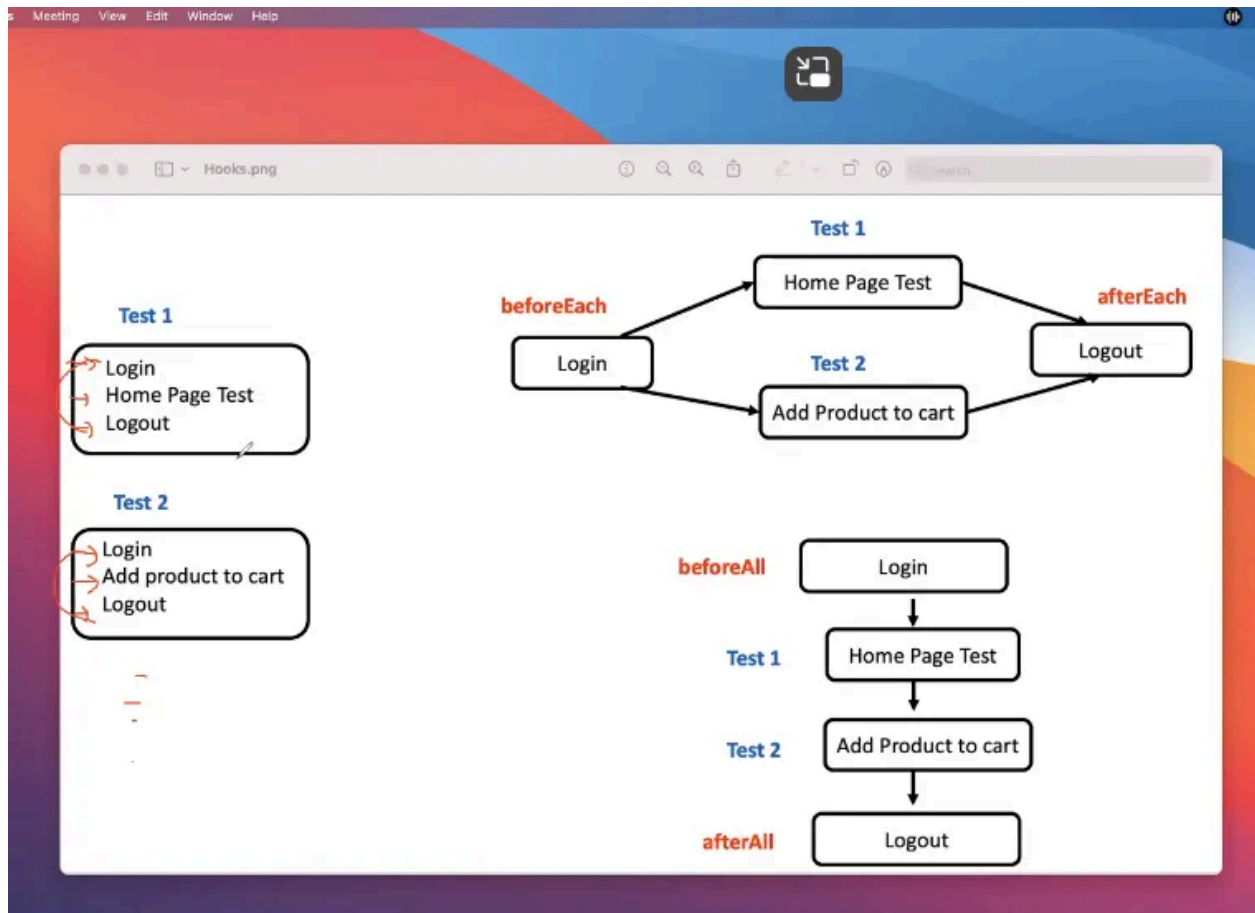


pavanpw-27

Hooks - beforeEach, afterEach, beforeAll & afterAll



none of the hooks accepts page as fixture. we need to use browser and then goto page level.
page fixture throws run time error.

pw config-

dont run in parallel.

```

9
10 /**
11  * @see https://playwright.dev/docs/test-configuration
12  */
13 module.exports = defineConfig({
14   testDir: './tests',
15   /* Run tests in files in parallel */
16   fullyParallel: false,
17   /* Fail the build on CI if you accidentally left test.only in
18   forbidOnly: !!process.env.CI,

