AZURE PLAYWRIGHT

La pièce maîtresse du spectacle automatisé !

AZUGFR
Azure User Group France

MTG Paris
Azure

#GLOBALAZURE
BY COMMUNITY · FOR COMMUNITY
2024

Christophe Peugnet - Adrien Clerbois

MICROSOFT MVP, TECHNICAL ARCHITECT @ SENSE OF TECH
Follow us on social media @tossnet1 @aclerbois

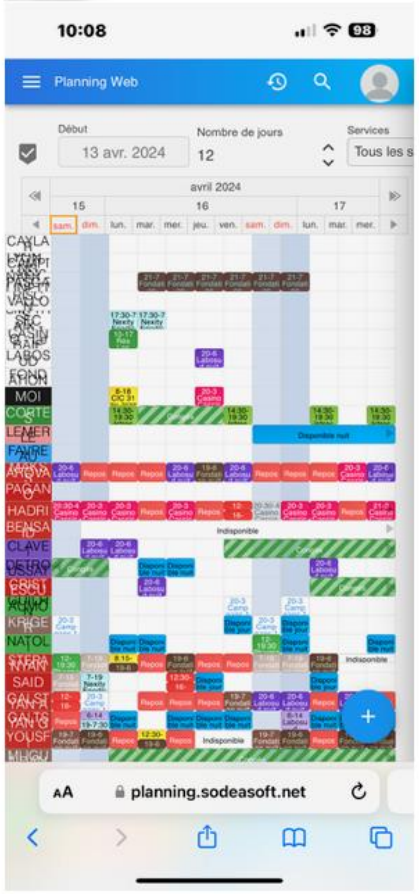
Bizarre

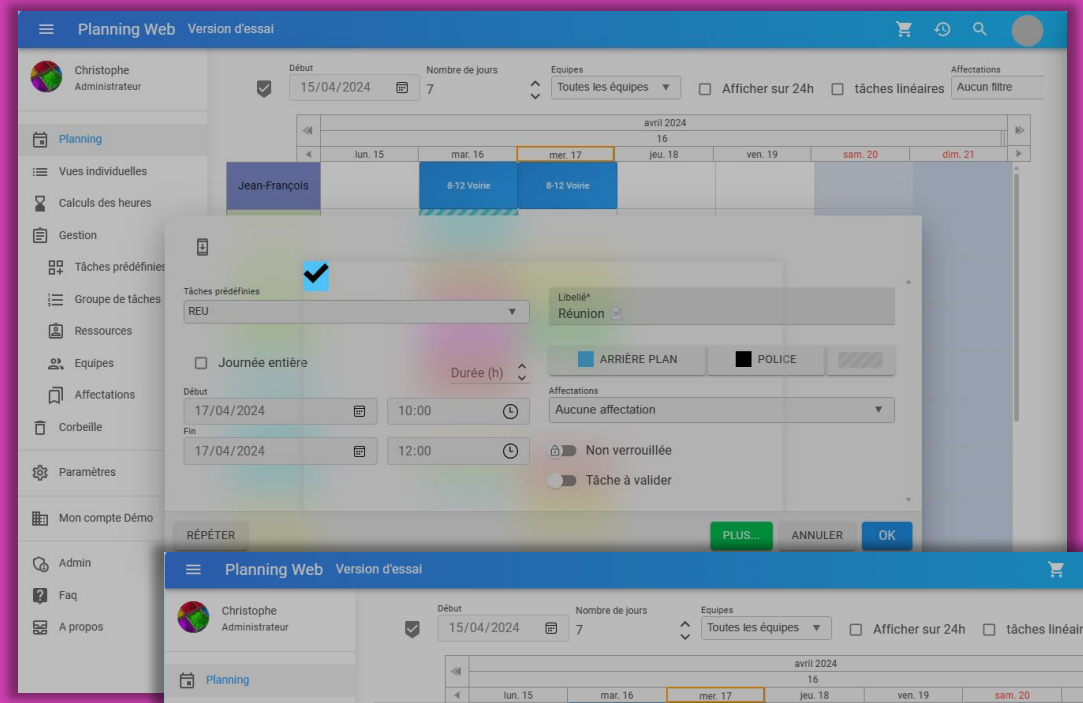
