.NET Conf China
2022

Playwright探索

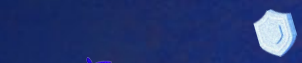
骆姜斌
智鹏瑞尔软件 (SSW China) – 解决方案架构师
FireUG技术社区组织者之一

FireUG

B站 抖音 公众号 微信群

目录

什么是Playwright？



什么是Playwright

- 微软开源的自动化的Web测试框架
  - 部分团队成员曾属于Puppeteer团队
  - Puppeteer - PuppeteerSharp
    - Playwright - https://github.com/microsoft/playwright
    - Playwright for .NET - https://github.com/microsoft/playwright-dotnet
    - Playwright for Java – https://github.com/microsoft/playwright-java
    - Playwright for Python - https://github.com/microsoft/playwright-python
- 主流浏览器支持
  - Chromium
  - Firefox
  - WebKit
- 项目活跃度
  - 48小时内响应Issue
  - 已解决 7000+ Issue
  - 每月多次发布

附带的工具

- Codegen
  - 脚本录制工具
- Playwright Inspector
  - 代码开发和调试的GUI工具
- **Trace Viewer**
  - 查看脚本执行过程的GUI工具
  - 包含页面截图, 脚本执行以及执行前后的相关信息

Playwright优点

- 一键安装, 自动下载浏览器
- Headless模式下允许录制视频
- 可以根据需要干涉修改请求
- 模拟各种设备
- 代码生成
- 便捷的调试方式

Playwright VS Selenium

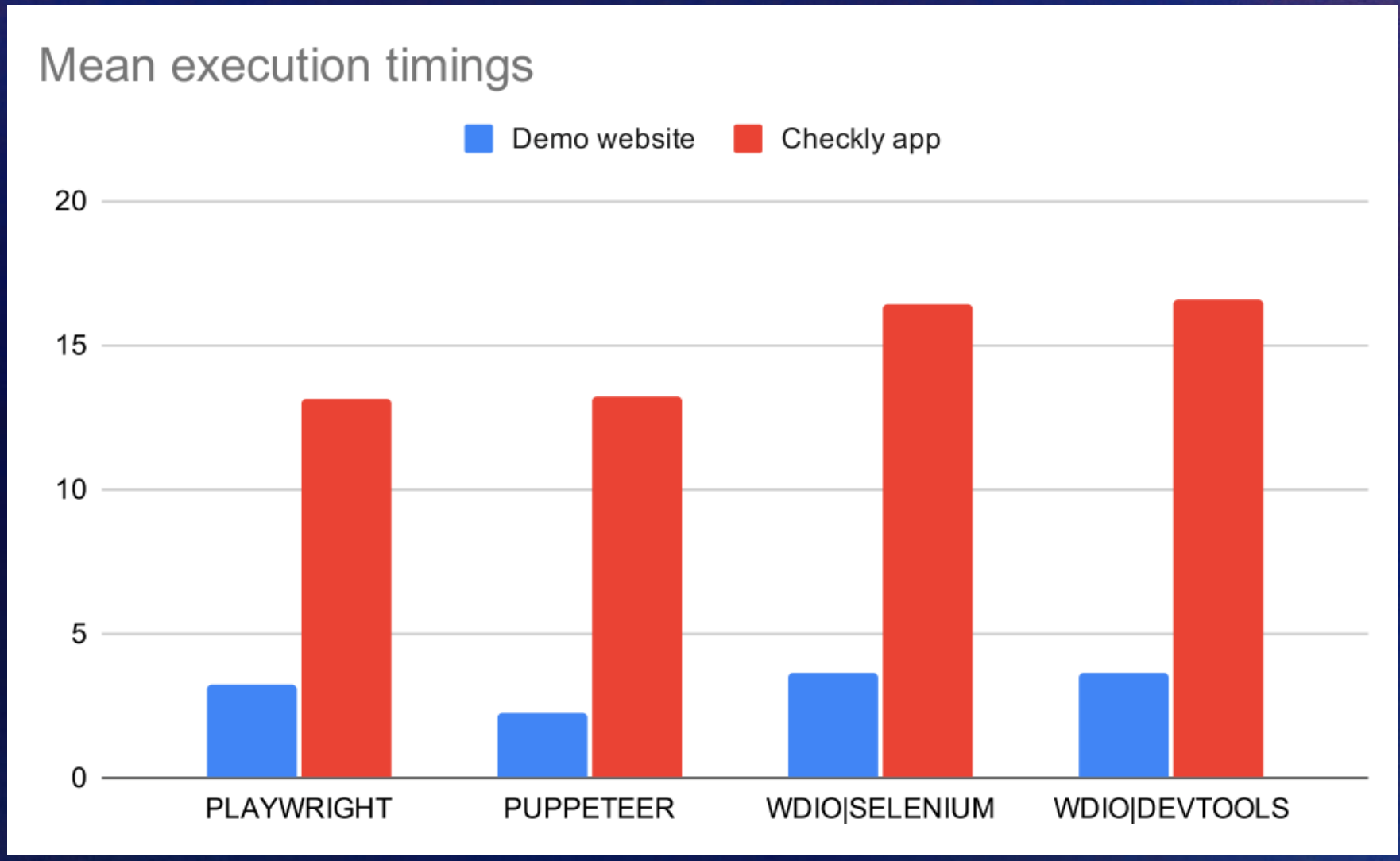
开发语言

Playwright	Selenium
C#	C#
JavaScript	JavaScript
Java	Java
Python	Python
TypeScript	Ruby

浏览器

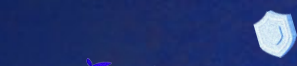
Playwright	Selenium
Chromium	Chromium
Firefox	Firefox
WebKit	WebKit
	IE

性能



https://blog.checklyhq.com/puppeteer-vs-selenium-vs-playwright-speed-comparison/#methodology-or-how-

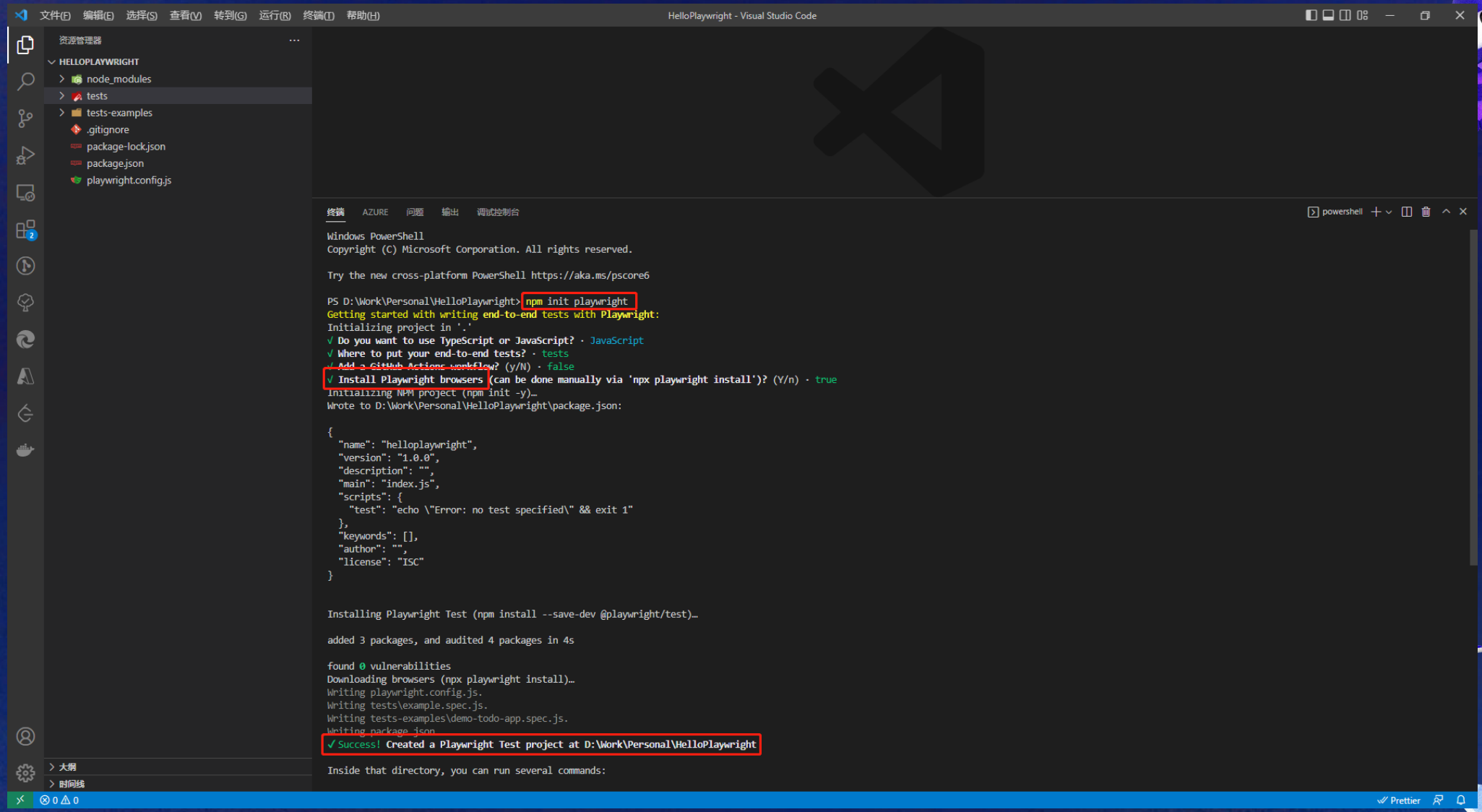
如何使用 (NodeJS)



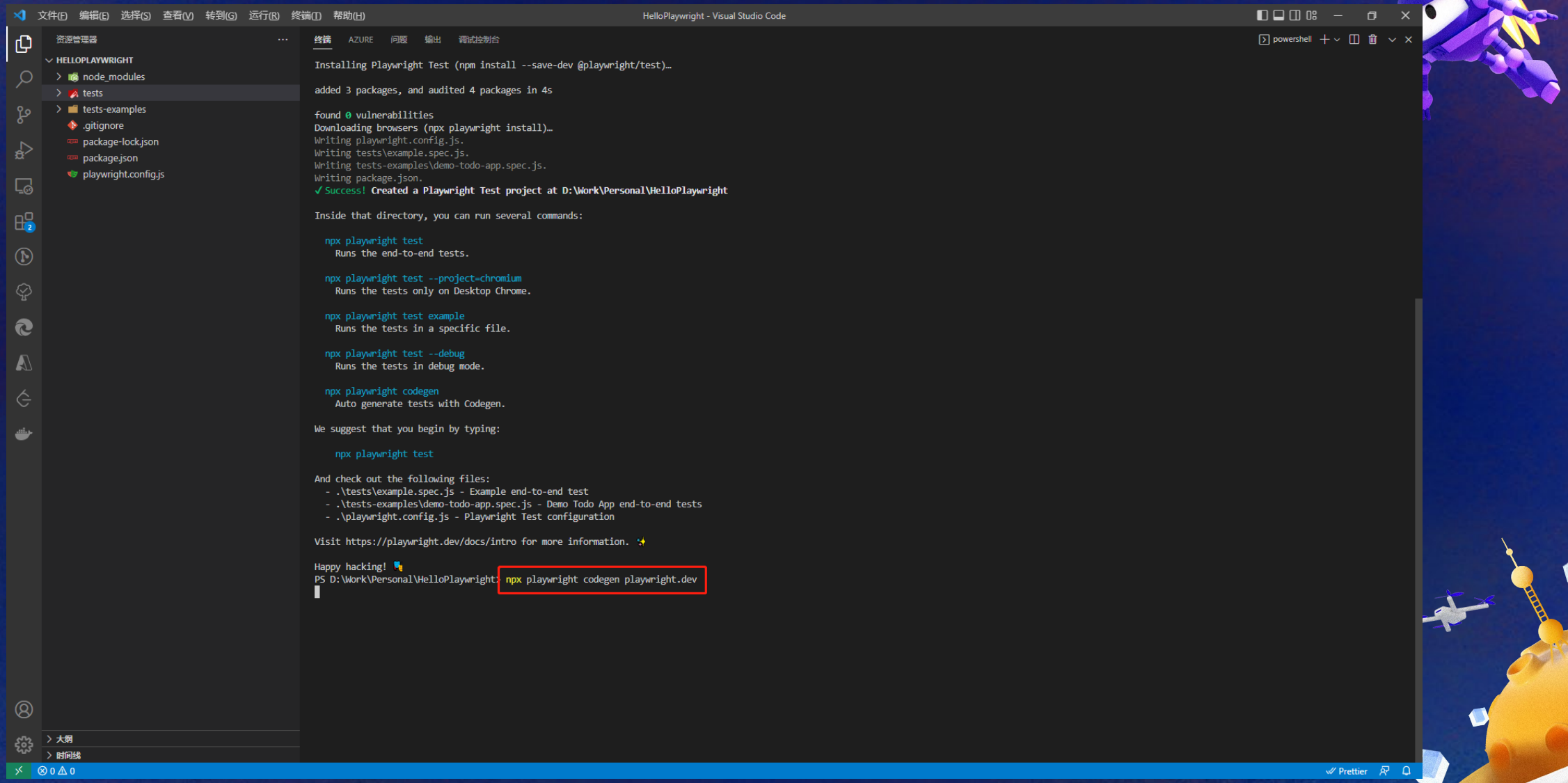
如何使用 - 环境准备 (NodeJS)

