FrameLocator

FrameLocator represents a view to the `iframe` on the page. It captures the logic sufficient to retrieve the `iframe` and locate elements in that iframe. FrameLocator can be created with either page.frame_locator() or locator.frame_locator() method.

Sync **Async**

```
locator = page.frame_locator("my-frame").get_by_text("Submit")
locator.click()
```

Strictness

Frame locators are strict. This means that all operations on frame locators will throw if more than one element matches a given selector.

Sync **Async**

```
# Throws if there are several frames in DOM:
page.frame_locator('.result-frame').get_by_role('button').click()

# Works because we explicitly tell locator to pick the first frame:
page.frame_locator('.result-frame').first.get_by_role('button').click()
```

Converting Locator to FrameLocator

If you have a Locator object pointing to an `iframe` it can be converted to FrameLocator using `:scope` CSS selector:

Sync **Async**

```
frameLocator = locator.frame_locator(":scope")
```

Methods

frame_locator

Added in: v1.17

When working with iframes, you can create a frame locator that will enter the iframe and allow selecting elements in that iframe.

Usage

```
frame_locator.frame_locator(selector)
```

Arguments

- `selector` str

  A selector to use when resolving DOM element.

Returns

- FrameLocator

get_by_alt_text

Added in: v1.27

Allows locating elements by their alt text.

Usage

For example, this method will find the image by alt text "Playwright logo":

```
<img alt='Playwright logo'>
```

Sync **Async**

```
page.get_by_alt_text("Playwright logo").click()
```

Arguments

- `text` str|Pattern

  Text to locate the element for.

- `exact` bool *(optional)*

  Whether to find an exact match: case-sensitive and whole-string. Default to false. Ignored when locating by a regular expression. Note that exact match still trims whitespace.

Returns

- Locator

get_by_label

Added in: v1.27

Allows locating input elements by the text of the associated `<label>` or `aria-labelledby` element, or by the `aria-label` attribute.

Usage

For example, this method will find inputs by label "Username" and "Password" in the following DOM:

```
<input aria-label="Username">
<label for="password-input">Password:</label>
<input id="password-input">
```

Sync **Async**

```
page.get_by_label("Username").fill("john")
page.get_by_label("Password").fill("secret")
```

Arguments

- `text` str|Pattern

Text to locate the element for.

- `exact` bool *(optional)*

  Whether to find an exact match: case-sensitive and whole-string. Default to false. Ignored when locating by a regular expression. Note that exact match still trims whitespace.

Returns

- Locator

get_by_placeholder

Added in: v1.27

Allows locating input elements by the placeholder text.

Usage

For example, consider the following DOM structure.

```
<input type="email" placeholder="name@example.com" />
```

You can fill the input after locating it by the placeholder text:

Sync **Async**

```
page.get_by_placeholder("name@example.com").fill("playwright@microsoft.com")
```

Arguments

- `text` str|Pattern

  Text to locate the element for.

- `exact` bool *(optional)*

  Whether to find an exact match: case-sensitive and whole-string. Default to false. Ignored when locating by a regular expression. Note that exact match still trims whitespace.

Returns

- Locator

get_by_role

Added in: v1.27

Allows locating elements by their ARIA role, ARIA attributes and accessible name.

Usage

Consider the following DOM structure.

```
<h3>Sign up</h3>
<label>
  <input type="checkbox" /> Subscribe
</label>
<br/>
<button>Submit</button>
```

You can locate each element by it's implicit role:

Sync | Async

```
expect(page.get_by_role("heading", name="Sign up")).to_be_visible()

page.get_by_role("checkbox", name="Subscribe").check()

page.get_by_role("button", name=re.compile("submit", re.IGNORECASE)).click()
```

Arguments

- `role` "alert"|"alertdialog"|"application"|"article"|"banner"|"blockquote"|"button"|"caption"|"cell"|"checkbox"|"code"|"columnheader"|"combobox"|"complementary"|"contentinfo"|"definition"|"deletion"|"dialog"|"directory"|"document"|"emphasis"|"feed"|"figure"|"form"|"generic"|"grid"|"gridcell"|"group"|"heading"|"img"|"insertion"|"link"|"list"|"listbox"|"listitem"|"log"|"main"|"marquee"|"math"|"meter"|"menu"|"menubar"|"menuitem"|"menuitemcheckbox"|"menuitemradio"|"navigation"|"none"|"note"|"option"|"paragraph"|"presentation"|"progressbar"|"radio"|"radi

ogroup"|"region"|"row"|"rowgroup"|"rowheader"|"scrollbar"|"search"|"searchbox"|"separator"|"slider"|"spinbutton"|"status"|"strong"|"subscript"|"superscript"|"switch"|"tab"|"table"|"tablist"|"tabpanel"|"term"|"textbox"|"time"|"timer"|"toolbar"|"tooltip"|"tree"|"treegrid"|"treeitem"

Required aria role.

- `checked` bool *(optional)*

  An attribute that is usually set by `aria-checked` or native `<input type=checkbox>` controls.

  Learn more about `aria-checked`.

- `disabled` bool *(optional)*

  An attribute that is usually set by `aria-disabled` or `disabled`.

  > **NOTE**
  > Unlike most other attributes, `disabled` is inherited through the DOM hierarchy. Learn more about `aria-disabled`.

- `exact` bool *(optional)* Added in: v1.28

  Whether `name` is matched exactly: case-sensitive and whole-string. Defaults to false. Ignored when `name` is a regular expression. Note that exact match still trims whitespace.

- `expanded` bool *(optional)*

  An attribute that is usually set by `aria-expanded`.