À Christophe PEUGNET - EBLM

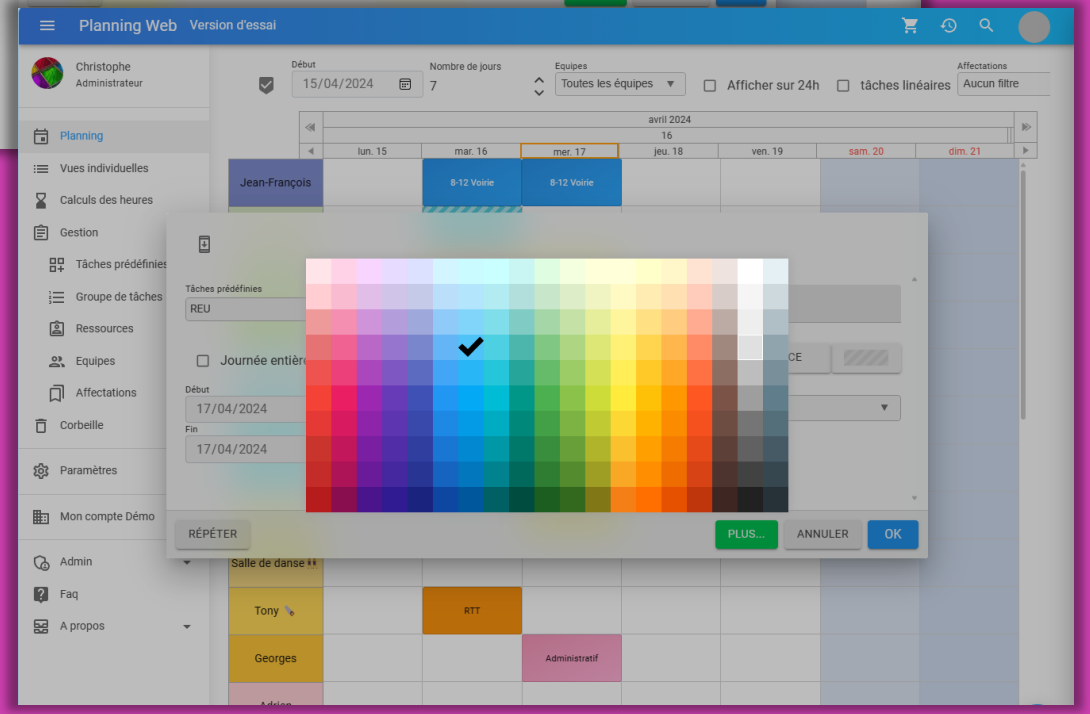
sam. 13/04

Vous avez répondu à ce message le 13/04/2024 11:21.

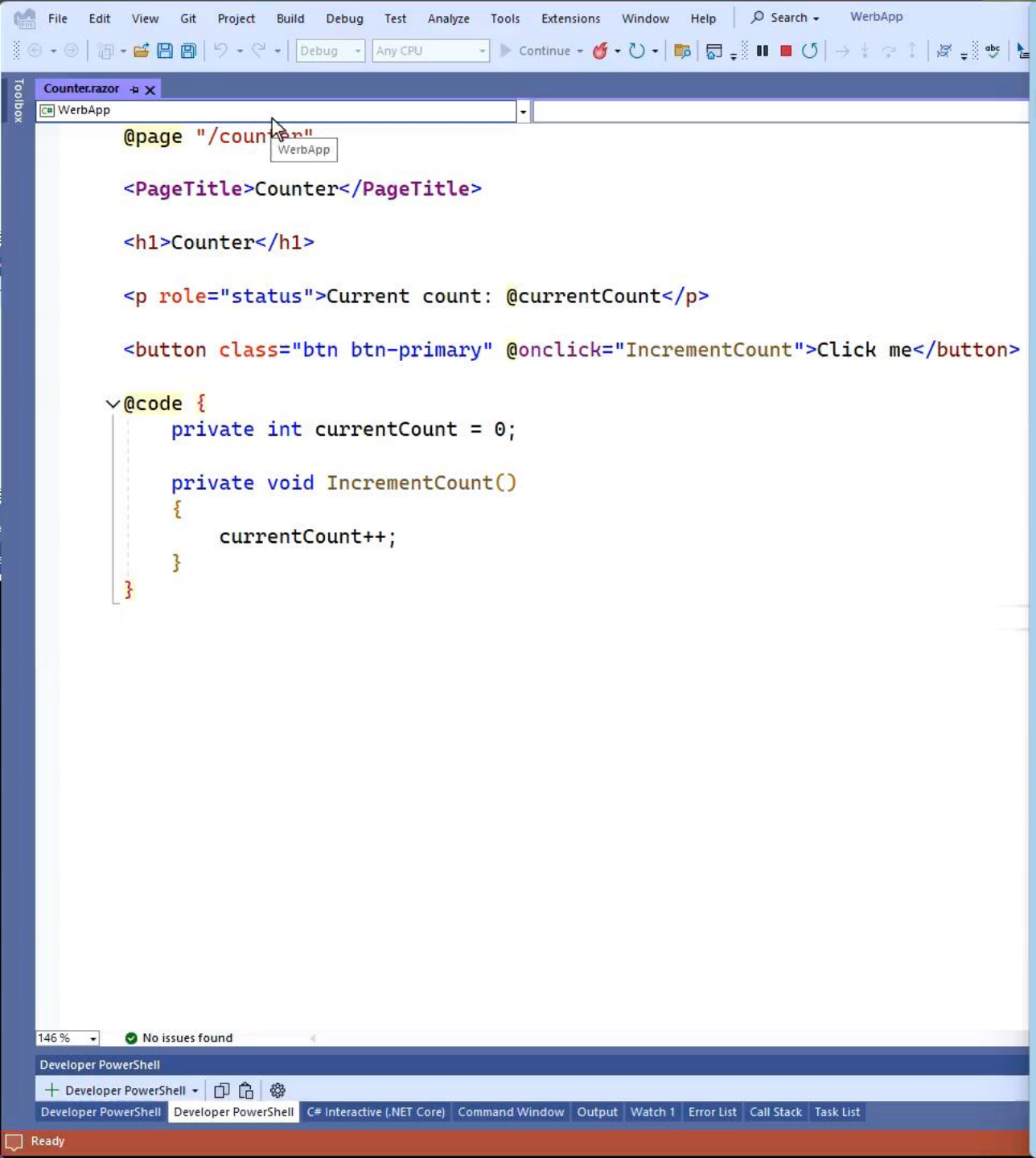
Bonjour M Peugnet,
Depuis 2 jours environ l'affichage sur mon téléphone se présent ainsi.
Doit-je faire un manipulation particulière pour remédier l'afficha des noms des salariés?
Merci bien
Cordialement

