Contents

[**Step 1: Create a project** 2](#_Toc183970647)

[**Step 2: Use the dev dependencies** 2](#_Toc183970648)

[**Step 3: To install packages in the package.json** 2](#_Toc183970649)

[**Step 5: To install playwright browsers** 2](#_Toc183970650)

[**Step 6: To create a tsconfig.json file automatically in your project** 3](#_Toc183970651)

[**Step 7: Configure the Jest in package.json** 3](#_Toc183970652)

[**Step 8: Configure test command in package.json** 3](#_Toc183970653)

[**Handling dropdown:** 4](#_Toc183970654)

[**Handling Windows:** 4](#_Toc183970655)

[**Handling Multiple Windows:** 5](#_Toc183970656)

[**Handling Multiple Frames:** 6](#_Toc183970657)

# **Step 1: Create a project**

Command: npm init -y

# **Step 2: Use the dev dependencies**

"devDependencies": {

"@types/jest": "^29.5.14",

"jest": "^29.7.0",

"jest-playwright-preset": "^4.0.0",

"playwright": "^1.48.2",

"ts-jest": "^29.2.5",

"typescript": "^5.6.3"

}

or using this:

npm install @types/jest@latest --save-dev

npm install jest@latest --save-dev

npm install jest-playwright-preset@latest --save-dev

npm install playwright@latest --save-dev

npm install typescript@latest --save-dev

npm install ts-jest@latest --save-dev

# **Step 3: To install packages in the package.json**

Command: npm i

**Step 4: Open Terminal using short cut**

Command: Control + shift + P

# **Step 5: To install playwright browsers**

Command: **Install playwright**

# **Step 6: To create a tsconfig.json file automatically in your project**

Command: **npx tsc --init**

# **Step 7: Configure the Jest in package.json**

"jest": {

"transform": {

"^.+\\.tsx?$": "ts-jest"

},

"moduleFileExtensions": [

"ts",

"tsx",

"js",

"jsx",

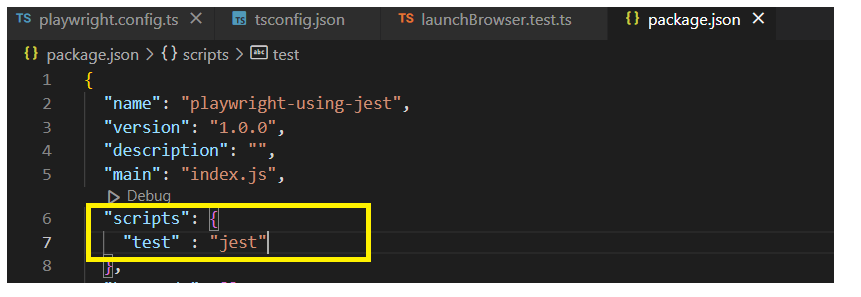
"json",

"node"

]

}

# **Step 8: Configure test command in package.json**



**Step 9: Run using command**

Command: **npm test**

# **Examples**

## **Handling dropdown:**

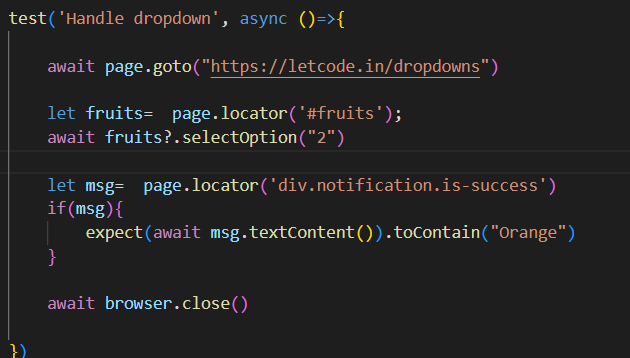
Where,

await fruits?.selectOption("2")

? is optional chaining used to handle null value

**fruits?**: The optional chaining (?.) checks if fruits is null or undefined before attempting to call selectOption.

**.selectOption("2")**: This function is typically used to select an option in a dropdown. "2" likely represents the value of the option you want to select.



## **Handling Windows:**

test('Test single page', async ()=>{

        await page.goto("https://letcode.in/windows")

        const [newWindow] = await Promise.all([

            context.waitForEvent("page"),

            await page.click("#home")

       ])

       await newWindow.waitForLoadState()

       expect(newWindow.url()).toContain("test")

       await newWindow.getByText("Log in").click()

       //await newWindow.waitForNavigation({ timeout: 60000 })

       expect(newWindow.url()).toContain("signin")

       await page.bringToFront()

       await page.click("#testing")

       await context.close(); // Ensure context is closed properly

       await browser.close();

    }, 60000)

})

## **Handling Multiple Windows:**

 test('Test multiple page', async ()=>{

        await page.goto("https://letcode.in/windows")

        const [multipleWindow] = await Promise.all([

            context.waitForEvent("page"),

            await page.click("#multi")

       ])

       await multipleWindow.waitForLoadState()

**const allwindows=multipleWindow.context().pages()**

       console.log("No of windows : " +  allwindows.length)

**allwindows.forEach((page: Page) => {**

        console.log("No of pages : " + page.url())

       });

    })

## **Handling Multiple Frames:**

    test("Check frames" , async ()=>{

        const frame=**page.frame({name: "firstFr"});**

        if(frame!= null)

        {

            await frame.fill("input[name='fname']", "Sumesh")

            await frame.fill("input[name='lname']", "m s")

            const frames **= frame.childFrames()**

            console.log("No. of inner frames :" +  frames.length)

            if(frames!= null)

                await frames[1].fill("input[name='email']", "Sumeshattec@gmail.com")

            else

                console.log("Wrong frame " )

            const parent **= frames[1].parentFrame()**

            await parent?.fill("input[name='lname']", "Manager")

        }

        else

            throw new Error("no such frames")

    })

## **Handling Multiple Elements:**

**Program**:

beforeAll(async() =>{

        browser= await chromium.launch({

            headless: false,

            args: ["--start-maximized"],

        })

        context= await browser.newContext({ viewport: null })

        page=await context.newPage()

        await page.goto("https://letcode.in/elements")

    })

    test("type git name" ,async () =>{

       const gitname=  page.locator("input[name='username']")

       await gitname?.fill("sumesh87")

       await gitname?.press("Enter")

    })

    test("Print all names from list" ,async () =>{

        await page.waitForSelector("app-gitrepos ol li", {timeout:5000})

        // locator.all() - to get list of locators

        const repos =  await page.locator("app-gitrepos ol li").all()

        console.log(repos.length);

        // use for-await to wait for promise to be resolved

        for await (const repo of repos)

        {

            console.log(await repo.innerText());

        }

        // Method 2: using map

        console.log("Method 2: ");

        const allUrls =  await Promise.all( repos.map(async (repo, i)=> {

            return await repo.innerText()

        }))

        console.log(allUrls);

    })

**O/p:**

at test/MultipleElements.test.ts:38:17

console.log

https://github.com/Sumesh87/AppiumTesting

at test/MultipleElements.test.ts:43:21

console.log

https://github.com/Sumesh87/cucumber\_Bdd

at test/MultipleElements.test.ts:43:21

console.log

https://github.com/Sumesh87/Cypress

at test/MultipleElements.test.ts:43:21

console.log

https://github.com/Sumesh87/seleniumPython

at test/MultipleElements.test.ts:43:21

**Method 2:**

console.log

[

'https://github.com/Sumesh87/AppiumTesting',

'https://github.com/Sumesh87/cucumber\_Bdd',

'https://github.com/Sumesh87/Cypress',

'https://github.com/Sumesh87/seleniumPython'

]