buttonDoubleClick"]');
  await button dblclick();
});
```

4. Drag and Drop

Description:

To perform a drag-and-drop action, we need to identify two elements: the source (where to drag from) and the destination (where to drop).

Approach 1: Using Mouse Functions

This approach involves using `hover()`, `mouse.down()`, `hover()` again on the target element, and finally using `mouse.up()` to drop the element.

Example DOM:

```
<div id="elementOne">Drag Me</div>
<div id="elementTwo">Drop Here</div>
```

Code:

```
import { test, expect } from '@playwright/test';

test('Drag and Drop', async ({ page }) => {
    await page.goto('https://yourwebsite.com');

    const elementOne = await page.locator('#elementOne');
    const elementTwo = await page.locator('#elementTwo');

    await elementOne.hover();
    await page.mouse.down();

    await elementTwo.hover();
    await page.mouse.up();
});
```

Approach 2: Using `dragTo()` Function

An alternative method is to use the `dragTo()` function, which simplifies the drag-and-drop process.

Example DOM:

```
<div id="elementOne">Drag Me</div>
<div id="elementTwo">Drop Here</div>
```

Code:

```
import { test, expect } from '@playwright/test';

test('Drag and Drop', async ({ page }) => {
    await page.goto('https://yourwebsite.com');

    const elementOne = await page.locator('#elementOne');
    const elementTwo = await page.locator('#elementTwo');

    await elementOne.dragTo(elementTwo);
});
```

Summary of Mouse Actions:

- **Mouse Hover:** Use the `hover()` function to move the cursor over an element.
 - **Right Click:** Use the `click()` function with `{ button: 'right' }` to perform a right-click.
 - **Double Click:** Use the `dblclick()` function to simulate a double-click on an element.
 - **Drag and Drop:** You can either use `mouse.down()` and `mouse.up()` or the `dragTo()` function to drag and drop an element.
-