- NodeJS
  - v14以上
- 代码编辑器
  - VSCode
- Playwright
  - npm install playwright

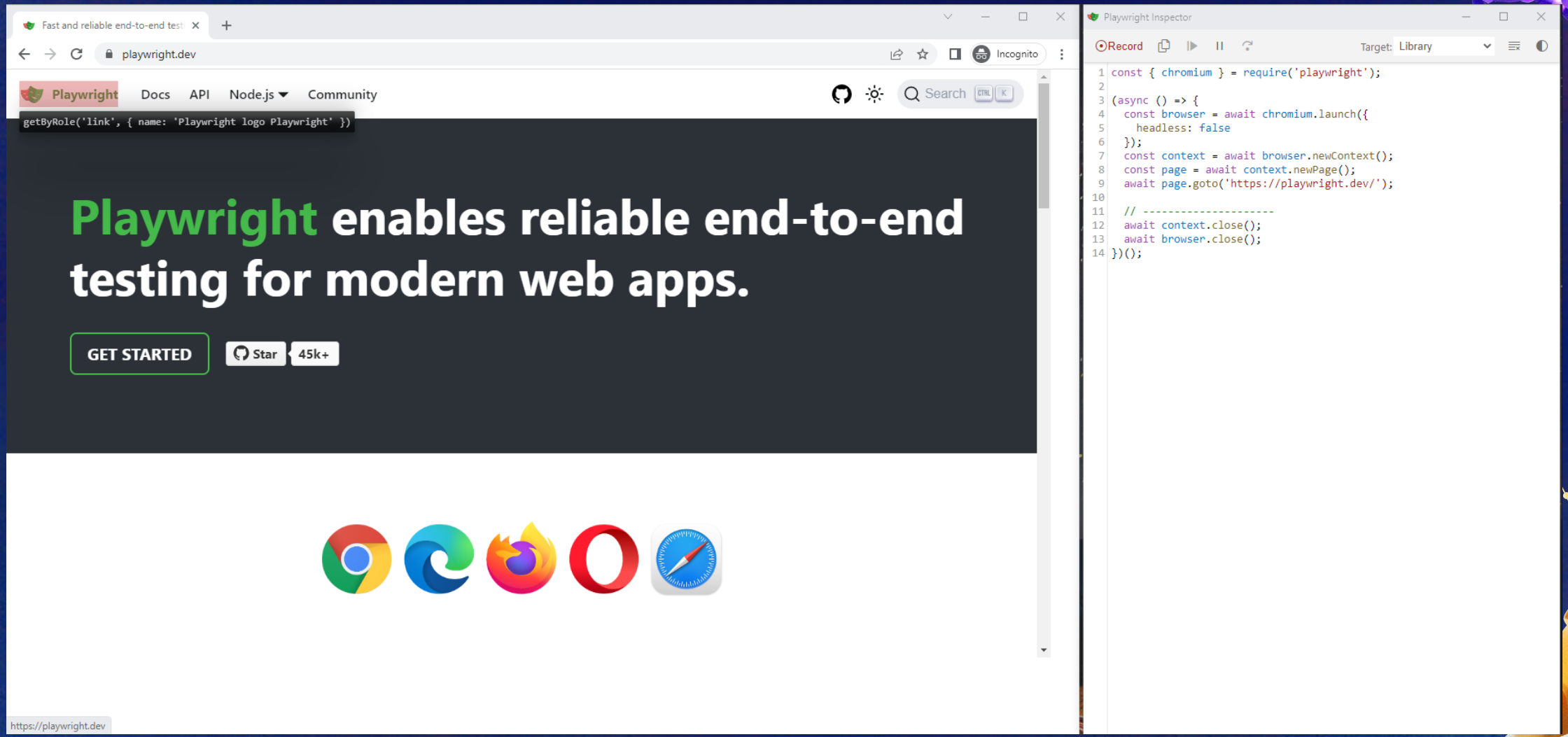
如何使用 – 项目初始化


文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
HelloPlaywright - Visual Studio Code
资源管理器
HELLOPLAYWRIGHT
node_modules
tests
tests-examples
.gitignore
package-lock.json
package.json
playwright.config.js
终端 AZURE 问题 输出 调试控制台
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\Work\Personal\HelloPlaywright> npm init playwright
Getting started with writing end-to-end tests with Playwright:
Initializing project in '.'
√ Do you want to use TypeScript or JavaScript? · JavaScript
√ Where to put your end-to-end tests? · tests
√ Add a GitHub Actions workflow? (y/N) · false
√ Install Playwright browsers (can be done manually via 'npx playwright install')? (Y/n) · true
Initializing NPM project (npm init -y)…
Wrote to D:\Work\Personal\HelloPlaywright\package.json:
{
"name": "helloplaywright",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}
Installing Playwright Test (npm install --save-dev @playwright/test)…
added 3 packages, and audited 4 packages in 4s
found 0 vulnerabilities
Downloading browsers (npx playwright install)…
Writing playwright.config.js.
Writing tests\example.spec.js.
Writing tests-examples\demo-todo-app.spec.js.
Writing package.json.
√ Success! Created a Playwright Test project at D:\Work\Personal\HelloPlaywright
Inside that directory, you can run several commands:
大纲
时间线
Prettier

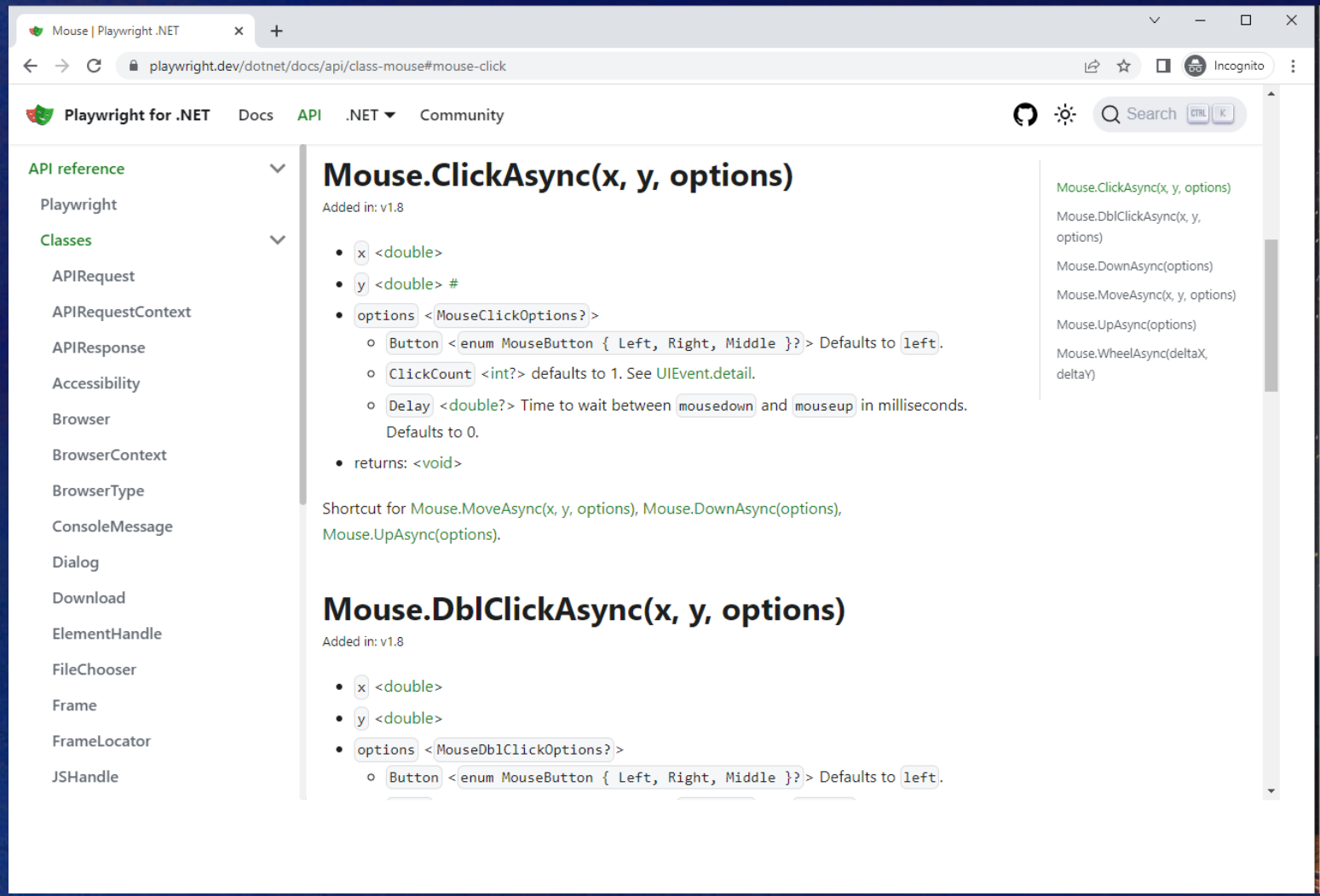
如何使用 – 录制脚本

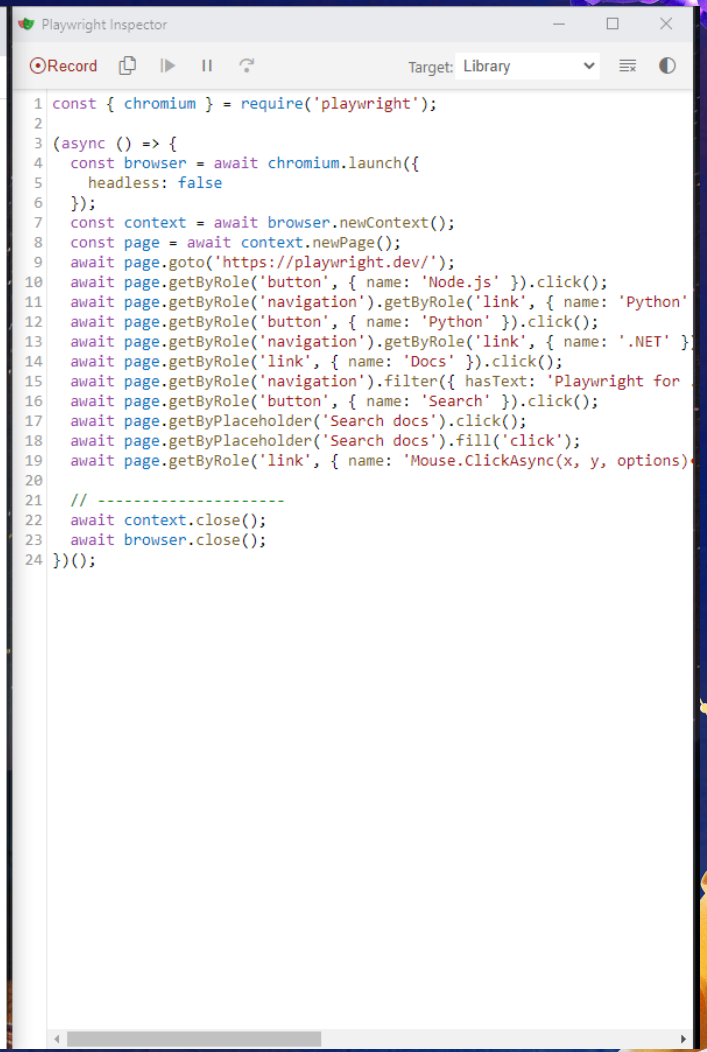

文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
HelloPlaywright - Visual Studio Code
资源管理器
HELLOPLAYWRIGHT
node_modules
tests
tests-examples
.gitignore
package-lock.json
package.json
playwright.config.js
终端 AZURE 问题 输出 调试控制台
powershell
Installing Playwright Test (npm install --save-dev @playwright/test)…
added 3 packages, and audited 4 packages in 4s
found 0 vulnerabilities
Downloading browsers (npx playwright install)…
Writing playwright.config.js.
Writing tests\example.spec.js.
Writing tests-examples\demo-todo-app.spec.js.
Writing package.json.
✓ Success! Created a Playwright Test project at D:\Work\Personal\HelloPlaywright
Inside that directory, you can run several commands:
npx playwright test
Runs the end-to-end tests.
npx playwright test --project=chromium
Runs the tests only on Desktop Chrome.
npx playwright test example
Runs the tests in a specific file.
npx playwright test --debug
Runs the tests in debug mode.
npx playwright codegen
Auto generate tests with Codegen.
We suggest that you begin by typing:
npx playwright test
And check out the following files:
- .\tests\example.spec.js - Example end-to-end test
- .\tests-examples\demo-todo-app.spec.js - Demo Todo App end-to-end tests
- .\playwright.config.js - Playwright Test configuration
Visit https://playwright.dev/docs/intro for more information. ✨
Happy hacking!
PS D:\Work\Personal\HelloPlaywright> npx playwright codegen playwright.dev
大纲
时间线
Prettier

如何使用 – 录制脚本

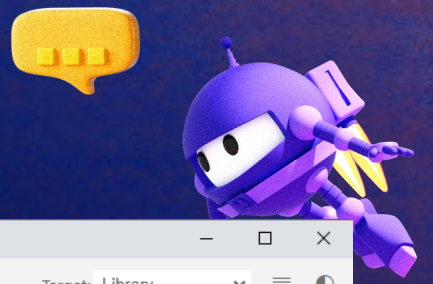

Fast and reliable end-to-end test
playwright.dev
Incognito
Playwright
Docs
API
Node.js
Community
Search
getByRole('link', { name: 'Playwright logo Playwright' })
Playwright enables reliable end-to-end testing for modern web apps.
GET STARTED
Star
45k+
https://playwright.dev
Playwright Inspector
Record
Target: Library
1 const { chromium } = require('playwright');
2
3 (async () => {
4 const browser = await chromium.launch({
5 headless: false
6 });
7 const context = await browser.newContext();
8 const page = await context.newPage();
9 await page.goto('https://playwright.dev/');
10
11 // ---------------------
12 await context.close();
13 await browser.close();
14 })();

