

Below is a markdown file that lists various ways in which an HTML DOM element can change every day, along with the corresponding XPath expressions to access these elements.

Handling Dynamic HTML DOM Elements with XPath

When dealing with HTML DOM elements that change daily, XPath expressions offer flexibility in accessing these elements. Here are some common scenarios in which elements and their values may change, along with the corresponding XPath code to access them.

1. Changing Element IDs

Scenario:

The ID of an element may change daily with a prefix or suffix added, e.g., `button_123`, `button_124`, etc.

XPath Solution:

```
//*[starts-with(@id, 'button_')]
```

2. Changing Element Class Names

Scenario:

The class attribute may have dynamic parts, e.g., `class="btn-primary day123"`, `class="btn-primary day124"`.

XPath Solution:

```
//*[contains(@class, 'btn-primary')]
```

3. Changing Element Text

Scenario:

The inner text of an element may vary daily, e.g., `<h1>Today's Special Offer</h1>`, `<h1>New Deals Available!</h1>`.

XPath Solution:

```
//h1[contains(text(), 'Special Offer') or contains(text(), 'Deals')]
```

4. Changing Attribute Values

Scenario:

An element's attribute value may change slightly, e.g., ``, ``.

XPath Solution:

```
//img[contains(@src, 'image_day')]
```

5. Changing Element Positions

Scenario:

The position of the element within the DOM may change daily, e.g., it might be the first item today and the third tomorrow.

XPath Solution:

```
//ul/li[position()=last()]
```

6. Changing Parent/Child Structure

Scenario:

The structure of parent and child elements may vary, e.g., `<div>Text</div>` today, and `<section><div>Text`

```
</div></section> tomorrow.
```

XPath Solution:

```
//*[text()='Text']
```

7. Conditional Attributes

Scenario:

An element might have different attributes depending on the condition, e.g., `<div data-active="true">Active</div>` vs. `<div data-active="false">Inactive</div>` .

XPath Solution:

```
//*[@data-active='true']
```

8. Changing Element Tags

Scenario:

The tag of the element might change, e.g., from `<div>` to `` .

XPath Solution:

```
//*[contains(text(),'Important Text')]
```

9. Dynamic Lists

Scenario:

A list of elements where items might be added or removed daily, e.g., `` tags inside an unordered list (``).

XPath Solution:

```
//ul/li[contains(text(), 'Item Text')]
```

10. Dynamic Data Attributes

Scenario:

Attributes like `data-*` might change, e.g., `<div data-day="1">` to `<div data-day="2">`.

XPath Solution:

```
//div[contains(@data-day, 'day')]
```

This markdown file lists common ways in which dynamic elements can change and provides corresponding XPath solutions to access them. Adjusting these XPath expressions based on the exact requirements can make your automation scripts more robust and adaptable to changes in the DOM structure.