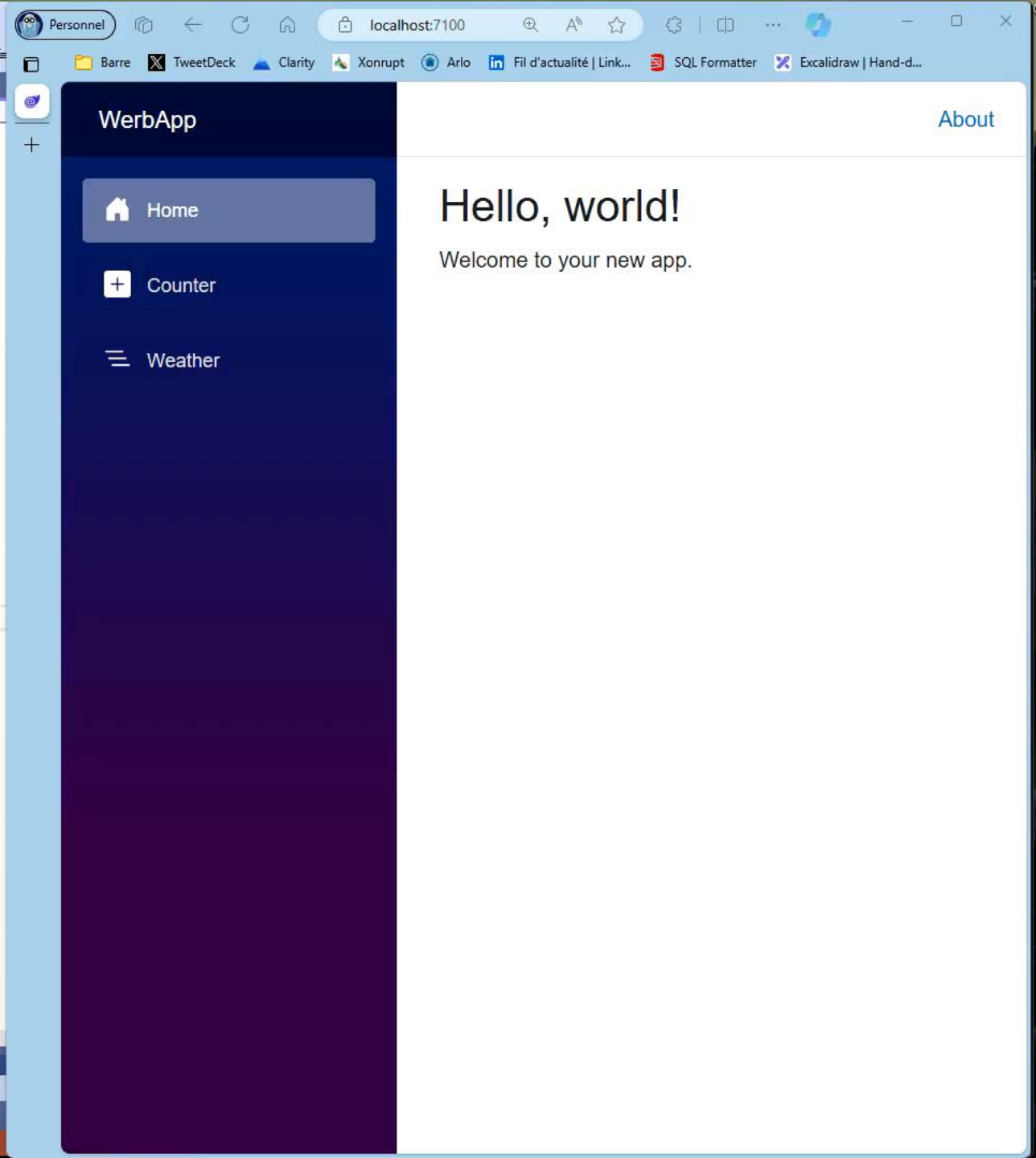


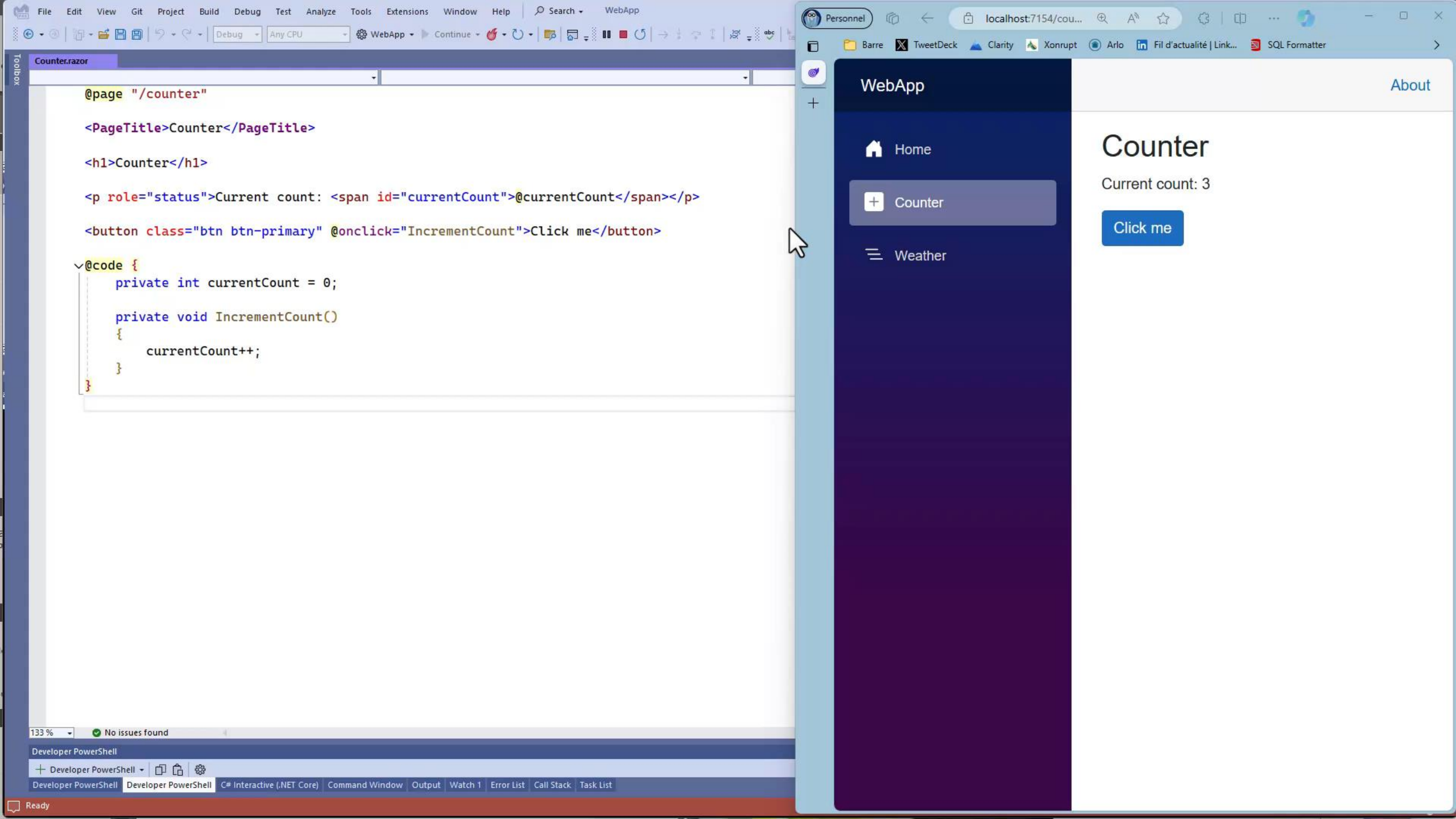




DÉPLOYER UN VENDREDI, QU'ILS DISAIENT


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
WerbApp
Debug
Any CPU
Continue
Toolbox
Counter.razor
WerbApp
@page "/counter"
WerbApp
<PageTitle>Counter</PageTitle>
<h1>Counter</h1>
Current count: @currentCount
<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
@code {
private int currentCount = 0;
private void IncrementCount()
{
currentCount++;
}
}
146 %
No issues found
Developer PowerShell
Developer PowerShell
Developer PowerShell
C# Interactive (.NET Core)
Command Window
Output
Watch 1
Error List
Call Stack
Task List
Ready


Personnel
localhost:7100
Barre
TweetDeck
Clarity
Xonrupt
Arlo
Fil d'actualité | Link...
SQL Formatter
Excalidraw | Hand-d...
WerbApp
About
Home
Counter
Weather
Hello, world!
Welcome to your new app.


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
WebApp
Debug
Any CPU
WebApp
Continue
Counter.razor
Toolbox
@page "/counter"
<PageTitle>Counter</PageTitle>
<h1>Counter</h1>
Current count: @currentCount
<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
@code {
private int currentCount = 0;
private void IncrementCount()
{
currentCount++;
}
}
133 %
No issues found
Developer PowerShell
Developer PowerShell
Developer PowerShell
C# Interactive (.NET Core)
Command Window
Output
Watch 1
Error List
Call Stack
Task List
Ready
Personnel
localhost:7154/cou...
Barre
TweetDeck
Clarity
Xonrupt
Arlo
Fil d'actualité | Link...
SQL Formatter
WebApp
About
Home
Counter
Weather
Counter
Current count: 3
Click me