如何使用 – 录制脚本


Mouse | Playwright .NET
playwright.dev/dotnet/docs/api/class-mouse#mouse-click
Incognito
Playwright for .NET
Docs
API
.NET
Community
Search
API reference
Playwright
Classes
APIRequest
APIRequestContext
APIResponse
Accessibility
Browser
BrowserContext
BrowserType
ConsoleMessage
Dialog
Download
ElementHandle
FileChooser
Frame
FrameLocator
JSHandle
Mouse.ClickAsync(x, y, options)
Added in: v1.8
x <double>
y <double> #
options <MouseClickOptions?>
Button <enum MouseButton { Left, Right, Middle }?> Defaults to left.
ClickCount <int?> defaults to 1. See UIEvent.detail.
Delay <double?> Time to wait between mousedown and mouseup in milliseconds. Defaults to 0.
returns: <void>
Shortcut for Mouse.MoveAsync(x, y, options), Mouse.DownAsync(options), Mouse.UpAsync(options).
Mouse.DblClickAsync(x, y, options)
Added in: v1.8
x <double>
y <double>
options <MouseDblClickOptions?>
Button <enum MouseButton { Left, Right, Middle }?> Defaults to left.
Mouse.ClickAsync(x, y, options)
Mouse.DblClickAsync(x, y, options)
Mouse.DownAsync(options)
Mouse.MoveAsync(x, y, options)
Mouse.UpAsync(options)
Mouse.WheelAsync(deltaX, deltaY)


Playwright Inspector
Record
Target: Library
const { chromium } = require('playwright');
(async () => {
const browser = await chromium.launch({
headless: false
});
const context = await browser.newContext();
const page = await context.newPage();
await page.goto('https://playwright.dev/');
await page.getByRole('button', { name: 'Node.js' }).click();
await page.getByRole('navigation').getByRole('link', { name: 'Python'
await page.getByRole('button', { name: 'Python' }).click();
await page.getByRole('navigation').getByRole('link', { name: '.NET' }
await page.getByRole('link', { name: 'Docs' }).click();
await page.getByRole('navigation').filter({ hasText: 'Playwright for
await page.getByRole('button', { name: 'Search' }).click();
await page.getByPlaceholder('Search docs').click();
await page.getByPlaceholder('Search docs').fill('click');
await page.getByRole('link', { name: 'Mouse.ClickAsync(x, y, options)
// ---------------------
await context.close();
await browser.close();
})();

如何使用 – 切换语言 (C#)



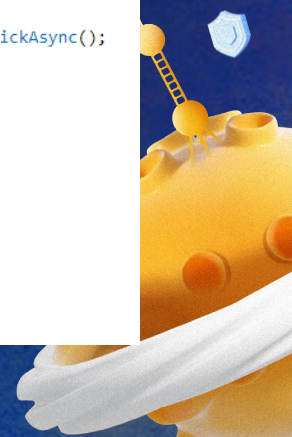
Playwright Inspector

Record

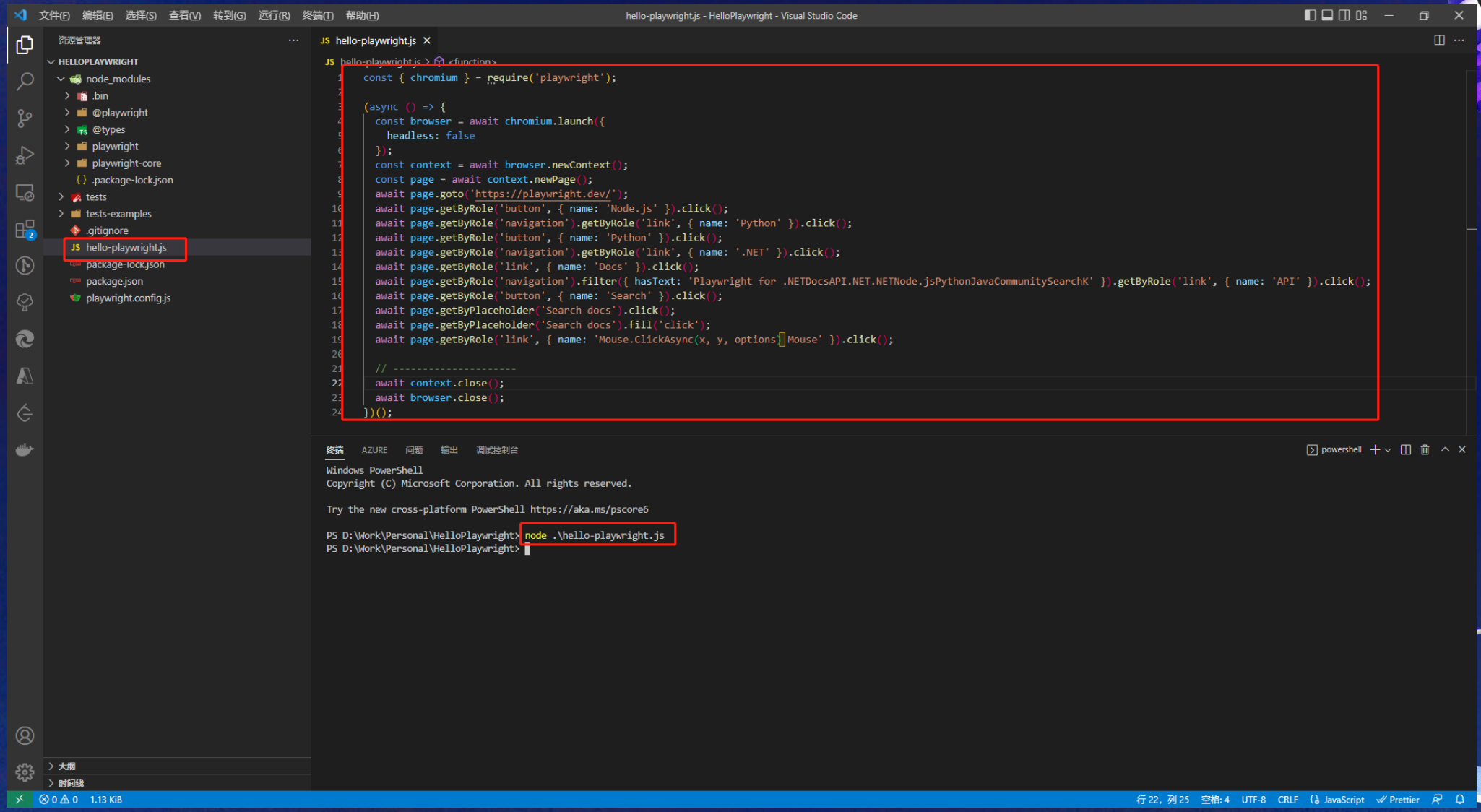
Target: Library

- Node.js
  - Test Runner
  - Library
- Java
  - Library
- Python
  - Pytest
  - Library
  - Library Async
- .NET C#
  - MSTest
  - NUnit
  - Library

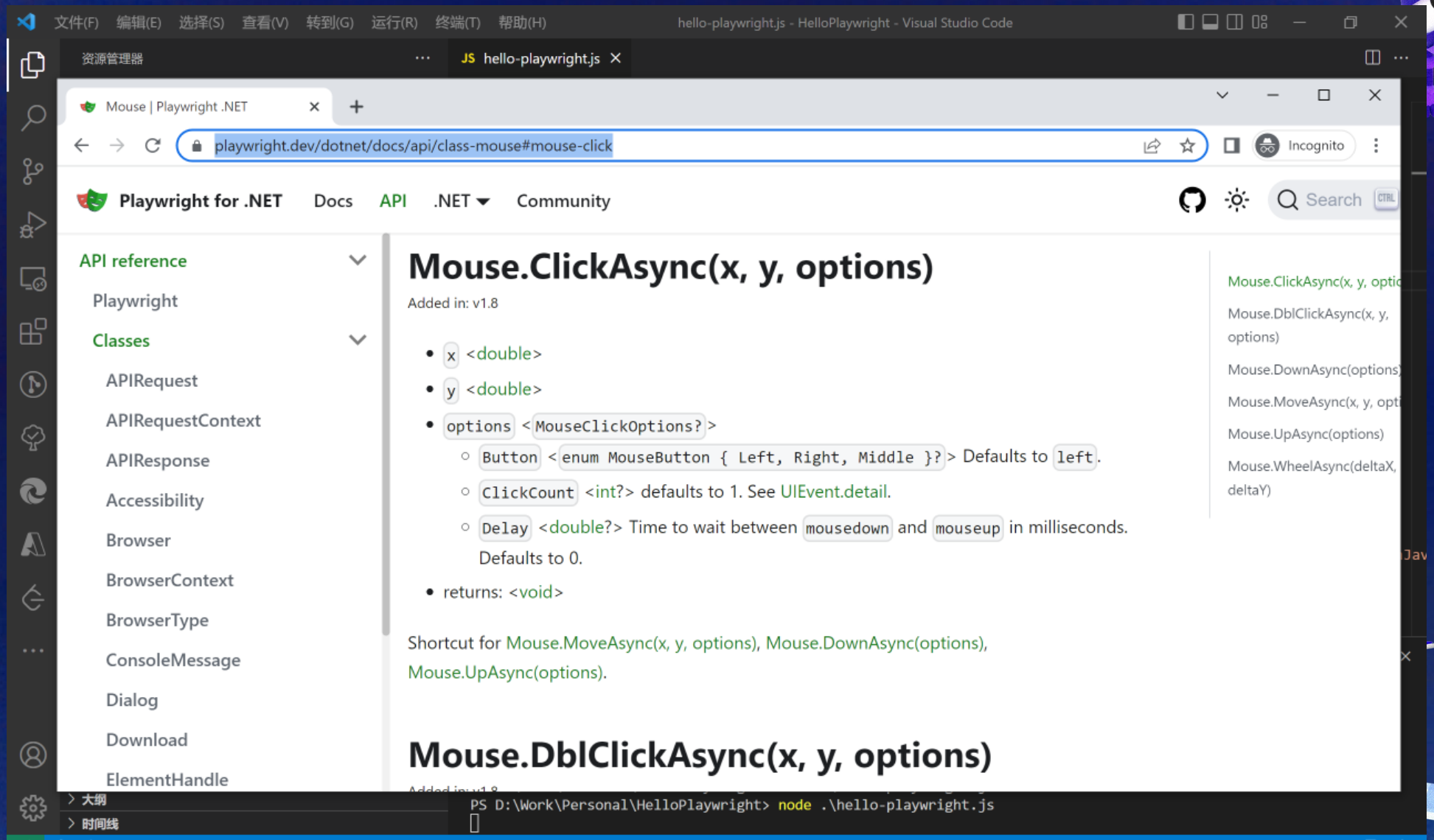
```
using Microsoft.Playwright;
using System;
using System.Threading.Tasks;

class Program
{
    public static async Task Main()
    {
        using var playwright = await Playwright.CreateAsync();
        await using var browser = await playwright.Chromium.LaunchAsync(new BrowserTypeLaunchOptions
        {
            Headless = false,
        });
        var context = await browser.NewContextAsync();

        var page = await context.NewPageAsync();

        await page.GotoAsync("https://playwright.dev/");

        await page.GetByRole(AriaRole.Button, new() { NameString = "Node.js" }).ClickAsync();

        await page.GetByRole(AriaRole.Navigation).GetByRole(AriaRole.Link, new() { NameString = "Python" }).ClickAsync();

        await page.GetByRole(AriaRole.Button, new() { NameString = "Python" }).ClickAsync();

        await page.GetByRole(AriaRole.Navigation).GetByRole(AriaRole.Link, new() { NameString = ".NET" }).ClickAsync();

        await page.GetByRole(AriaRole.Link, new() { NameString = "Docs" }).ClickAsync();

        await page.GetByRole(AriaRole.Navigation).Filter(new() { HasTextString = "Playwright for .NETDocsAPI.NET.NETNode.jsPythonJavaCommunitySearchK" }).GetByRole(AriaRole.Link, new() { NameString = "API" }).ClickAsync();

        await page.GetByRole(AriaRole.Button, new() { NameString = "Search" }).ClickAsync();

        await page.GetByPlaceholder("Search docs").ClickAsync();

        await page.GetByPlaceholder("Search docs").FillAsync("click");

        await page.GetByRole(AriaRole.Link, new() { NameString = "Mouse.ClickAsync(x, y, options)• Mouse" }).ClickAsync();

    }
}
```



