pavanpw-36

Reporters

we can change the report we want either in pw config or in cmd line.

built in reporters-
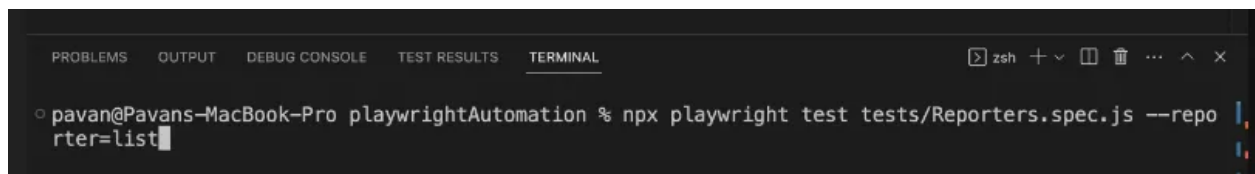
list reporter-

set it directly without pw config on cmd-

tick marks in report.
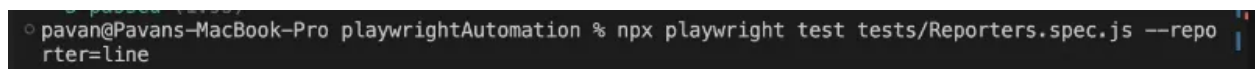
line reporter-

nothing in reports.

dot reporter-

..F

F means fail

.. means pass
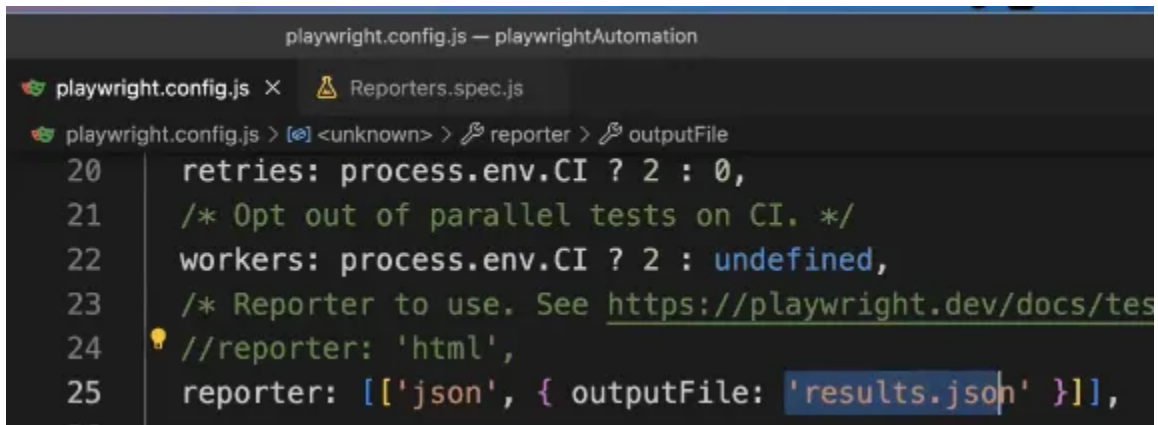
html reporter-

default report which we get, see it either in pw report folder or using cmd.

json reporter-

slightly different in config file.
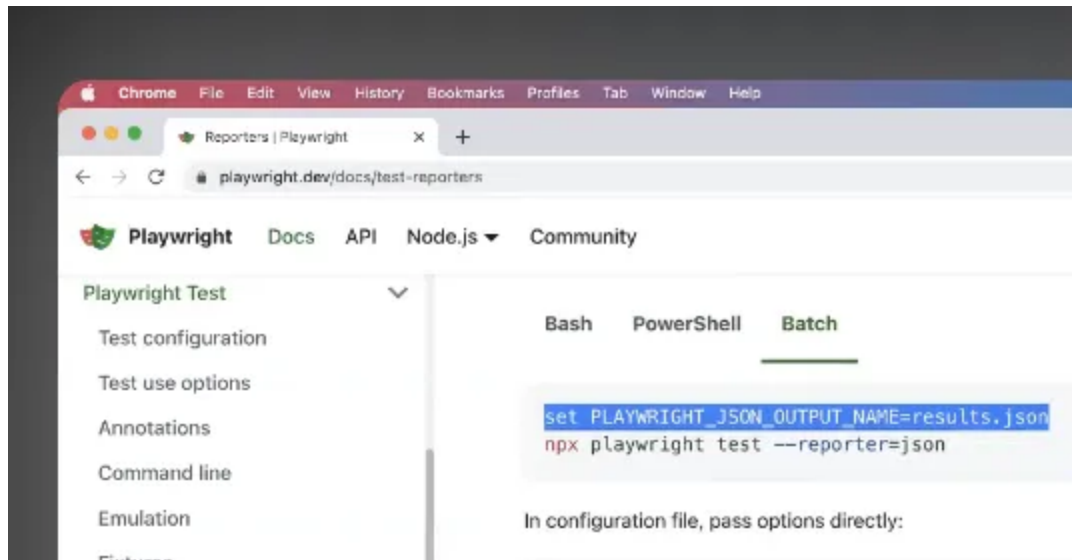

output is created at project root level.