1 Ajout d'un **projet de test**

dotnet new nunit -n PlaywrightTests -o Test

2 Ajout du **Nuget Microsoft.Playwright**.xxx

dotnet add package Microsoft.PlayWright.NUnit

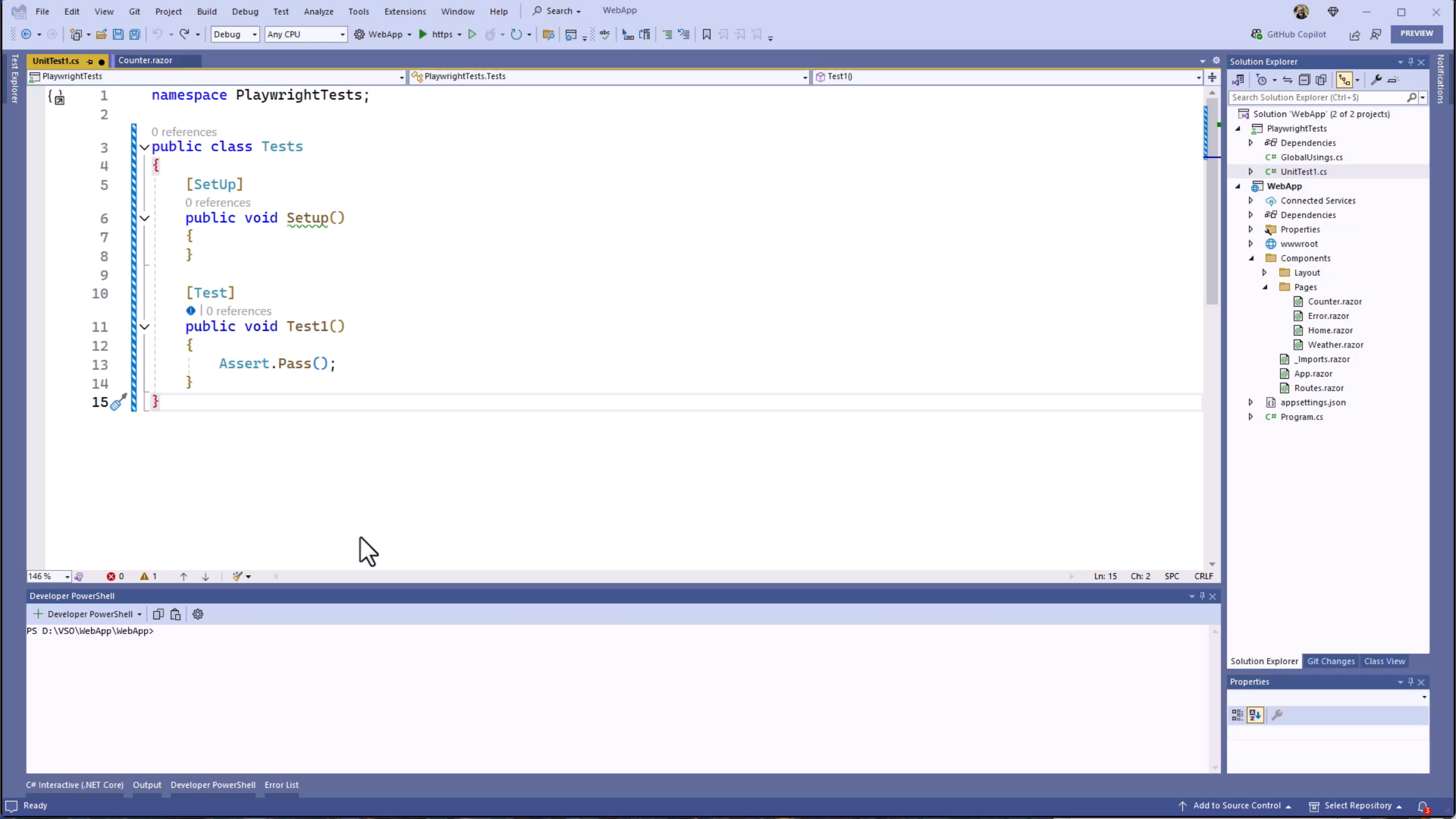
3 Compiler ce projet

4 `\bin\…\playwright.ps1 install`

RÉSUMONS

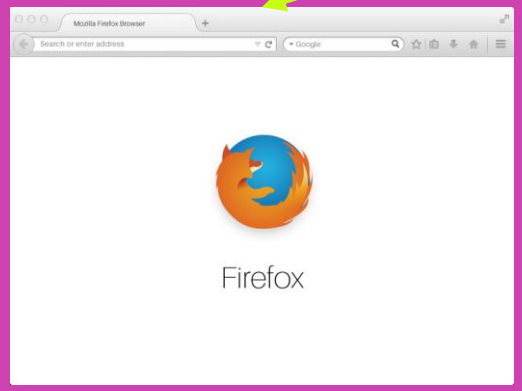
ECRIVONS NOTRE PREMIER TEST

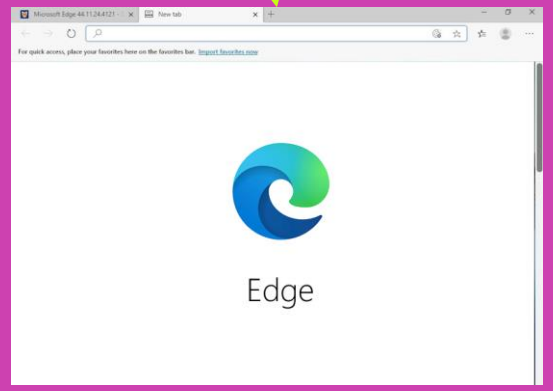
```
namespace PlaywrightTests;

0 references
public class Tests
{
    [SetUp]
    0 references
    public void Setup()
    {
    }

    [Test]
    0 references
    public void Test1()
    {
        Assert.Pass();
    }
}
```

