

Just change line 2 for all files-

```
login.spec.ts X
mytest > login.spec.ts > ...
1 import {test, expect, type Browser, type Page, type Locator} from '@playwright/test'
2 import {webkit, chromium, firefox} from '@playwright/test'
```

```
login.spec.ts X login2.spec.ts login3.spec.ts login4.spec.ts login5.spec.ts
mytest > login.spec.ts > test('login') callback
1 import {test, expect, type Browser, type Page, type Locator} from '@playwright/test'
2 import {chromium} from '@playwright/test'
3
4 test('login', async() => {
5
6     //open browser.
7     const browser:Browser=await chromium.launch({headless:false})
8     //open page.
9     const page:Page=await browser.newPage()
10    //open url.
11    await page.goto("https://naveenautomationlabs.com/opencart/index.php?route=account/login")
12
13    //enter email, password, click login.
14    //used css locators.
15    const emailId:Locator=await page.locator("#input-email")
16    const passwordText:Locator=await page.locator("#input-password")
17    const loginButton:Locator=await page.locator("[value='Login']")
```

```
17    const loginButton:Locator=await page.locator("[value='Login']")
18
19    //enter the credentials.
20    //click login.
21    await emailId.fill("pwtest@opencart.com")
22    await passwordText.fill("playwright@123")
23    await loginButton.click()
24
25    //wait for page title.
26    const title=await page.title()
27    console.log(`home page title is ${title}`)
28
29    //take screenshot and give path where to save.
30    //it will be saved in root directory of project.
31    await page.screenshot({path: 'homepage.png'})
```

```

31     await page.screenshot({path: 'homepage.png'})
32
33     //asset title.
34     expect(title).toEqual('Account Login')
35
36     //close browser.
37     browser.close()
38 });

```

We can use `@playwright/test` for both.
For browsers only playwright also works.

```

login2.spec.ts x login3.spec.ts login4.spec.ts login5.spec.ts
mytest > login2.spec.ts > test('login') callback
1 // import {test, expect, type Browser, type Page, type Locator} from 'playwright'
2 //here only playwright wont work.
3 //Module '"playwright"' has no exported member 'test'.ts(2305)
4 // Module '"playwright"' has no exported member 'expect'.ts(2305)
5
6 import {test, expect, type Browser, type Page, type Locator} from '@playwright/test'
7 import {chromium} from 'playwright' //simply writing playwright also works
8
9 test('login', async() => {
10
11     //open browser.
12     const browser:Browser=await chromium.launch({headless:false})
13     //open page.
14     const page:Page=await browser.newPage()
15     //open url.
16     await page.goto("https://naveenautomationlabs.com/opencart/index.php?route=account/login")
17
18     await page.goto("https://naveenautomationlabs.com/opencart/index.php?route=account/login")
19
20     //enter email, password, click login.
21     //used css locators.
22     const emailId:Locator=await page.locator("#input-email")
23     const passwordText:Locator=await page.locator("#input-password")
24     const loginButton:Locator=await page.locator("[value='Login']")
25

```

```

21  const passwordTextLocator=await page.locator('#input-password')
22  const loginButton:Locator=await page.locator("[value='Login']")
23
24  //enter the credentials.
25  //click login.
26  await emailId.fill("pwtest@opencart.com")
27  await passwordText.fill("playwright@123")
28  await loginButton.click()
29
30  //wait for page title.
31  const title=await page.title()
32  console.log(`home page title is ${title}`)
33
34  //take screenshot and give path where to save.
35  //it will be saved in root directory of project.
36  await page.screenshot({path: 'homepage.png'})
37

```

```

36  await page.screenshot({path: 'homepage.png'})
37
38  //asset title.
39  expect(title).toEqual('Account Login')
40
41  //close browser.
42  browser.close()
43  });

