1.Session Storage

1. **Scope:** Session storage is scoped to the current page/tab and is not shared across tabs or windows.
2. **Lifetime:** The stored data only lasts for the duration of the page session. Once the tab or window is closed, the session storage is cleared.
3. **Manipulation:** You can directly interact with session storage by using JavaScript's sessionStorage API in Playwright to set, get, and delete key-value pairs.

**Playwright Example with Session Storage**

**Setting and Retrieving Session Storage**

In the following example, we demonstrate how to set and retrieve session storage in Playwright using the evaluate method:

typescript

Copy code

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

test('Manipulate session storage', async ({ page }) => {

// Navigate to the application page

await page.goto('https://example.com');

// Set a session storage item

await page.evaluate(() => {

sessionStorage.setItem('sessionKey', 'sessionValue');

});

// Retrieve the session storage item and verify its value

const sessionValue = await page.evaluate(() => sessionStorage.getItem('sessionKey'));

expect(sessionValue).toBe('sessionValue');

// Remove the session storage item

await page.evaluate(() => {

sessionStorage.removeItem('sessionKey');

});

// Verify that the item is removed

const removedSessionValue = await page.evaluate(() => sessionStorage.getItem('sessionKey'));

expect(removedSessionValue).toBeNull();

});