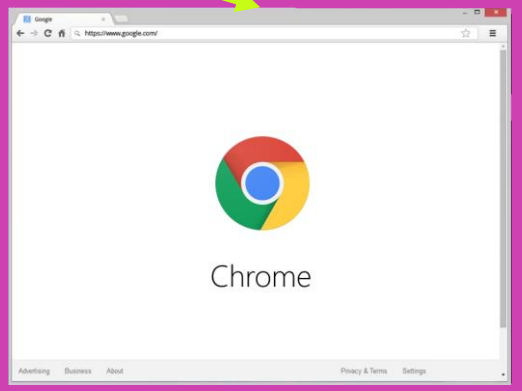


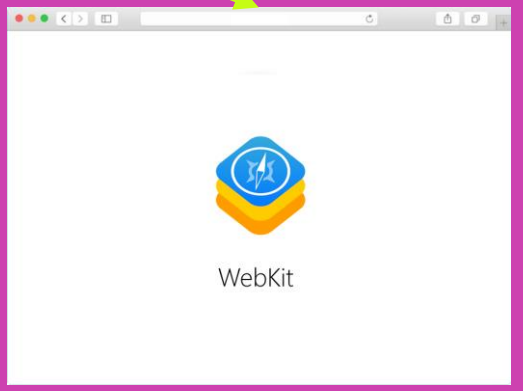
Chrome Devtools Protocol

Web InspectorProtocol









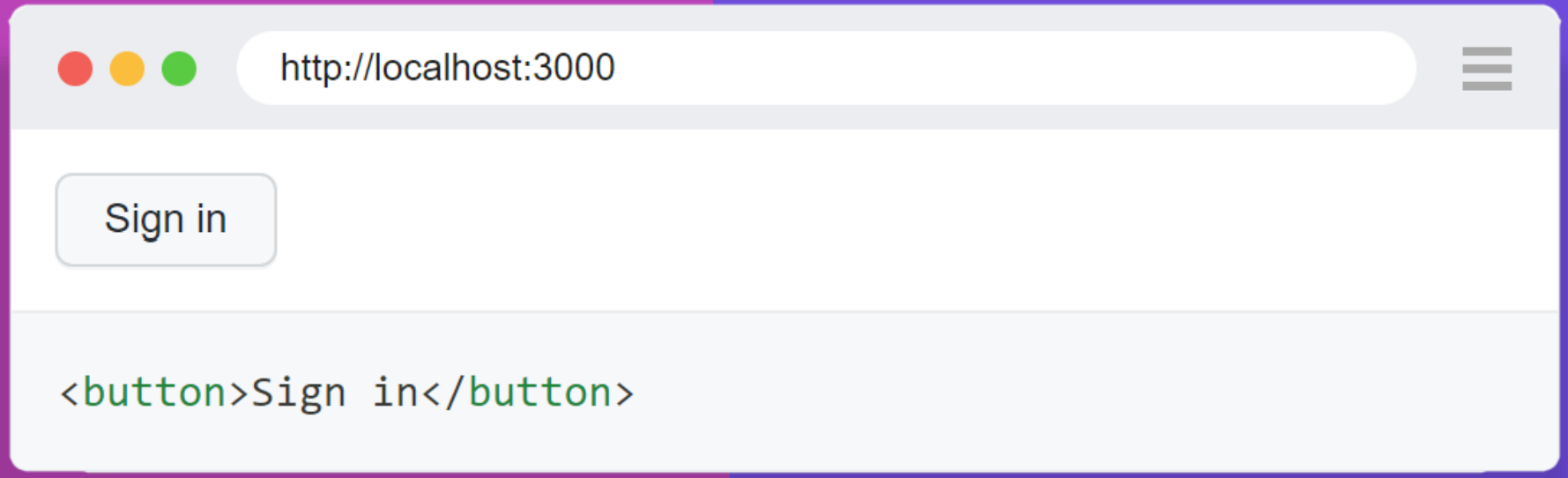
ISOLATION

Tests written with Playwright execute in isolated clean-slate environments called browser contexts. This isolation model improves reproducibility and prevents cascading test failures.

MULTI LANGAGE

Rust, Go,
Ruby,
Crystal,
Perl …

LOCATORS




```
await page.GetByRole(AriaRole.Button, new() { Name = "Sign in" }).ClickAsync();
```

LOCATORS

Quick Guide

These are the recommended built in locators.

- Page.GetByRole() to locate by explicit and implicit accessibility attributes.
- Page.GetByText() to locate by text content.
- Page.GetByLabel() to locate a form control by associated label's text.
- Page.GetByPlaceholder() to locate an input by placeholder.
- Page.GetByAltText() to locate an element, usually image, by its text alternative.
- Page.GetByTitle() to locate an element by its title attribute.
- Page.GetByTestId() to locate an element based on its `data-testid` attribute (other attributes can be configured).

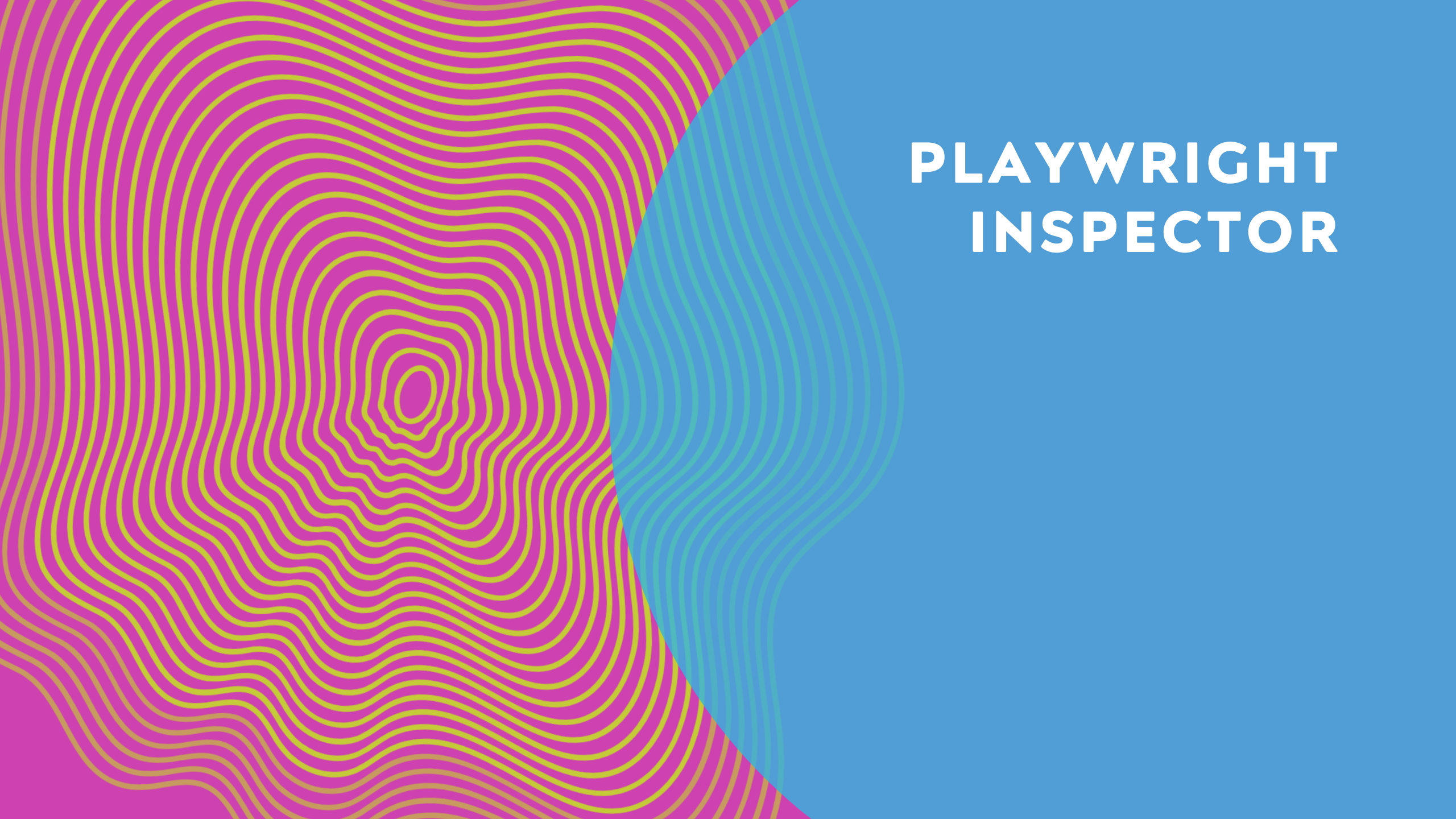
```
await page.GetByLabel("User Name").FillAsync("John");

await page.GetByLabel("Password").FillAsync("secret-password");

await page.GetByRole(AriaRole.Button, new() { Name = "Sign in" }).ClickAsync();

await Expect(Page.GetByText("Welcome, John!")).ToBeVisibleAsync();
```