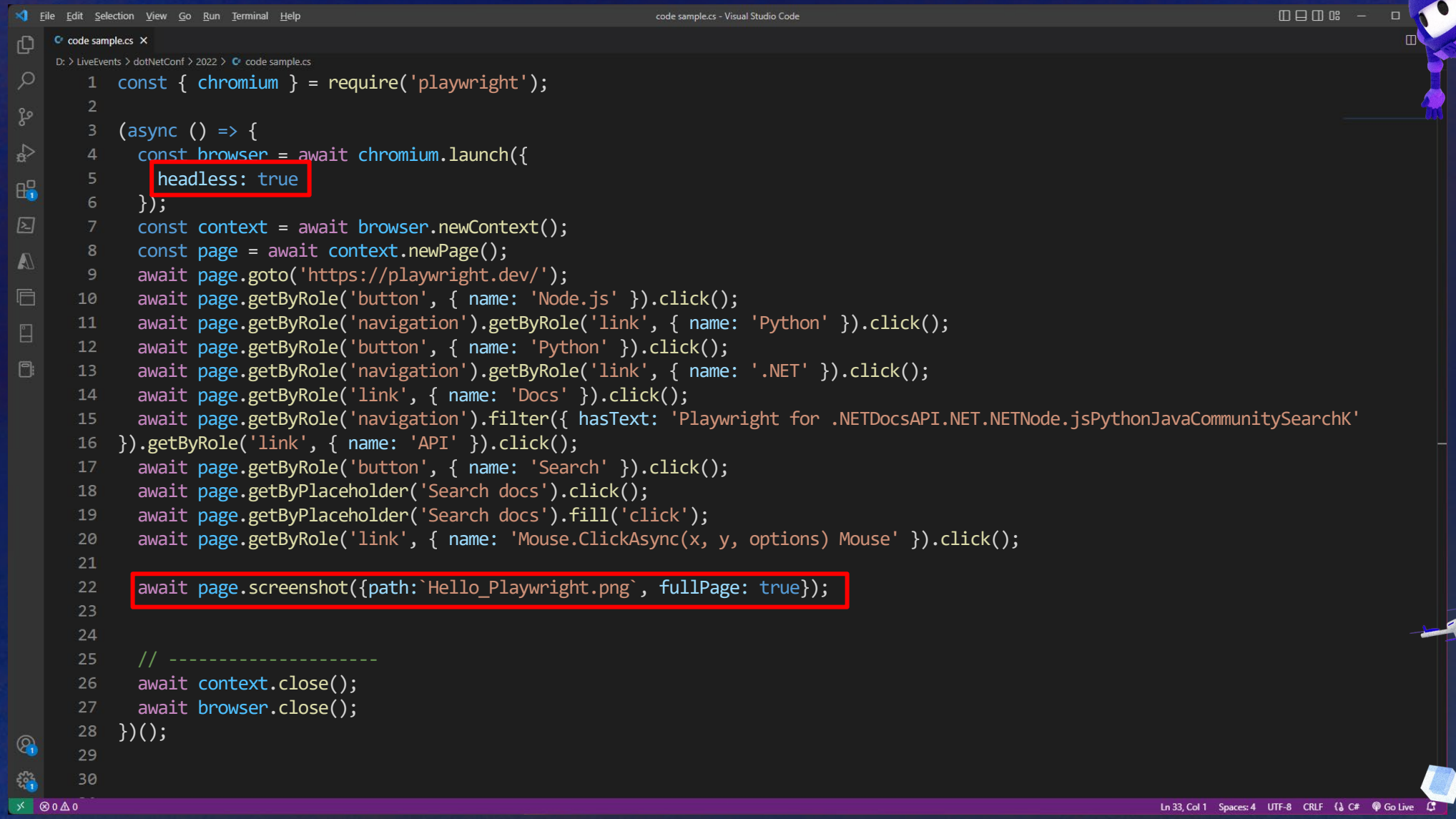
如何使用 – 运行


文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
hello-playwright.js - HelloPlaywright - Visual Studio Code
资源管理器
HELLOPLAYWRIGHT
node_modules
.bin
@playwright
@types
playwright
playwright-core
.package-lock.json
tests
tests-examples
.gitignore
hello-playwright.js
package-lock.json
package.json
playwright.config.js
hello-playwright.js
const { chromium } = require('playwright');
(async () => {
const browser = await chromium.launch({
headless: false
});
const context = await browser.newContext();
const page = await context.newPage();
await page.goto('https://playwright.dev/');
await page.getByRole('button', { name: 'Node.js' }).click();
await page.getByRole('navigation').getByRole('link', { name: 'Python' }).click();
await page.getByRole('button', { name: 'Python' }).click();
await page.getByRole('navigation').getByRole('link', { name: '.NET' }).click();
await page.getByRole('link', { name: 'Docs' }).click();
await page.getByRole('navigation').filter({ hasText: 'Playwright for .NETDocsAPI.NET.NETNode.jsPythonJavaCommunitySearchK' }).getByRole('link', { name: 'API' }).click();
await page.getByRole('button', { name: 'Search' }).click();
await page.getByPlaceholder('Search docs').click();
await page.getByPlaceholder('Search docs').fill('click');
await page.getByRole('link', { name: 'Mouse.ClickAsync(x, y, options)Mouse' }).click();
// ---------------------
await context.close();
await browser.close();
})();
终端 AZURE 问题 输出 调试控制台
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\Work\Personal\HelloPlaywright> node .\hello-playwright.js
PS D:\Work\Personal\HelloPlaywright>
大纲
时间线
0 0 1.13 KiB
行 22，列 25 空格: 4 UTF-8 CRLF JavaScript Prettier

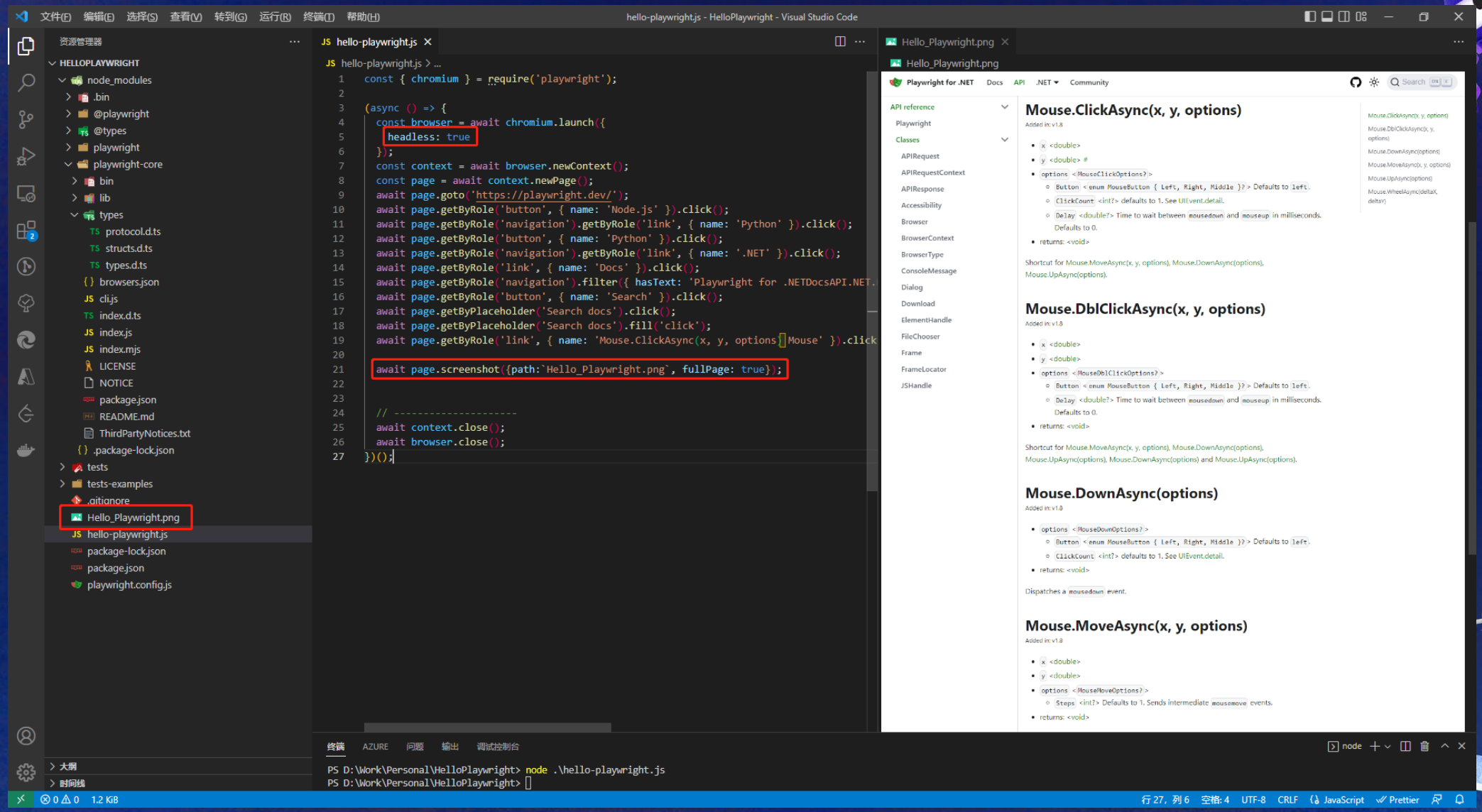
如何使用 – 运行


文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
hello-playwright.js - HelloPlaywright - Visual Studio Code
资源管理器
hello-playwright.js
Mouse | Playwright .NET
playwright.dev/dotnet/docs/api/class-mouse#mouse-click
Incognito
Playwright for .NET
Docs
API
.NET
Community
Search
API reference
Playwright
Classes
APIRequest
APIRequestContext
APIResponse
Accessibility
Browser
BrowserContext
BrowserType
ConsoleMessage
Dialog
Download
ElementHandle
Mouse.ClickAsync(x, y, options)
Added in: v1.8
x <double>
y <double>
options <MouseClickOptions?>
Button <enum MouseButton { Left, Right, Middle }?> Defaults to left.
ClickCount <int?> defaults to 1. See UIEvent.detail.
Delay <double?> Time to wait between mousedown and mouseup in milliseconds. Defaults to 0.
returns: <void>
Shortcut for Mouse.MoveAsync(x, y, options), Mouse.DownAsync(options), Mouse.UpAsync(options).
Mouse.DblClickAsync(x, y, options)
Mouse.ClickAsync(x, y, optio
Mouse.DblClickAsync(x, y, options)
Mouse.DownAsync(options)
Mouse.MoveAsync(x, y, opti
Mouse.UpAsync(options)
Mouse.WheelAsync(deltaX, deltaY)
Mouse.DblClickAsync(x, y, options)
大纲
时间线
PS D:\Work\Personal\HelloPlaywright> node .\hello-playwright.js
行 9，列 23
空格: 2
UTF-8
CRLF
JavaScript
Prettier

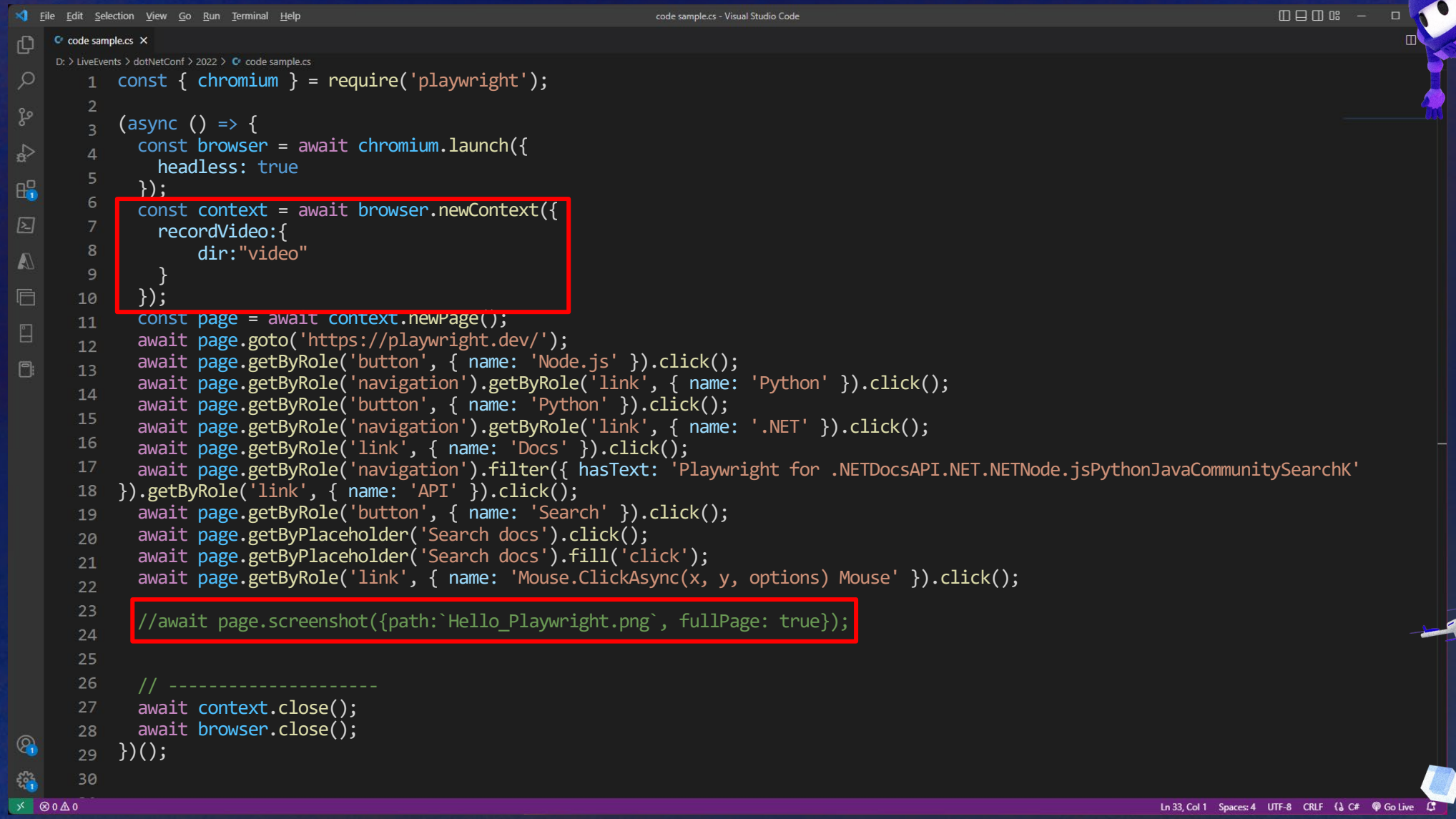
如何使用 – 截图


File Edit Selection View Go Run Terminal Help
code sample.cs - Visual Studio Code
code sample.cs
D: > LiveEvents > dotNetConf > 2022 > code sample.cs
const { chromium } = require('playwright');
(async () => {
const browser = await chromium.launch({
headless: true
});
const context = await browser.newContext();
const page = await context.newPage();
await page.goto('https://playwright.dev/');
await page.getByRole('button', { name: 'Node.js' }).click();
await page.getByRole('navigation').getByRole('link', { name: 'Python' }).click();
await page.getByRole('button', { name: 'Python' }).click();
await page.getByRole('navigation').getByRole('link', { name: '.NET' }).click();
await page.getByRole('link', { name: 'Docs' }).click();
await page.getByRole('navigation').filter({ hasText: 'Playwright for .NETDocsAPI.NET.NETNode.jsPythonJavaCommunitySearchK'
}).getByRole('link', { name: 'API' }).click();
await page.getByRole('button', { name: 'Search' }).click();
await page.getByPlaceholder('Search docs').click();
await page.getByPlaceholder('Search docs').fill('click');
await page.getByRole('link', { name: 'Mouse.ClickAsync(x, y, options) Mouse' }).click();
await page.screenshot({path:`Hello_Playwright.png`, fullPage: true});
// ---------------------
await context.close();
await browser.close();
})();
Ln 33, Col 1 Spaces: 4 UTF-8 CRLF C# Go Live

