Iframes

Iframes, or inline frames, are HTML elements that allow an external webpage to be embedded within a current page. Interacting with iframes in Playwright requires specialized locators to effectively retrieve elements from within the iframe context. This chapter delves into using the FrameLocator class for managing and interacting with iframes.

**FrameLocator Overview**

The FrameLocator represents a view to the iframe on the page, encapsulating the logic needed to retrieve the iframe and locate elements within it. You can create a FrameLocator using one of the following methods:

* locator.contentFrame()
* page.frameLocator()
* locator.frameLocator()

**Example**

To interact with a button inside an iframe:

javascript

Copy code

const locator = page.locator('#my-frame').contentFrame().getByText('Submit');

await locator.click();

**Strictness of Frame Locators**

FrameLocator instances are strict, meaning any operations performed on them will throw an error if multiple elements match the selector provided.

**Example**

The following code will throw an error if there are several frames in the DOM:

javascript

Copy code

// Throws if there are several frames in DOM:

await page.locator('.result-frame').contentFrame().getByRole('button').click();

To avoid this, you can specify which frame to interact with:

javascript

Copy code

// Works because we explicitly tell locator to pick the first frame:

await page.locator('.result-frame').contentFrame().first().getByRole('button').click();

**Converting between Locators**

You can convert a Locator object pointing to an iframe to a FrameLocator using locator.contentFrame().

Similarly, a FrameLocator can be converted back to a Locator using frameLocator.owner(), pointing to the same iframe.

**Methods for FrameLocator**

1. **frameLocator**
   * **Added in**: v1.17
   * **Usage**: Create a frame locator to enter the iframe and select elements within it.
   * **Syntax**:

javascript

Copy code

frameLocator.frameLocator(selector);

* + **Arguments**:
    - selector: A string selector used for resolving the DOM element.
  + **Returns**: FrameLocator

1. **getByAltText**
   * **Added in**: v1.27
   * **Usage**: Locate elements by their alt text.

javascript

Copy code

await page.getByAltText('Playwright logo').click();

* + **Arguments**:
    - text: Alt text for locating the element.
    - options: Optional object for exact match settings.
  + **Returns**: Locator

1. **getByLabel**
   * **Added in**: v1.27
   * **Usage**: Locate input elements by associated label text or aria-labelledby.

javascript

Copy code

await page.getByLabel('Username').fill('john');

* + **Arguments**: Similar to getByAltText.
  + **Returns**: Locator

1. **getByPlaceholder**
   * **Added in**: v1.27
   * **Usage**: Locate input elements by placeholder text.

javascript

Copy code

await page.getByPlaceholder('name@example.com').fill('playwright@microsoft.com');

* + **Arguments**: Similar to getByAltText.
  + **Returns**: Locator

1. **getByRole**
   * **Added in**: v1.27
   * **Usage**: Locate elements by ARIA role, attributes, and accessible names.

javascript

Copy code

await page.getByRole('button', { name: /submit/i }).click();

* + **Arguments**: Multiple options for precise role matching.
  + **Returns**: Locator

1. **getByTestId**
   * **Added in**: v1.27
   * **Usage**: Locate elements by test id.

javascript

Copy code

await page.getByTestId('directions').click();

* + **Arguments**: Similar to getByAltText.
  + **Returns**: Locator

1. **getByText**
   * **Added in**: v1.27
   * **Usage**: Locate elements containing specific text.

javascript

Copy code

page.getByText('world');

* + **Arguments**: Similar to getByAltText.
  + **Returns**: Locator

1. **getByTitle**
   * **Added in**: v1.27
   * **Usage**: Locate elements by their title attribute.

javascript

Copy code

await expect(page.getByTitle('Issues count')).toHaveText('25 issues');

* + **Arguments**: Similar to getByAltText.
  + **Returns**: Locator

1. **locator**
   * **Added in**: v1.17
   * **Usage**: Find an element matching the specified selector within the locator's subtree.

javascript

Copy code

frameLocator.locator(selectorOrLocator);

* + **Arguments**: Selector or locator to resolve the DOM element.
  + **Returns**: Locator

1. **owner**
   * **Added in**: v1.43
   * **Usage**: Returns a Locator pointing to the same iframe.

javascript

Copy code

const frameLocator = page.locator('iframe[name="embedded"]').contentFrame();

const locator = frameLocator.owner();

* + **Returns**: Locator

**Conclusion**

Using FrameLocator in Playwright allows for structured and efficient interaction with iframes. Understanding the methods and strictness associated with frame locators is crucial for seamless automation of complex web applications.