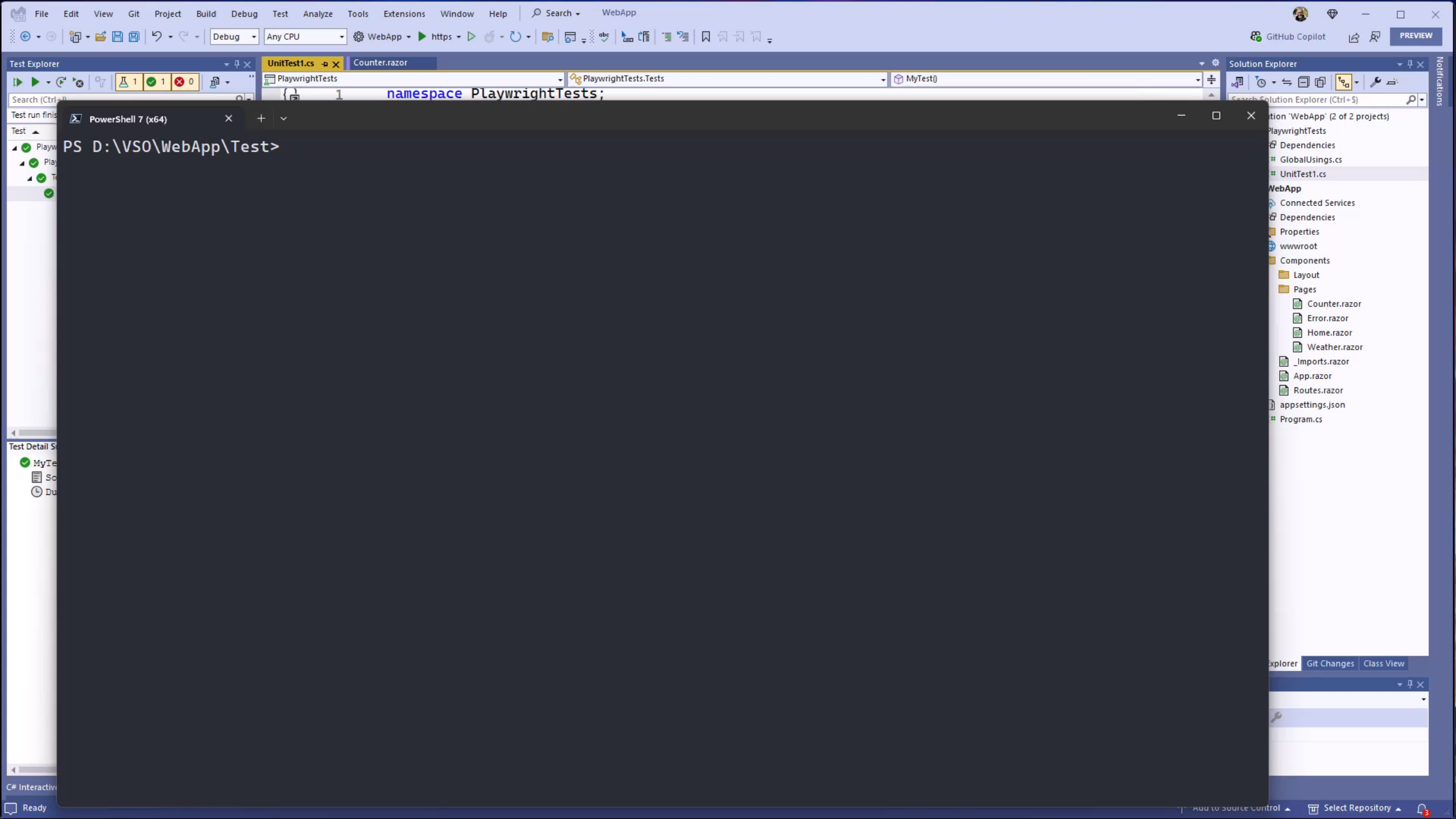
```
// Text input
await page.GetByRole(AriaRole.Textbox).FillAsync("Peter");

// Check the checkbox
await page.GetByLabel("I agree to the terms above").CheckAsync();

// Assert the checked state
Assert.True(await page.GetByLabel("Subscribe to newsletter").IsCheckedAsync());

// Generic click
await page.GetByRole(AriaRole.Button).ClickAsync();

// Double click
await page.GetByText("Item").DblClickAsync();

// Press keys one by one
await page.Locator("#area").TypeAsync("Hello World!");

// Hit Enter
await page.GetByText("Submit").PressAsync("Enter");

// Select multiple files
await page.GetByLabel("Upload files").SetInputFilesAsync(new[] { "file1.txt", "file12.txt" });
```