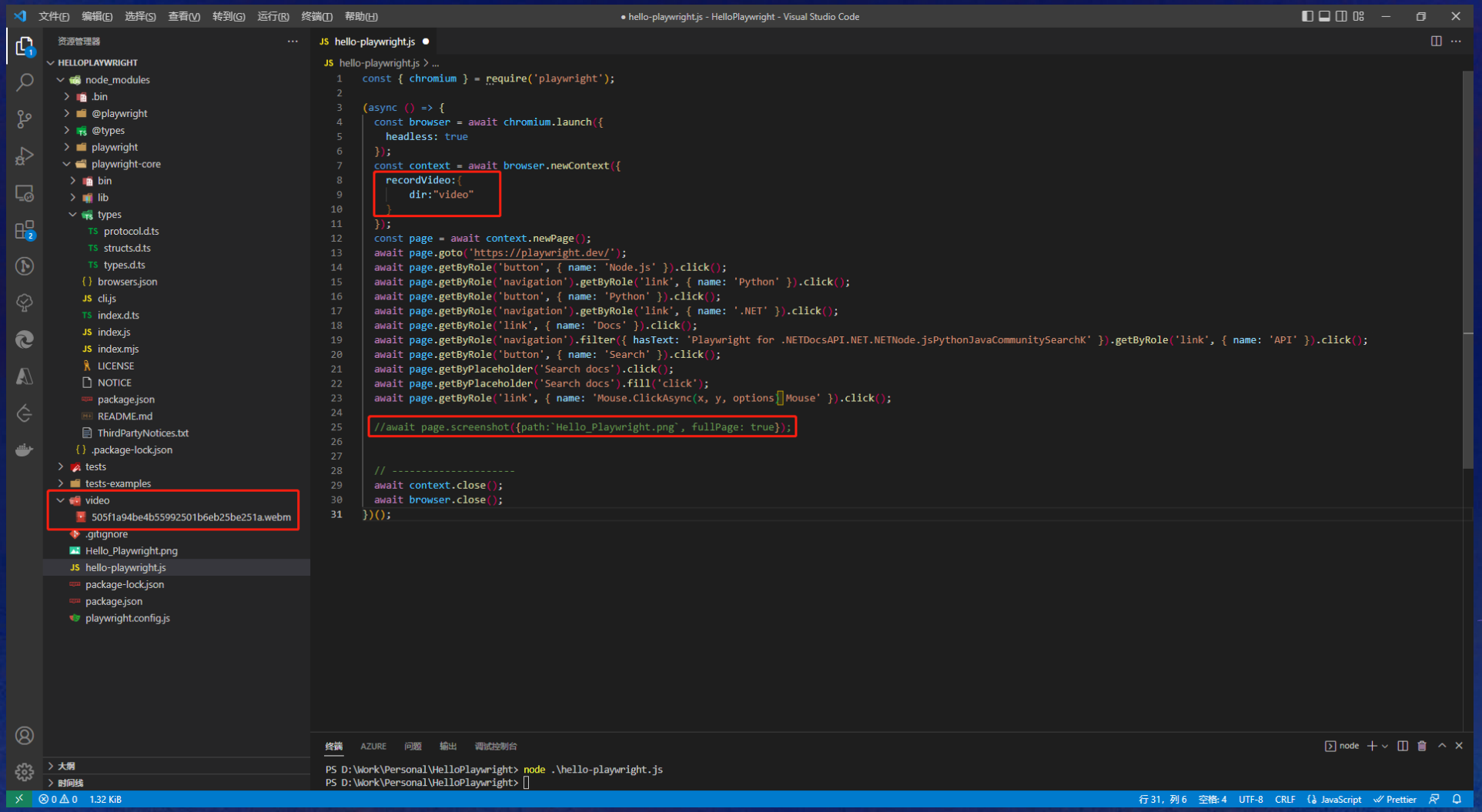
如何使用 – 截图结果 (NodeJS)



如何使用 – 视频录制

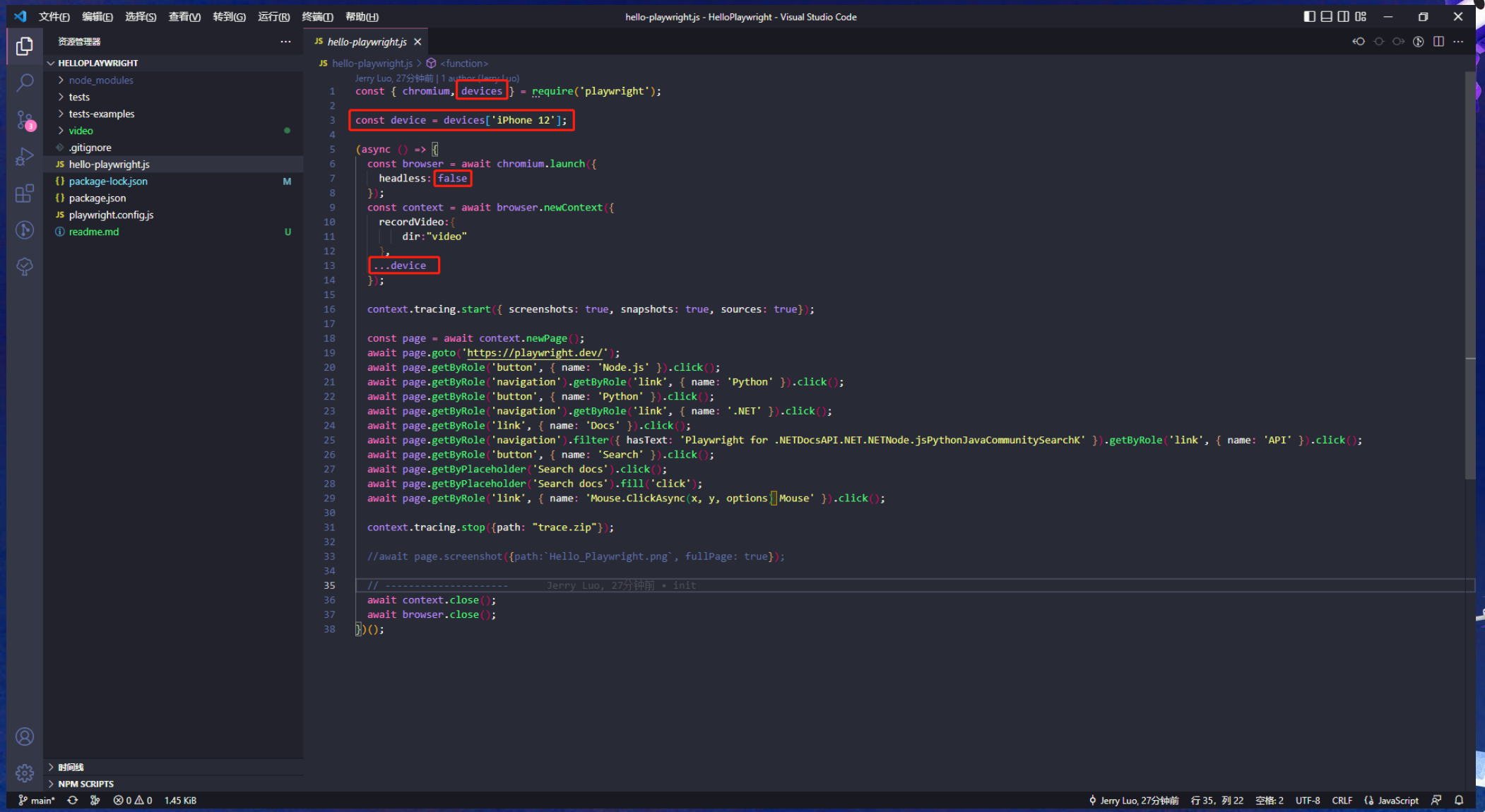

File Edit Selection View Go Run Terminal Help
code sample.cs - Visual Studio Code
code sample.cs
D: > LiveEvents > dotNetConf > 2022 > code sample.cs
const { chromium } = require('playwright');
(async () => {
const browser = await chromium.launch({
headless: true
});
const context = await browser.newContext({
recordVideo:{
dir:"video"
}
});
const page = await context.newPage();
await page.goto('https://playwright.dev/');
await page.getByRole('button', { name: 'Node.js' }).click();
await page.getByRole('navigation').getByRole('link', { name: 'Python' }).click();
await page.getByRole('button', { name: 'Python' }).click();
await page.getByRole('navigation').getByRole('link', { name: '.NET' }).click();
await page.getByRole('link', { name: 'Docs' }).click();
await page.getByRole('navigation').filter({ hasText: 'Playwright for .NETDocsAPI.NET.NETNode.jsPythonJavaCommunitySearchK'
}).getByRole('link', { name: 'API' }).click();
await page.getByRole('button', { name: 'Search' }).click();
await page.getByPlaceholder('Search docs').click();
await page.getByPlaceholder('Search docs').fill('click');
await page.getByRole('link', { name: 'Mouse.ClickAsync(x, y, options) Mouse' }).click();
//await page.screenshot({path:`Hello_Playwright.png`, fullPage: true});
// ---------------------
await context.close();
await browser.close();
})();
Ln 33, Col 1 Spaces: 4 UTF-8 CRLF C# Go Live