```

Write await before every js or ts line else in report we get error-
Browser.close also needs await.

```
login3.spec.ts login4.spec.ts login5.spec.ts X
mytest > login5.spec.ts > test('login') callback
1 import {test, expect, type Browser, type Page, type Locator, webkit} from '@playwright/test'
2 import {chromium} from 'playwright' //simply writing playwright also works
3
4 test('login', async() => {
5
6     //open browser.
7     //to run in real chrome not chromium engine.
8     const browser:Browser=await chromium.launch({headless:false, channel:'chrome'})
9     //open page.
10    const page:Page=await browser.newPage()
11    //open url.
12    await page.goto("https://naveenautomationlabs.com/opencart/index.php?route=account/login")
13
14    //enter email, password, click login.
15    //used css locators.
16    const emailId:Locator=await page.locator("#input-email")
17    const passwordText:Locator=await page.locator("#input-password")
18    const loginButton:Locator=await page.locator("[value='Login']")
19
```

```
18    const loginButton:Locator=await page.locator("[value='Login']")
19
20    //enter the credentials.
21    //click login.
22    await emailId.fill("pwtest@opencart.com")
23    await passwordText.fill("playwright@123")
24    await loginButton.click()
25
26    //wait for page title.
27    const title=await page.title()
28    console.log(`home page title is ${title}`)
29
30    //take screenshot and give path where to save.
31    //it will be saved in root directory of project.
32    await page.screenshot({path: 'homepage.png'})
33
```

```
32    await page.screenshot({path: 'homepage.png'})
33
34    //asset title.
35    expect(title).toEqual('Account Login')
36
37    //close browser.
38    //without await any ts or js lines give error in report.
39    await browser.close()
40    });
```

To run in webkit safari-

```
login3.spec.ts x login4.spec.ts
mytest > login3.spec.ts > test('login') callback
1 import {test, expect, type Browser, type Page, type Locator, webkit} from '@playwright/test'
2 import {chromium} from 'playwright' //simply writing playwright also works
3
4 test('login', async() => {
5
6     //open browser.
7     //open safari browser.
8     const browser:Browser=await webkit.launch({headless:false})
9     //open page.
10    const page:Page=await browser.newPage()
11    //open url.
12    await page.goto("https://naveenautomationlabs.com/opencart/index.php?route=account/login")
13
14    //enter email, password, click login.
15    //used css locators.
16    const emailId:Locator=await page.locator("#input-email")
17    const passwordText:Locator=await page.locator("#input-password")
18    const loginButton:Locator=await page.locator("[value='Login']")
19
20    const loginButton:Locator=await page.locator("[value='Login']")
21
22    //enter the credentials.
23    //click login.
24    await emailId.fill("pwtest@opencart.com")
25    await passwordText.fill("playwright@123")
26    await loginButton.click()
27
28    //wait for page title.
29    const title=await page.title()
30    console.log(`home page title is ${title}`)
31
32    //take screenshot and give path where to save.
33    //it will be saved in root directory of project.
34    await page.screenshot({path: 'homepage.png'})
35
36    //asset title.
37    expect(title).toEqual('Account Login')
38
39    //close browser.
40    browser.close()
41});
```

To run in real chrome browser not chromium engine-

```
... login4.spec.ts X
mytest > login4.spec.ts > test('login') callback
1 import {test, expect, type Browser, type Page, type Locator, webkit} from '@playwright/test'
2 import {chromium} from 'playwright' //simply writing playwright also works
3
4 test('login', async() => {
5
6     //open browser.
7     //to run in real chrome not chromium engine.
8     const browser:Browser=await chromium.launch({headless:false, channel:'chrome'})
9     //open page.
10    const page:Page=await browser.newPage()
11    //open url.
12    await page.goto("https://navveenautomationlabs.com/opencart/index.php?route=account/login")
13
14    //enter email, password, click login.
15    //used css locators.
16    const emailId:Locator=await page.locator("#input-email")
17    const passwordText:Locator=await page.locator("#input-password")
18    const loginButton:Locator=await page.locator("[value='Login']")
19
```

```
18    const loginButton:Locator=await page.locator("[value='Login']")
19
20    //enter the credentials.
21    //click login.
22    await emailId.fill("pwtest@opencart.com")
23    await passwordText.fill("playwright@123")
24    await loginButton.click()
25
26    //wait for page title.
27    const title=await page.title()
28    console.log(`home page title is ${title}`)
29
30    //take screenshot and give path where to save.
31    //it will be saved in root directory of project.
32    await page.screenshot({path:'homepage.png'})
33
```

```

31 //it will be saved in root directory of project.
32 await page.screenshot({path: 'homepage.png'})
33
34 //asset title.
35 expect(title).toEqual('Account Login')
36
37 //close browser.
38 browser.close()
39 });

```

Update directory in pw config-
We need to run test from this directory only.

```

9 /**
10  * See https://playwright.dev/
11  */
12 export default defineConfig({
13   testDir: './mytest',
14   /* Run tests in files in parallel
15   fullyParallel: true

```