```

change number of workers to 1-

```

21  /* Opt out of parallel tests on CI. */
22   workers: process.env.CI ? 1 : undefined,
23   /* Reporter to use. See https://playwright.dev/

```

alert can be written after the code which generates the dialog-

```

//Add product to cart await page.locator('//a[normalize-space()="Samsung galaxy s6"]').click()
await page.locator('//a[normalize-space()="Add to cart"]').click()

```

```

page.on('dialog', async dialog=>{ expect(dialog.message()).toContain('Product added.') await
dialog.accept() })

```

note-

we can add assertions in dialog also.

no need to use page fixture once declared as multiple instances will be created if we keep creating it.

```
1 //
2 import { test, expect } from '@playwright/test';
3
4 test('Home Page Test', async ({ page }) => {
5   await page.goto('https://www.demoblaze.com/index.html')
6   //Login
7   await page.locator('#login2').click()
8   await page.locator('#loginusername').fill('pavanol')
9   await page.locator('#loginpassword').fill('test@123')
10  await page.locator('//button[normalize-space()='Log in"]').click()
11
12  //Home Page
13  const products=await page.$$('.hrefch')
14  expect(products).toHaveLength(9)
15
16  //Logout
17  await page.locator('#logout2').click()
18 });
```

```
1 test('Add Product to cart Test', async ({ page }) => {
2   await page.goto('https://www.demoblaze.com/index.html')
3   //Login
4   await page.locator('#login2').click()
5   await page.locator('#loginusername').fill('pavano1')
6   await page.locator('#loginpassword').fill('test@123')
7   await page.locator('//button[normalize-space()='Log in"]').click()
8
9   //Add product to cart
10  await page.locator('//a[normalize-space()='Samsung galaxy s6"]').click()
11  await page.locator('//a[normalize-space()='Add to cart"]').click()
12
13  //we can add assertions in dialog also.
14  //dialog can also come after code which generated the dialog.
15  page.on('dialog', async dialog=>{
16    expect(dialog.message()).toContain('Product added.')
17    await dialog.accept()
18  })
19  //Logout
20  await page.locator('#logout2').click()
21
22 });
```

```

1  //
2  import { test, expect } from '@playwright/test';
3
4  let page;
5
6  test.beforeEach(async ({browser})=>{
7    page=await browser.newPage();
8    await page.goto('https://www.demoblaze.com/index.html')
9    //Login
10   await page.locator('#login2').click()
11   await page.locator('#loginusername').fill('pavano1')
12   await page.locator('#loginpassword').fill('test@123')
13   await page.locator('//button[normalize-space()='Log in"]').click()
14 });
15
16 test.afterEach(async()=>{
17   await page.locator('#logout2').click()
18 } )
19
20 test('Home Page Test', async () => {
21   const products=await page.$$('.hrefch')
22   expect(products).toHaveLength(9)
23 });
24
25
26 test('Add Product to cart Test', async () => {
27   await page.locator('//a[normalize-space()='Samsung galaxy s6"]').click()
28   await page.locator('//a[normalize-space()='Add to cart"]').click()
29
30   page.on('dialog', async dialog=>{
31     expect(dialog.message()).toContain('Product added.')
32     await dialog.accept()
33   })
34 });

```

```

1  //
2  import { test, expect } from '@playwright/test';
3
4  let page;
5
6  test.beforeAll(async ({browser})=>{
7    page=await browser.newPage();
8    await page.goto('https://www.demoblaze.com/index.html')
9    //Login
10   await page.locator('#login2').click()
11   await page.locator('#loginusername').fill('pavano1')
12   await page.locator('#loginpassword').fill('test@123')
13   await page.locator('//button[normalize-space()='Log in"]').click()
14 });
15
16 test.afterAll(async()=>{
17   await page.locator('#logout2').click()
18 } )
19
20 test('Home Page Test', async () => {
21   const products=await page.$$('.hrefch')
22   expect(products).toHaveLength(9)
23 });
24
25
26 test('Add Product to cart Test', async () => {
27   await page.locator('//a[normalize-space()='Samsung galaxy s6"]').click()
28   await page.locator('//a[normalize-space()='Add to cart"]').click()
29
30   page.on('dialog', async dialog=>{
31     expect(dialog.message()).toContain('Product added.')
32     await dialog.accept()
33   })
34 });

```