如何使用 – 视频录制运行结果

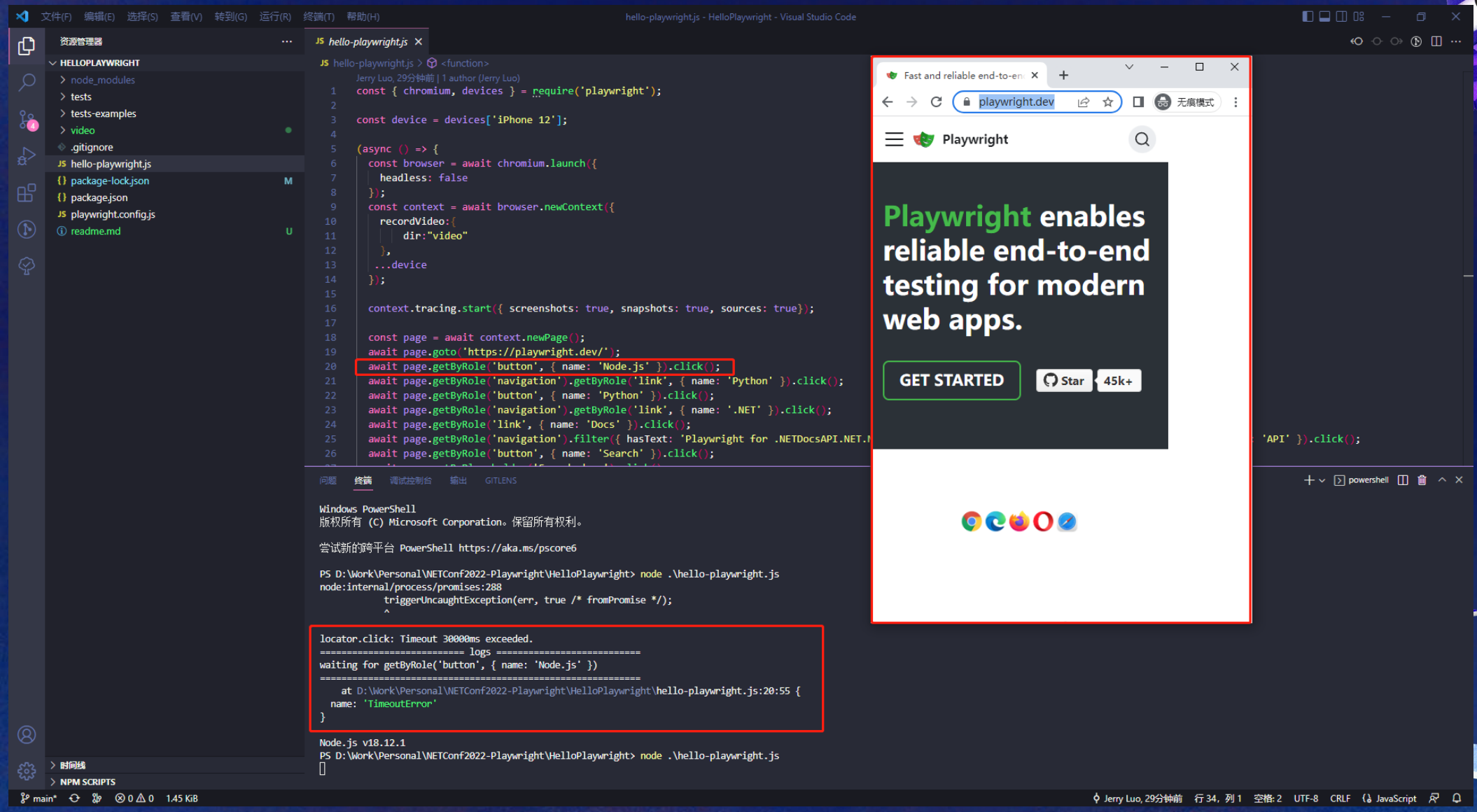


如何使用 – 视频

如何使用 – 设备模拟

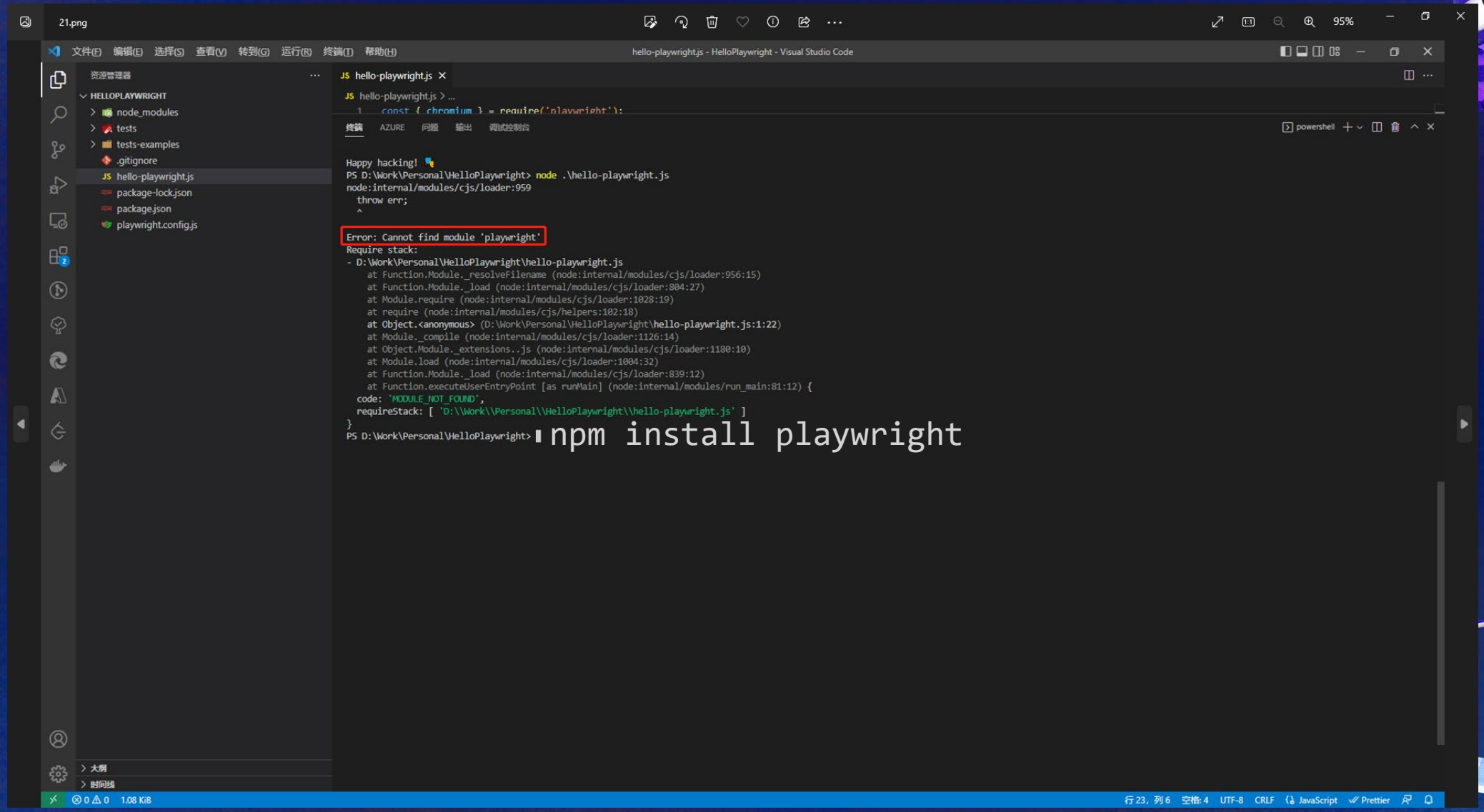

文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
hello-playwright.js - HelloPlaywright - Visual Studio Code
资源管理器
HELLOPLAYWRIGHT
node_modules
tests
tests-examples
video
.gitignore
hello-playwright.js
package-lock.json
package.json
playwright.config.js
readme.md
hello-playwright.js
const { chromium, devices } = require('playwright');
const device = devices['iPhone 12'];
(async () => {
const browser = await chromium.launch({
headless: false
});
const context = await browser.newContext({
recordVideo:{
dir:"video"
},
...device
});
context.tracing.start({ screenshots: true, snapshots: true, sources: true});
const page = await context.newPage();
await page.goto('https://playwright.dev/');
await page.getByRole('button', { name: 'Node.js' }).click();
await page.getByRole('navigation').getByRole('link', { name: 'Python' }).click();
await page.getByRole('button', { name: 'Python' }).click();
await page.getByRole('navigation').getByRole('link', { name: '.NET' }).click();
await page.getByRole('link', { name: 'Docs' }).click();
await page.getByRole('navigation').filter({ hasText: 'Playwright for .NETDocsAPI.NET.NETNode.jsPythonJavaCommunitySearchK' }).getByRole('link', { name: 'API' }).click();
await page.getByRole('button', { name: 'Search' }).click();
await page.getByPlaceholder('Search docs').click();
await page.getByPlaceholder('Search docs').fill('click');
await page.getByRole('link', { name: 'Mouse.ClickAsync(x, y, options)Mouse' }).click();
context.tracing.stop({path: "trace.zip"});
//await page.screenshot({path:`Hello_Playwright.png`, fullPage: true});
// ---------------------
Jerry Luo, 27分钟前 • init
await context.close();
await browser.close();
})();
时间线
NPM SCRIPTS
main*
1.45 KiB
Jerry Luo, 27分钟前
行 35，列 22
空格: 2
UTF-8
CRLF
JavaScript

如何使用 – 设备模拟

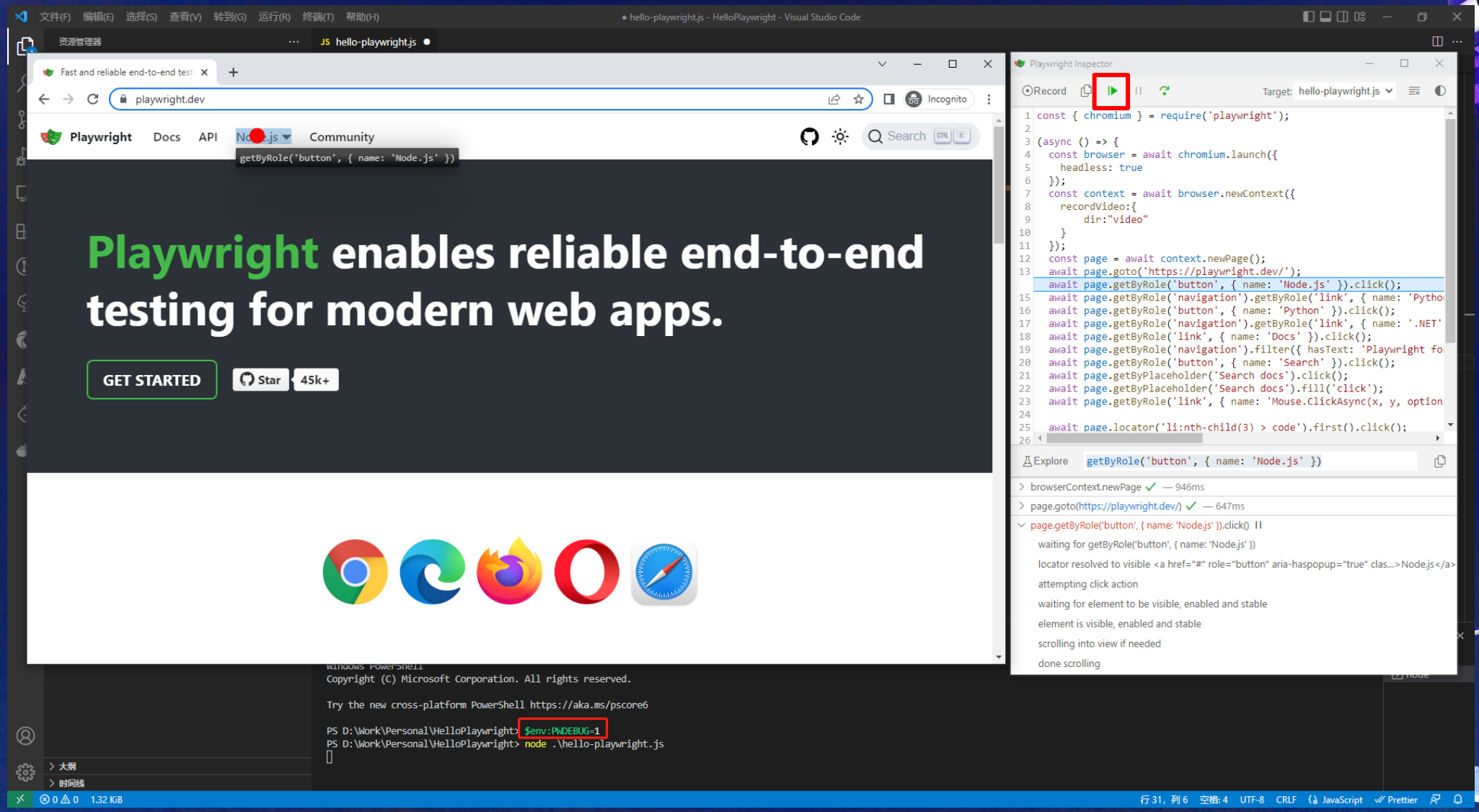

hello-playwright.js - HelloPlaywright - Visual Studio Code
HELLOPLAYWRIGHT
node_modules
tests
tests-examples
video
.gitignore
hello-playwright.js
package-lock.json
package.json
playwright.config.js
readme.md
const { chromium, devices } = require('playwright');
const device = devices['iPhone 12'];
(async () => {
const browser = await chromium.launch({
headless: false
});
const context = await browser.newContext({
recordVideo:{
dir:"video"
},
...device
});
context.tracing.start({ screenshots: true, snapshots: true, sources: true});
const page = await context.newPage();
await page.goto('https://playwright.dev/');
await page.getByRole('button', { name: 'Node.js' }).click();
await page.getByRole('navigation').getByRole('link', { name: 'Python' }).click();
await page.getByRole('button', { name: 'Python' }).click();
await page.getByRole('navigation').getByRole('link', { name: '.NET' }).click();
await page.getByRole('link', { name: 'Docs' }).click();
await page.getByRole('navigation').filter({ hasText: 'Playwright for .NETDocsAPI.NET.
await page.getByRole('button', { name: 'Search' }).click();
'API' }).click();
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
尝试新的跨平台 PowerShell https://aka.ms/pscore6
PS D:\Work\Personal\NETConf2022-Playwright\HelloPlaywright> node .\hello-playwright.js
node:internal/process/promises:288
triggerUncaughtException(err, true /* fromPromise */);
locator.click: Timeout 30000ms exceeded.
========== logs ==========
waiting for getByRole('button', { name: 'Node.js' })
==========
at D:\Work\Personal\NETConf2022-Playwright\HelloPlaywright\hello-playwright.js:20:55 {
name: 'TimeoutError'
}
Node.js v18.12.1
PS D:\Work\Personal\NETConf2022-Playwright\HelloPlaywright> node .\hello-playwright.js
Fast and reliable end-to-en
playwright.dev
Playwright
Playwright enables reliable end-to-end testing for modern web apps.
GET STARTED
Star
45k+

常见问题

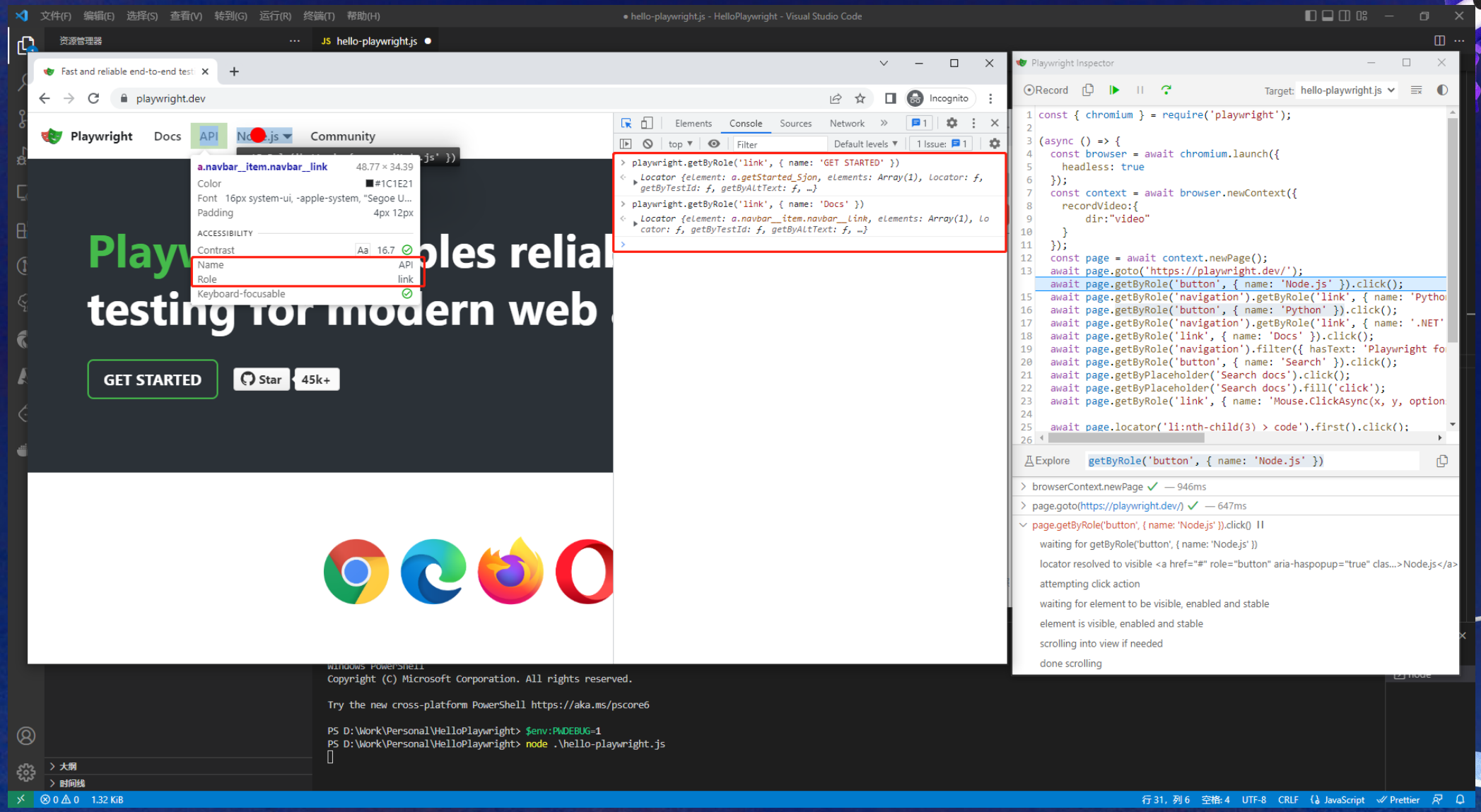
常见问题 – Cannot find module ‘playwright’



