Built-in Locators in Playwright

Playwright provides several built-in locators to find and interact with elements on a webpage efficiently. Below are the different types of built-in locators and their usage:

---

1. getByAltText() – Locating Elements Using the `alt` Attribute

- Generally used for images that have an `alt` attribute.
- If an element has an `alt` attribute, we can easily locate it using `getByAltText()`.

Example DOM:

```
<img src='/web/2e3ff' alt='company-branding'>
```

Playwright Code:

```
const logo = await page.getByAltText('company-branding');
await expect(logo).toBeVisible();
```

Purpose:

- Mostly used to check if an image is present on the webpage.

---

2. getByPlaceholder() – Locating Input Fields Using the `placeholder` Attribute

- Commonly used for input fields.
- If an element has a `placeholder` attribute, we can locate it using `getByPlaceholder()`.

Example DOM:

```
<input class='unamefield' type='password' placeholder='passplaceholder'>
```

Playwright Code:

```
await page.getByPlaceholder('passplaceholder').fill('testLife@123');
```

Purpose:

- Mainly used to input text into an input field.

---

3. getByRole() – Locating Actionable Elements by Their Role

- Actionable elements like buttons, links, input fields, and dropdowns can be identified using `getByRole()`.
- **Note:** `role` is not an attribute; it is inferred from the tag name.

Element	Role
`<button>`	`button`
`<a>`	`link`
`<input>`	`textbox`
`<select>`	`combobox`

Example DOM:

```
<button type="submit">Submit</button>
<a href="#">Click here</a>
<input type="text" placeholder="Enter name">
<select>
    <option value="1">Option 1</option>
    <option value="2">Option 2</option>
</select>
```

Playwright Code:

```
await page.getByRole('button', { type: 'Submit' }).click();
await page.getByRole('link').click();
await page.getByRole('textbox', { type: 'text' }).fill('yogesh');
await page.getByRole('combobox').click();
```

Purpose:

- **Purpose for Buttons:** Used to interact with buttons by clicking them.
- **Purpose for Links:** Used to click on hyperlinks (`<a>` elements).
- **Purpose for Textboxes:** Used to fill text into input fields.
- **Purpose for Dropdowns:** Used to interact with dropdown (`select`) elements.

4. getByText() – Locating Elements by Visible Text

Used to verify that specific text is displayed correctly on a webpage.

Example DOM:

```
<div id='user'>yogesh</div>
```

Playwright Code:

```
await expect(await page.getByText('yogesh')).toBeVisible();
```

For dynamic text:

```
const name = await page.locator("//input[@id='user']").textContent();
await expect(page.getByText(name)).toBeVisible();
```

---

5. getByLabel() – Locating Form Elements

- If an element has an associated `<label>`, we can locate it using `getByLabel()`.

Example DOM:

```
<label for="email">Email</label>
<input id="email" type="email">

<label for="password">Password</label>
<input id="password" type="password">
```

Playwright Code:

```
const emailField = await page.getByLabel('Email');
await emailField.fill('yogesh@example.com');

const passwordField = await page.getByLabel('Password');
await passwordField.fill('mySecurePass');
```

Check Visibility:

```
await expect(emailField).toBeVisible();
```

---

6. getByTitle() – Locating Elements Using the `title` Attribute

- If an element has a `title` attribute, we can locate it using `getByTitle()`.

Example DOM:

```
<button title="Submit Form">Submit</button>
<a href="#" title="Go to Home">Home</a>
<img src="logo.png" title="Company Logo">
```

Playwright Code:

```
const submitButton = await page.getByTitle('Submit Form');
await submitButton.click();

const homeLink = await page.getByTitle('Go to Home');
await homeLink.click();
```

Check Visibility:

```
await expect(submitButton).toBeVisible();
```

---

7. getByTestId() – Locating Elements Using `data-testid` Attribute

- If an element has a `data-testid` attribute, we can locate it using `getByTestId()`.

Example DOM:

```
<button data-testid="submit-btn">Submit</button>
<a href="#" data-testid="home-link">Home</a>
<input type="text" data-testid="username-input">
```

Playwright Code:

```
const submitButton = await page.getByTestId('submit-btn');
await submitButton.click();

const homeLink = await page.getByTestId('home-link');
await homeLink.click();
```

Check Visibility:

```
await expect(submitButton).toBeVisible();
```
