

In the playwright, we can Create tests in two different ways **first Approach** by using code gen plugin provided by playwright also called Code generator can record and **Second Approach** is we can write our own scripts.

To run the Particular Script:

```
npx playwright test homePageTest.spec.js
```

To run the Particular Script on a Particular Browser:

```
npx playwright test homePage.spec.js --project=chromium
```

To show the Report:

```
npx playwright show-report
```

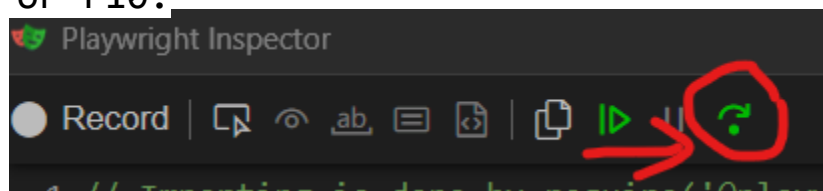
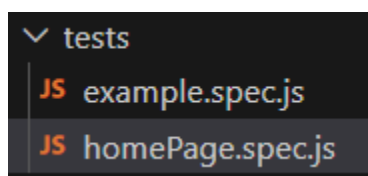
To run the Particular Script in headed Mode:

```
npx playwright test homePage.spec.js --project=chromium  
--headed
```

If you want to test script one by one debug mode:

```
npx playwright test homePage.spec.js --project=chromium  
--headed --debug
```

Which open two windows one is browser and other is playwright inspector. We can run the script lines one by one using step over or F10.



```
homePage.spec.js X
tests > JS homePage.spec.js > ...
1 // Importing is done by require('@playwright/test') and saved in test and expect
2 const {test, expect} = require('@playwright/test')
3 // page contains many functions
4 /* Anonomous function(Arrow function): () => {}
5 async before page and await before page methods are used throughout automation
6 Javascript is asynchronised language execute in asynchronized manner means it does
7 not wait for previous line means parallelly all the lines are executed
8 to handle this we use the javaScript Promise
9 async returns a promise
10 await makes sure wait for the promise*/
11 test("Home Page", async ({page}) => {
12     await page.goto("https://www.demoblaze.com/index.html") - 3563ms
13     const ptitle = await page.title() - 10ms
14     console.log("Page Title: " + ptitle)
15     const purl = await page.url()
16     console.log("Page URL: " + purl)
17     await expect(page).toHaveTitle("STORE") - 20ms
18     await expect(page).toHaveURL("https://www.demoblaze.com/index.html") - 5ms
19     await page.close() - 2ms
20 })
```