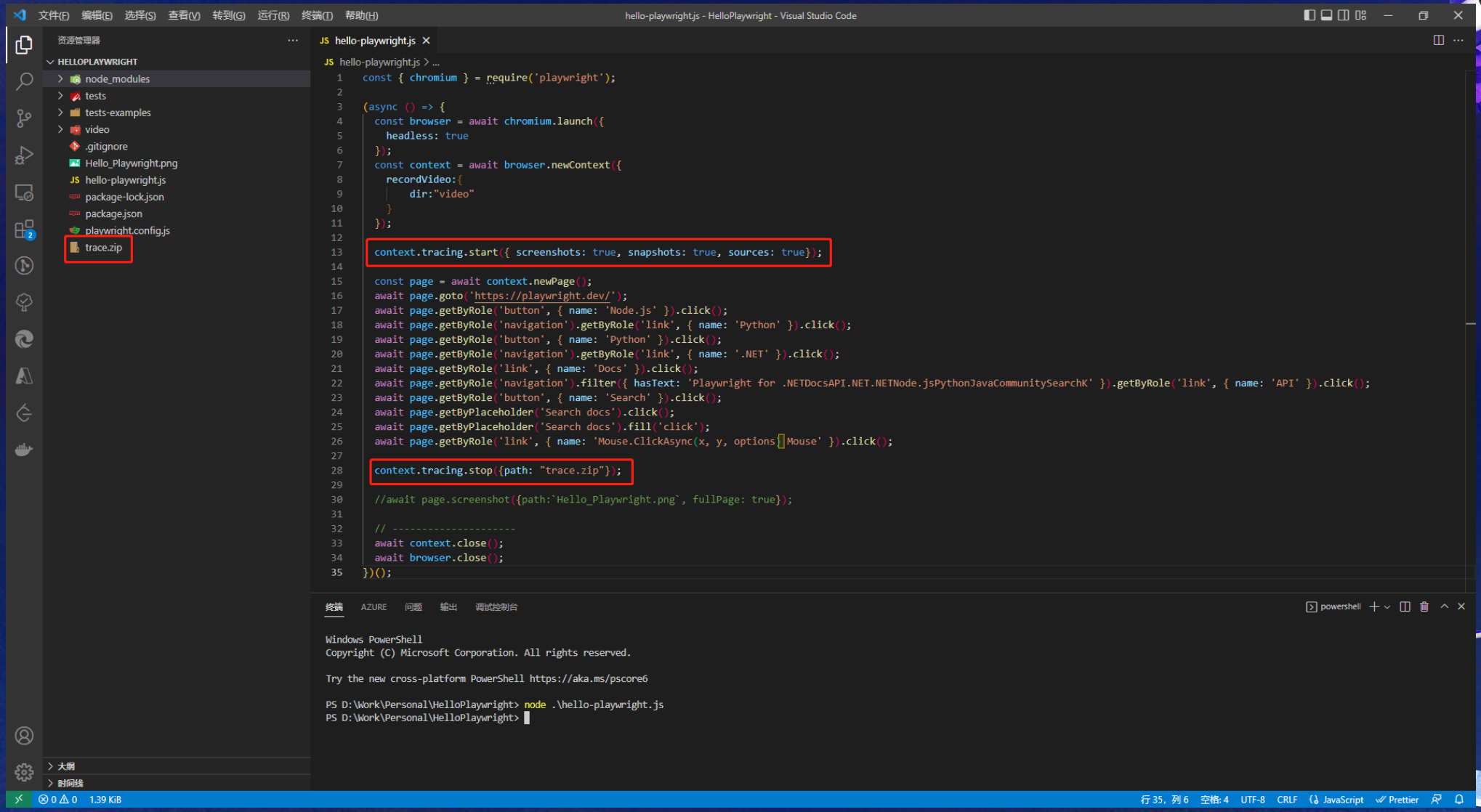
常见问题 – 如何使用Playwright Inspector



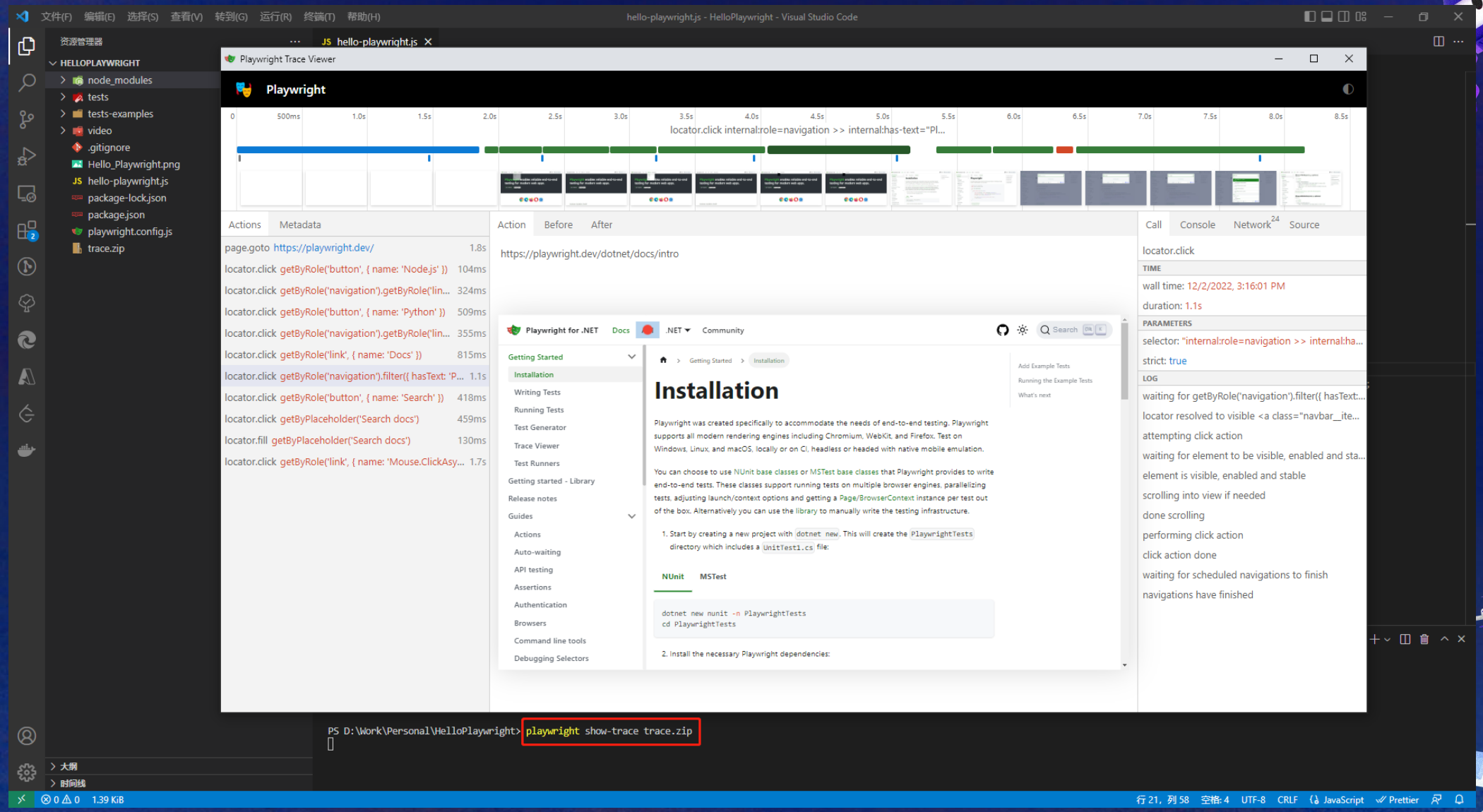
常见问题 – 如何使用Playwright Inspector



常见问题 – 如何使用Trace Viewer

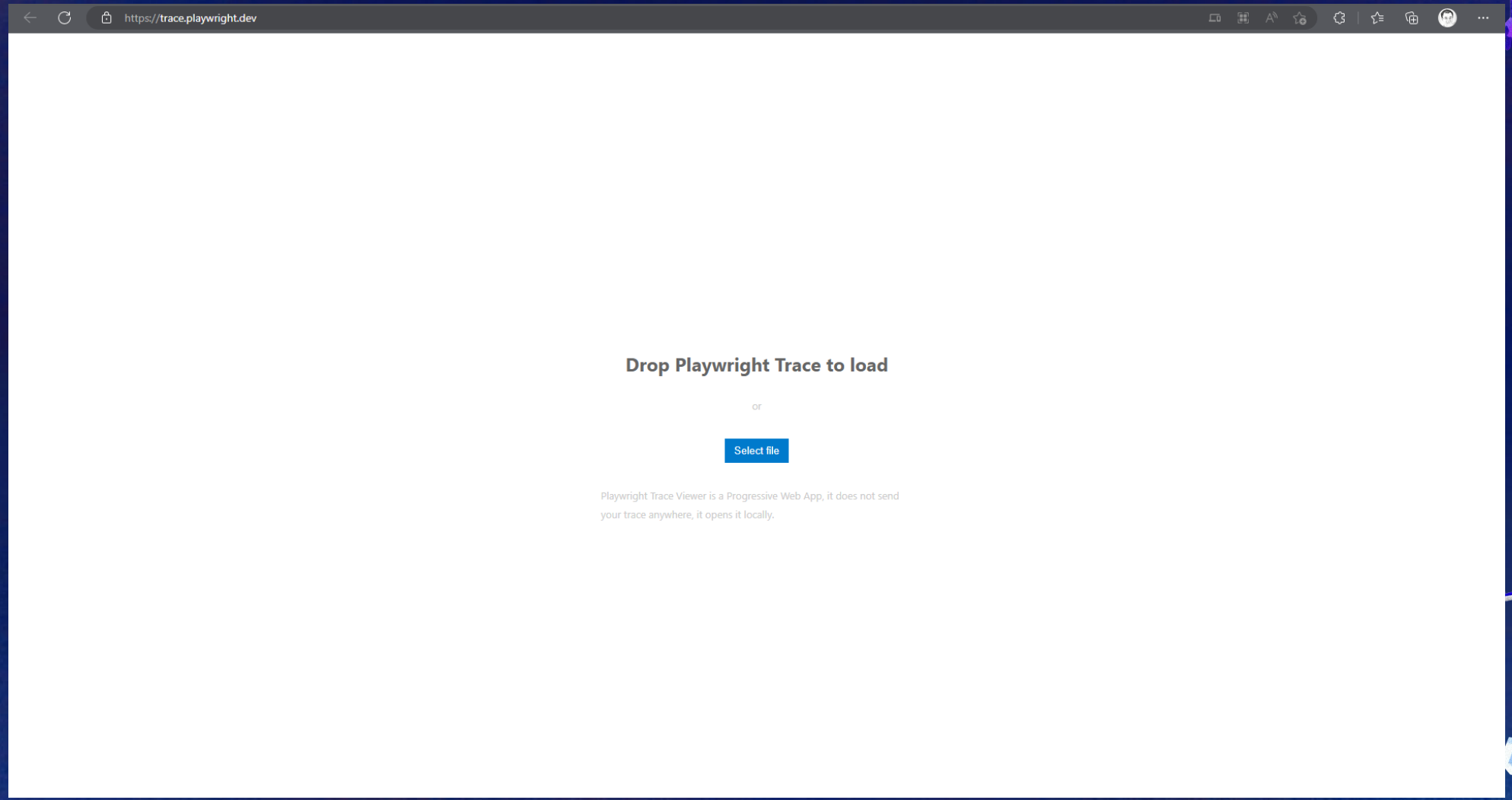


常见问题 – 如何使用Trace Viewer

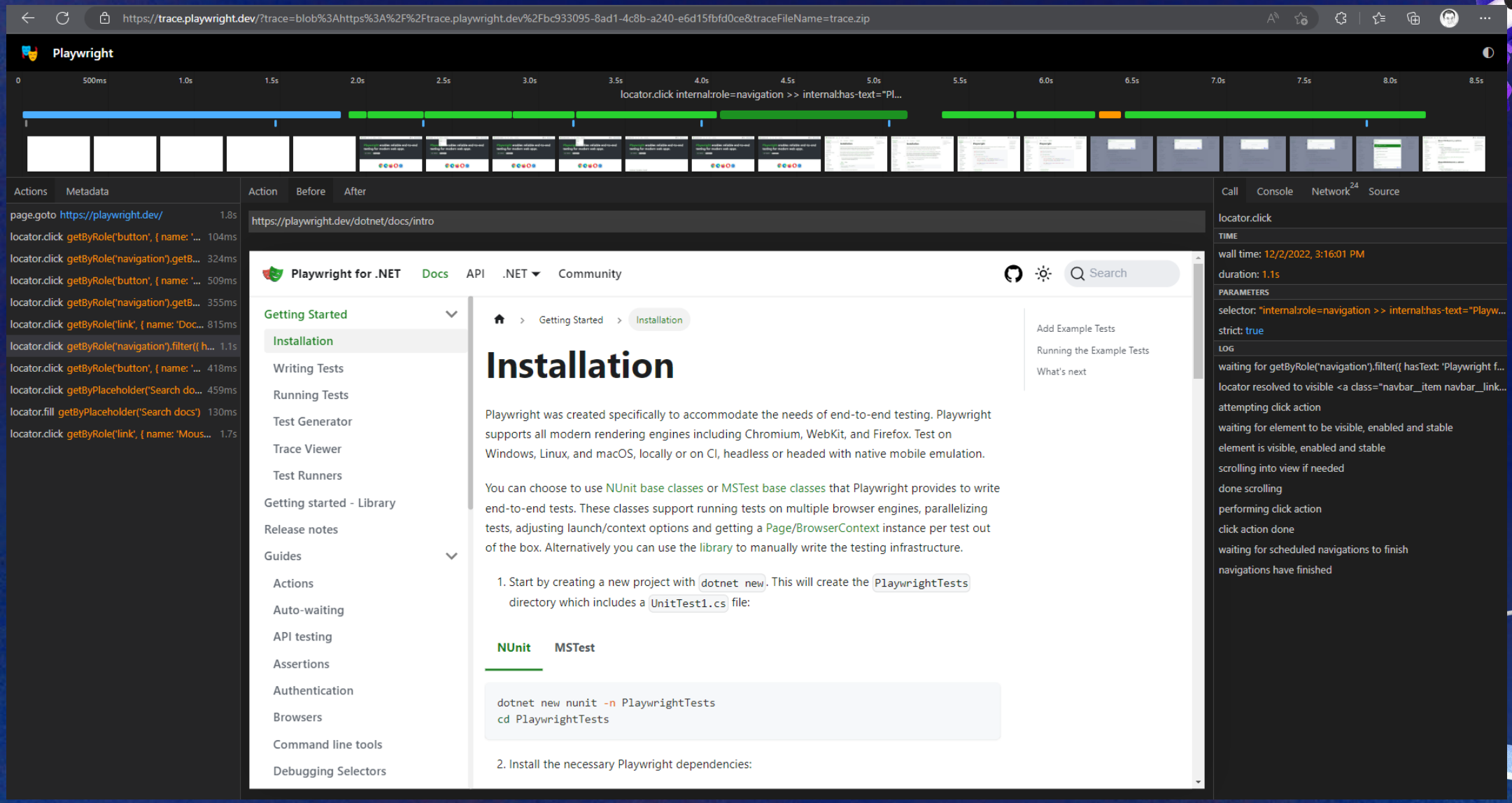


常见问题 – 如何使用Trace Viewer

https://trace.playwright.dev



常见问题 – 如何使用Trace Viewer

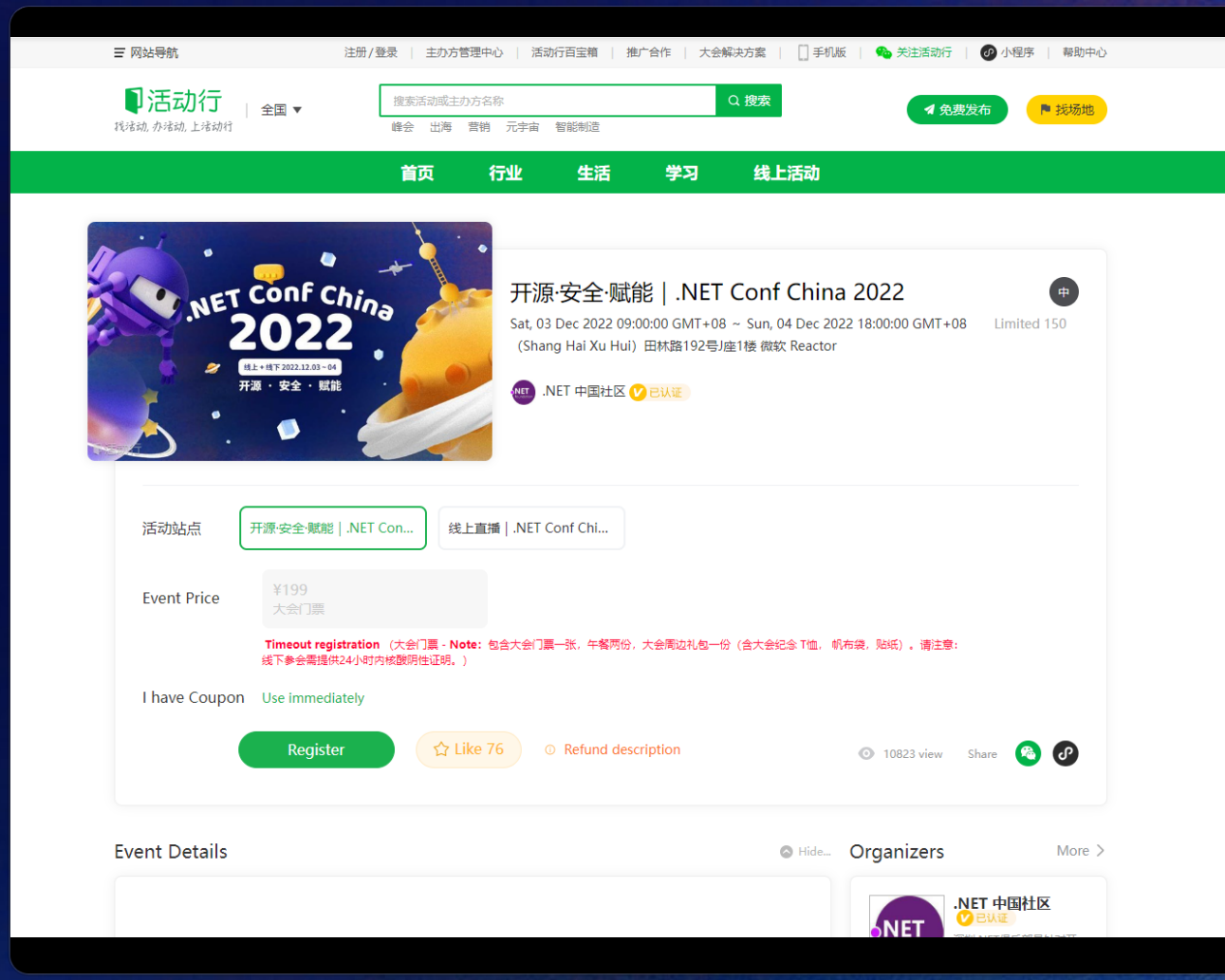


Talk is cheap, show me the code

Demo – 单元测试 (C#)

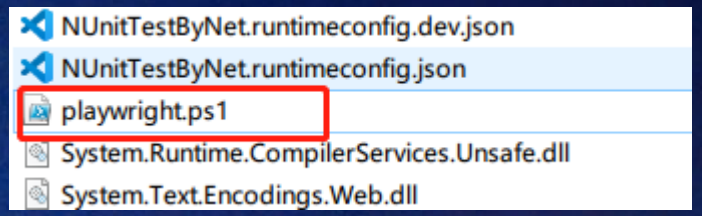
NET Conf China 2022 活动页微信扫码登录

1. 点击登录按钮
2. 选择微信扫码登录
3. 二维码截图
4. 本地打开截图
5. 微信扫码
6. 跳回当前页, 根据用户信息验证是否成功



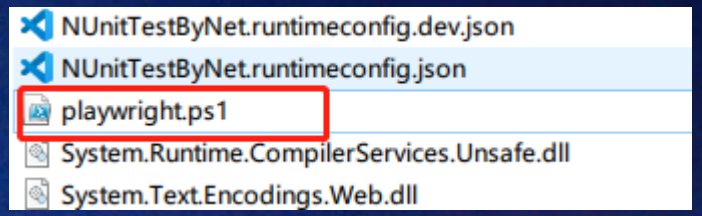
Demo - 环境准备 (C#)

- 创建单元测试项目
  - NUnit
  - MSTest
- 安装Playwright相关依赖
  - Microsoft.Playwright.NUnit
  - Microsoft.Playwright.MSTest
- (Optional) 安装浏览器

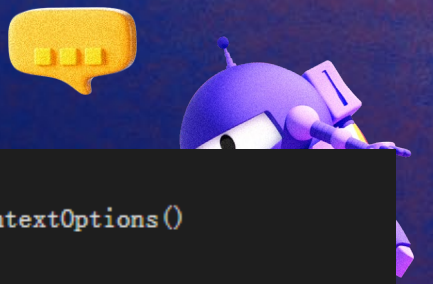


Demo - 环境准备 (C#)

- 创建单元测试项目
  - NUnit
  - MSTest
- 安装Playwright相关依赖
  - Microsoft.Playwright.NUnit
  - Microsoft.Playwright.MSTest
- (Optional) 安装浏览器



Demo – 代码



```
0 个引用|0 项更改|0 名作者，0 项更改
public class Tests
{
    [Parallelizable(ParallelScope.Self)]
    [TestFixture]
    0 个引用|0 项更改|0 名作者，0 项更改
    public class Tests01 : PageTest
    {
        [Test]
        |0 个引用|0 项更改|0 名作者，0 项更改
        public async Task TestLoginInNETConfChina2022Page()
        {
            await Page.GotoAsync("https://www.huodongxing.com/event/4674917451700");
            //点击登录按钮
            await Page.GetByRole(AriaRole.Link, new() { NameString = "登录" })
                .Filter(new() { HasTextString = "登录" }).ClickAsync();

            //点击微信登录按钮，并且等待，直到二维码获取完成
            await Page.RunAndWaitForRequestAsync(async () =>
            {
                await Page.GetByRole(AriaRole.Link, new() { NameString = "微信" }).ClickAsync();
            }, new Regex(".*qrcode.*"));

            //页面截图
            string screenShotPath = $"LoginQRCode{DateTime.Now.Ticks}.png";
            await Page.ScreenshotAsync(new PageScreenshotOptions() { Path = screenShotPath, FullPage = true });

            //打开截图等待扫码