will generate json in command line itself as we didnt specify path.-

to generate in file-

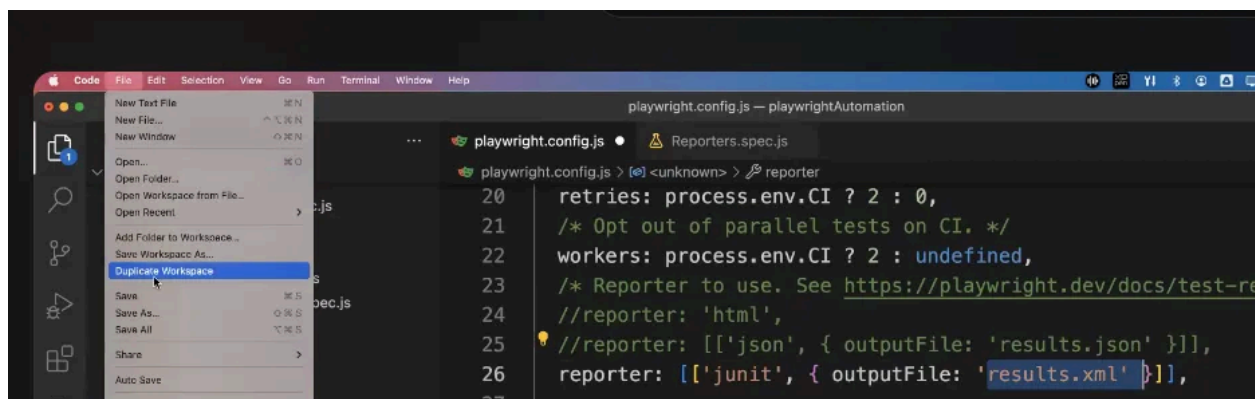first we need to run bash command and then the report generation command.
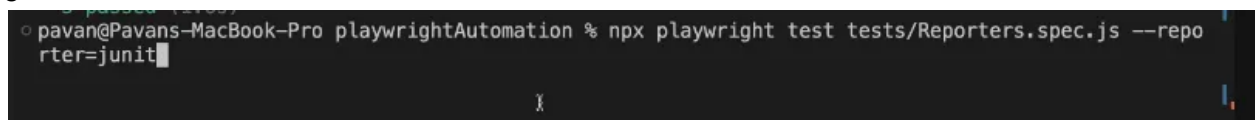
junit reporter-
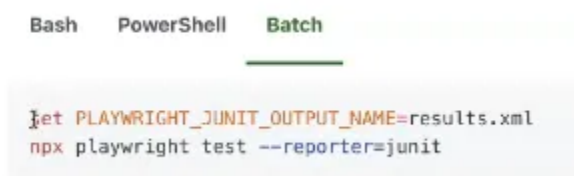

instead of json, xml is generated.

generated in console.-

set the bash command for specific location-

add multiple reporters-

playwright.config.js 8 ●   ⚗ Reporters.spec.js

playwright.config.js > [◉] <unknown> > 🔧 reporter

```js
24    //reporter: 'html',
25    //reporter: [['json', { outputFile: 'results.json' }]],
26    //reporter: [['junit', { outputFile: 'results.xml' }]],
27
28    reporter: [['list'],
29               ['html'],
30               ['junit', { outputFile: 'results.xml' }],
31               [['json', { outputFile: 'results.json' }]
32              ],
33
```

```js
1   //
2   const { test, expect } = require('@playwright/test');
3
4   test('Test1', async ({ page }) ⇒ {
5       await page.goto('https://www.demoblaze.com/index.html');
6       await expect(page).toHaveTitle('STORE');
7   })
8
9   test('Test2', async ({ page }) ⇒ {
10      await page.goto('https://demo.opencart.com/');
11      await expect(page).toHaveTitle('Your Store');
12  })
13
14
15  test('Test3', async ({ page }) ⇒ {
16      await page.goto('https://demo.nopcommerce.com/');
17      await expect(page).toHaveTitle('nopCommerce demo store');
18  })
```

codesnap.dev