Handling Bootstrap Dropdowns in Playwright

Bootstrap dropdowns often don't have a `<select>` tag in the HTML structure. Instead, they are typically built using other HTML elements like `<button>` or `<div>`. These types of dropdowns are known as **Bootstrap dropdowns**.

Example DOM for a Bootstrap Dropdown:

```
<div class="dropdown bootstrap-select show-tick">
    <button type="button" class="btn btn-default multiselect" data-
toggle="dropdown">
        Select Options
    </button>
    <ul class="dropdown-menu">
        <li><label><input type="checkbox" value="Option 1"> Option </label></li>
        <li><label><input type="checkbox" value="Option 2"> Option </label></li>
        <li><label><input type="checkbox" value="Option 3"> Option </label></li>
        <li><label><input type="checkbox" value="Option 4"> Option </label></li>
        <li><label><input type="checkbox" value="Option 5"> Option </label></li>
        <li><label><input type="checkbox" value="Option 6"> Option </label></li>
        <li><label><input type="checkbox" value="Option 7"> Option </label></li>
        <li><label><input type="checkbox" value="Option 8"> Option </label></li>
    </ul>
</div>
```

This dropdown uses a `<button>` to toggle the options and a `<ul>` list with `<input>` checkboxes to represent the options.

Click on the Bootstrap Dropdown and Count the List Items (Using XPath)

You can click on the dropdown button and count the number of items in the list using XPath.

Example: Count List Items Using XPath

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

test('Handle Bootstrap dropdown and count the list items using XPath',
async ({ page }) => {
    await page.goto('https://your-website-url.com'); // Replace with actual
URL

    // Click the dropdown button
    await page.locator('//button[contains(@class,
"multiselect")]').click();

    // Locate all checkbox options ONLY inside the correct dropdown
    const options = page.locator('//button[contains(@class,
"multiselect")]/following-sibling::ul//li//label//input');

    // Assert the count of options
    await expect(options).toHaveCount(8);
});
```

Count the List Items Using JavaScript Array

Instead of XPath, you can use Playwright's `$$` to gather elements and check the count using a JavaScript array.

Example: Count List Items Using JS Array

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

test('Handle Bootstrap dropdown and count the list items using JS array',
async ({ page }) => {
    await page.goto('https://your-website-url.com'); // Replace with actual
URL

    // Click the dropdown button
    await page.locator('//button[contains(@class,
"multiselect")]').click();

    // Locate all checkbox options ONLY inside the correct dropdown
    const options = await page.$$('//button[contains(@class,
"multiselect")]/following-sibling::ul//li//label//input');

    // Assert the count of options
    await expect(options.length).toBe(8);
});
```

Select Multiple Options (e.g., Option 4 and Option 8)

You can loop through the available options and select multiple ones, such as "Option 4" and "Option 8".

Example: Select Multiple Options

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

test('Handle Bootstrap dropdown and select multiple options', async ({ page
}) => {
    await page.goto('https://your-website-url.com'); // Replace with actual
URL

    // Click the dropdown button
    await page.locator('//button[contains(@class,
"multiselect")]').click();

    // Locate all checkbox options ONLY inside the correct dropdown
    const options = await page.$$('//button[contains(@class,
"multiselect")]/following-sibling::ul//li//label//input');

    // Loop through the options and select "Option 4" and "Option 8"
    for (const option of options) {
        const value = await option.getAttribute('value'); // Get the value
attribute
        if (value === 'Option 4' || value === 'Option 8') {
            await option.click(); // Select the option
        }
    }
    // Assert the count of options
    await expect(options.length).toBe(8);
});
```

Select a Single Option (e.g., Option 4)

You can also select a single option, such as "Option 4", by matching its value and clicking it.

Example: Select a Single Option

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

test('Handle Bootstrap dropdown and select a single option', async ({ page
}) => {
    await page.goto('https://your-website-url.com'); // Replace with actual
URL

    // Click the dropdown button
    await page.locator('//button[contains(@class,
"multiselect")]').click();

    // Locate all checkbox options ONLY inside the correct dropdown
    const options = await page.$$('//button[contains(@class,
"multiselect")]/following-sibling::ul//li//label//input');

    // Loop through the options and select "Option 4"
    for (const option of options) {
        const value = await option.getAttribute('value'); // Get the value
attribute
        if (value == 'Option 4') {
            await option.click(); // Select the option
            break; // Stop after selecting the option
        }
    }

    // Assert the count of options
    await expect(options.length).toBe(8);
});
```