            string qrCodeFilePath = $"{ Directory.GetCurrentDirectory()}\\{screenShotPath}";
            OpenQRCodeAsync(qrCodeFilePath);

            //等待扫码成功后,回到NET Conf 2022 活动页面
            await Page.WaitForURLAsync("https://www.huodongxing.com/event/4674917451700",
                new PageWaitForURLOptions() { Timeout = 0 });

            //通过用户信息来诊断是否登录成功
            var currentUserInfo = Page.GetByRole(AriaRole.Link, new() { NameString = "Hi 阿白" });
            await Expect(currentUserInfo).ToBeVisibleAsync();
        }
```

```
0 个引用|0 项更改|0 名作者，0 项更改
public override BrowserNewContextOptions ContextOptions()
{
    //设置视频保存目录
    return new BrowserNewContextOptions() { RecordVideoDir = "video" };
}
```

```
private void OpenQRCodeAsync(string qrCodeFilePath)
{
    //建立新的系统进程
    System.Diagnostics.Process process = new System.Diagnostics.Process();
    //设置文件名，此处为图片的真实路径+文件名
    process.StartInfo.FileName = qrCodeFilePath;
    //此为关键部分。设置进程运行参数，此时为最大化窗口显示图片。
    process.StartInfo.Arguments = "rundll32.exe %systemroot%\\system32\\shimgvw.dll,ImageView_Fullscreen";
    //此项为是否使用Shell执行程序，因系统默认为true，此项也可不设，但若设置必须为true
    process.StartInfo.UseShellExecute = true;
    //此处可以更改进程所打开窗体的显示样式，可以不设
    process.StartInfo.WindowStyle = System.Diagnostics.ProcessWindowStyle.Hidden;
    process.Start();
    process.Close();
}
```

Demo – 运行

```
//打开截图等待扫码
string qrCodePilePath = $"{ Directory.GetCurrentDirectory()}\\{screenShotPath}";
OpenQRCodeAsync(qrCodePilePath);

//等待扫码成功后,回到NET Conf 2022 活动页面
await Page.WaitForURLAsync("https://www.huodongxing.com/event/4674917451700",
    new PageWaitForURLOptions() { Timeout = 0 });
```

LoginQRCode63805685239... 66%

使用微信扫一扫登录

「活动行」

测试资源管理器

搜索测试资源管理器 (Ctrl+E)

测试	持续时间	特征
NUnitTestByNet (1)	18.9 秒	
NUnitTestByNet (1)	18.9 秒	
Tests+Tests01 (1)	18.9 秒	
TestLoginInNETConfChina20...	18.9 秒	

测试详细信息摘要

TestLoginInNETConfChina2022Page

源: UnitTest1.cs 行 18

持续时间: 18.9 秒

Demo – 视频

FireUG

B站 抖音 公众号 微信群

Thank you!

Let's build amazing apps with .NET 7

get.dot.net/7