

One liner code to drag and drop-

```
dragdrop1.spec.ts x dragdrop2.spec.ts
mytest > dragdrop1.spec.ts > test('drag and drop ') callback
1 import {test, expect, Browser, Page, Locator, FrameLocator} from '@playwright/test'
2 import { log } from 'console';
3 import { webkit, chromium, firefox } from 'playwright'
4
5 test('drag and drop ', async()=>{
6
7     const browser:Browser = await chromium.launch({headless: false, channel: 'chrome'});
8     const page:Page = await browser.newPage();
9
10    //go to url.
11    await page.goto("https://jqueryui.com/resources/demos/droppable/default.html");
12
13    //one liner code.
14    //single:
15    //source locator from where to drag.
16    //drag to method to be used.
17    //destination locator where to drop.
18    await page.locator("#draggable").dragTo(page.locator("#droppable"));
19
20    await page.waitForTimeout(3000)
21
22 })
```

Using mouse actions-

... dragdrop2.spec.ts X

mytest > dragdrop2.spec.ts > test('drag and drop ') callback

```
1 import {test, expect, Browser, Page, Locator, FrameLocator} from '@playwright/test'
2 import { log } from 'console';
3 import { webkit, chromium, firefox } from 'playwright'
4
5 test('drag and drop ', async()=>{
6
7     const browser:Browser = await chromium.launch({headless: false, channel: 'chrome'});
8     const page:Page = await browser.newPage();
9
10    //go to url.
11    await page.goto("https://jqueryui.com/resources/demos/droppable/default.html");
12
13    //multiple commands:
14    //manual steps.
15
16    //first hover over the source element.
17    await page.locator("#draggable").hover();
18
19    //then press the mouse button down.
20    await page.mouse.down();
21
22    //then hover over the destination element.
23    await page.locator("#droppable").hover();
24
25    //then release the mouse button.
26    await page.mouse.up();
27
28    await page.waitForTimeout(3000)
29
30 })
```