  Learn more about `aria-expanded`.

- `include_hidden` bool *(optional)*

  Option that controls whether hidden elements are matched. By default, only non-hidden elements, as defined by ARIA, are matched by role selector.

  Learn more about `aria-hidden`.

- `level` int *(optional)*

  A number attribute that is usually present for roles `heading`, `listitem`, `row`, `treeitem`, with default values for `<h1>-<h6>` elements.

Learn more about `aria-level`.

- `name` str|Pattern *(optional)*

  Option to match the accessible name. By default, matching is case-insensitive and searches for a substring, use `exact` to control this behavior.

  Learn more about accessible name.

- `pressed` bool *(optional)*

  An attribute that is usually set by `aria-pressed`.

  Learn more about `aria-pressed`.

- `selected` bool *(optional)*

  An attribute that is usually set by `aria-selected`.

  Learn more about `aria-selected`.

Returns

- Locator

Details

Role selector **does not replace** accessibility audits and conformance tests, but rather gives early feedback about the ARIA guidelines.

Many html elements have an implicitly defined role that is recognized by the role selector. You can find all the supported roles here. ARIA guidelines **do not recommend** duplicating implicit roles and attributes by setting `role` and/or `aria-*` attributes to default values.

get_by_test_id

Added in: v1.27

Locate element by the test id.

Usage

Consider the following DOM structure.

```
<button data-testid="directions">Itinéraire</button>
```

You can locate the element by it's test id:

Sync | Async

```
page.get_by_test_id("directions").click()
```

Arguments

- `test_id` str|Pattern

  Id to locate the element by.

Returns

- Locator

Details

By default, the `data-testid` attribute is used as a test id. Use selectors.set_test_id_attribute() to configure a different test id attribute if necessary.

get_by_text

Added in: v1.27

Allows locating elements that contain given text.

See also locator.filter() that allows to match by another criteria, like an accessible role, and then filter by the text content.

Usage

Consider the following DOM structure:

```
<div>Hello <span>world</span></div>
<div>Hello</div>
```

You can locate by text substring, exact string, or a regular expression:

Sync **Async**

```
# Matches <span>
page.get_by_text("world")

# Matches first <div>
page.get_by_text("Hello world")

# Matches second <div>
page.get_by_text("Hello", exact=True)

# Matches both <div>s
page.get_by_text(re.compile("Hello"))

# Matches second <div>
page.get_by_text(re.compile("^hello$", re.IGNORECASE))
```

Arguments

- `text` str|Pattern

  Text to locate the element for.

- `exact` bool *(optional)*

  Whether to find an exact match: case-sensitive and whole-string. Default to false. Ignored when locating by a regular expression. Note that exact match still trims whitespace.

Returns

- Locator

Details

Matching by text always normalizes whitespace, even with exact match. For example, it turns multiple spaces into one, turns line breaks into spaces and ignores leading and trailing whitespace.

Input elements of the type `button` and `submit` are matched by their `value` instead of the text content. For example, locating by text `"Log in"` matches `<input type=button value="Log`

`in">`.

get_by_title

Added in: v1.27

Allows locating elements by their title attribute.

Usage

Consider the following DOM structure.

```
<span title='Issues count'>25 issues</span>
```

You can check the issues count after locating it by the title text:

Sync **Async**

```
expect(page.get_by_title("Issues count")).to_have_text("25 issues")
```

Arguments

- `text` str|Pattern

  Text to locate the element for.

- `exact` bool *(optional)*

  Whether to find an exact match: case-sensitive and whole-string. Default to false. Ignored when locating by a regular expression. Note that exact match still trims whitespace.

Returns

- Locator

locator

Added in: v1.17

The method finds an element matching the specified selector in the locator's subtree. It also accepts filter options, similar to locator.filter() method.

Learn more about locators.

Usage

```
frame_locator.locator(selector_or_locator)
frame_locator.locator(selector_or_locator, **kwargs)
```

Arguments

- `selector_or_locator` str|Locator

  A selector or locator to use when resolving DOM element.

- `has` Locator *(optional)*

  Matches elements containing an element that matches an inner locator. Inner locator is queried against the outer one. For example, `article` that has `text=Playwright` matches `<article><div>Playwright</div></article>`.

  Note that outer and inner locators must belong to the same frame. Inner locator must not contain FrameLocators.

- `has_not` Locator *(optional)* Added in: v1.33

  Matches elements that do not contain an element that matches an inner locator. Inner locator is queried against the outer one. For example, `article` that does not have `div` matches `<article><span>Playwright</span></article>`.

  Note that outer and inner locators must belong to the same frame. Inner locator must not contain FrameLocators.

- `has_not_text` str|Pattern *(optional)* Added in: v1.33

  Matches elements that do not contain specified text somewhere inside, possibly in a child or a descendant element. When passed a [string], matching is case-insensitive and searches for a substring.

- `has_text` str|Pattern *(optional)*

Matches elements containing specified text somewhere inside, possibly in a child or a descendant element. When passed a [string], matching is case-insensitive and searches for a substring. For example, `"Playwright"` matches `<article><div>Playwright</div></article>`.

Returns

- Locator

nth

Added in: v1.17

Returns locator to the n-th matching frame. It's zero based, `nth(0)` selects the first frame.

Usage

```
frame_locator.nth(index)
```

Arguments

- `index` int

Returns

- FrameLocator

Properties

first

Added in: v1.17

Returns locator to the first matching frame.

Usage

```
frame_locator.first
```

Returns

- FrameLocator

last

Added in: v1.17

Returns locator to the last matching frame.

Usage

```
frame_locator.last
```

Returns

- FrameLocator