ASSERTIONS

Assertion	Description
Expect(Locator).ToBeCheckedAsync()	Checkbox is checked
Expect(Locator).ToBeDisabledAsync()	Element is disabled
Expect(Locator).ToBeEditableAsync()	Element is editable
Expect(Locator).ToBeEmptyAsync()	Container is empty
Expect(Locator).ToBeEnabledAsync()	Element is enabled
Expect(Locator).ToBeFocusedAsync()	Element is focused
Expect(Locator).ToBeHiddenAsync()	Element is not visible
Expect(Locator).ToBeVisibleAsync()	Element is visible
Expect(Locator).ToContainTextAsync()	Element contains text
Expect(Locator).ToHaveAttributeAsync()	Element has a DOM attr.

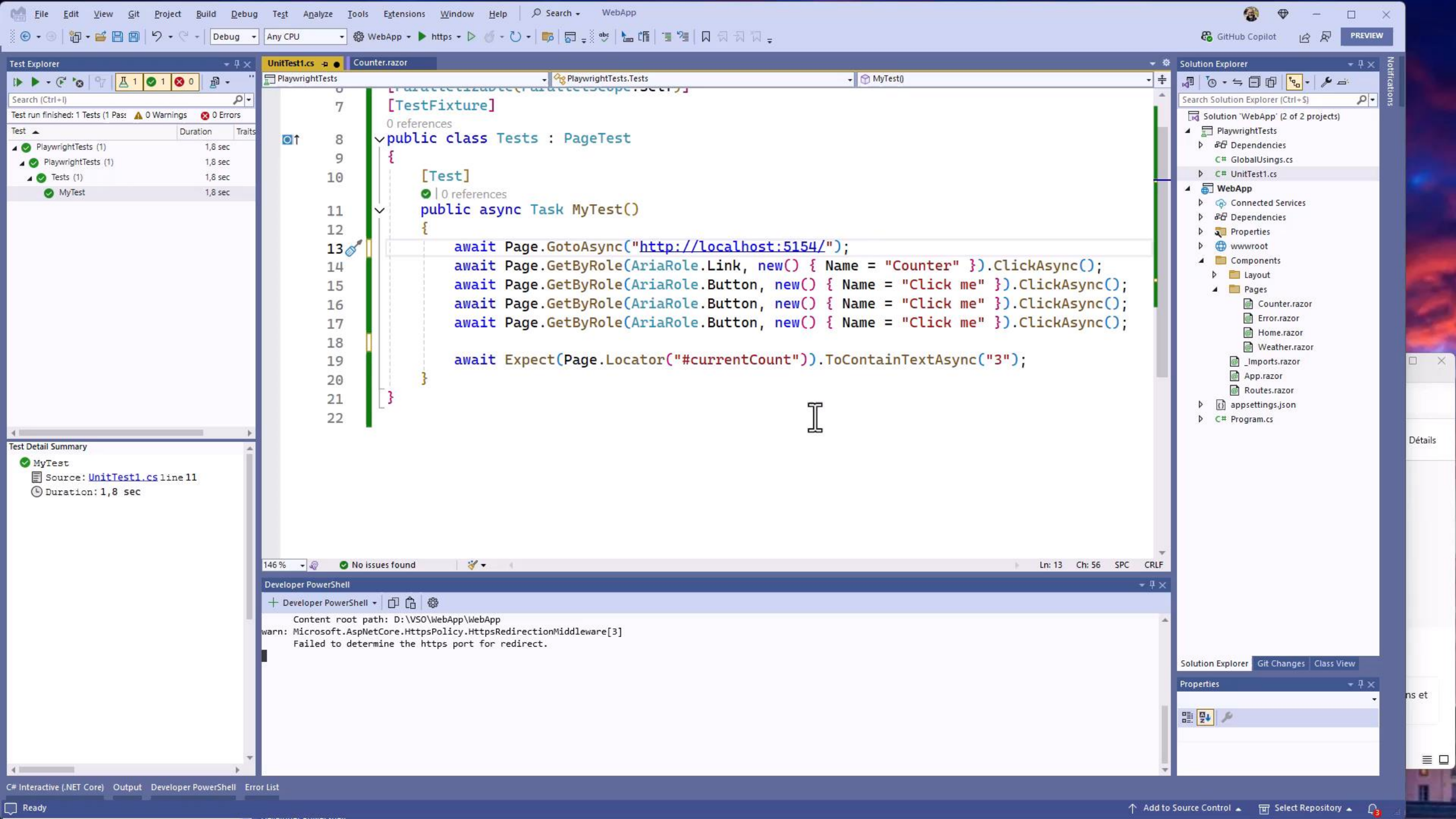
Assertion	Description
Expect(Locator).ToHaveClassAsync()	Element has a class property
Expect(Locator).ToHaveCountAsync()	List has exact number of children
Expect(Locator).ToHaveCSSAsync()	Element has CSS property
Expect(Locator).ToHaveIdAsync()	Element has an ID
Expect(Locator).ToHaveJSPropertyAsync()	Element has a JavaScript property
Expect(Locator).ToHaveTextAsync()	Element matches text
Expect(Locator).ToHaveValueAsync()	Input has a value
Expect(Locator).ToHaveValuesAsync()	Select has options selected
Expect(Page).ToHaveTitleAsync()	Page has a title
Expect(Page).ToHaveURLAsync()	Page has a URL
Expect(ApiResponse).ToBeOKAsync()	Response has an OK status


PLAYWRIGHT
INSPECTOR

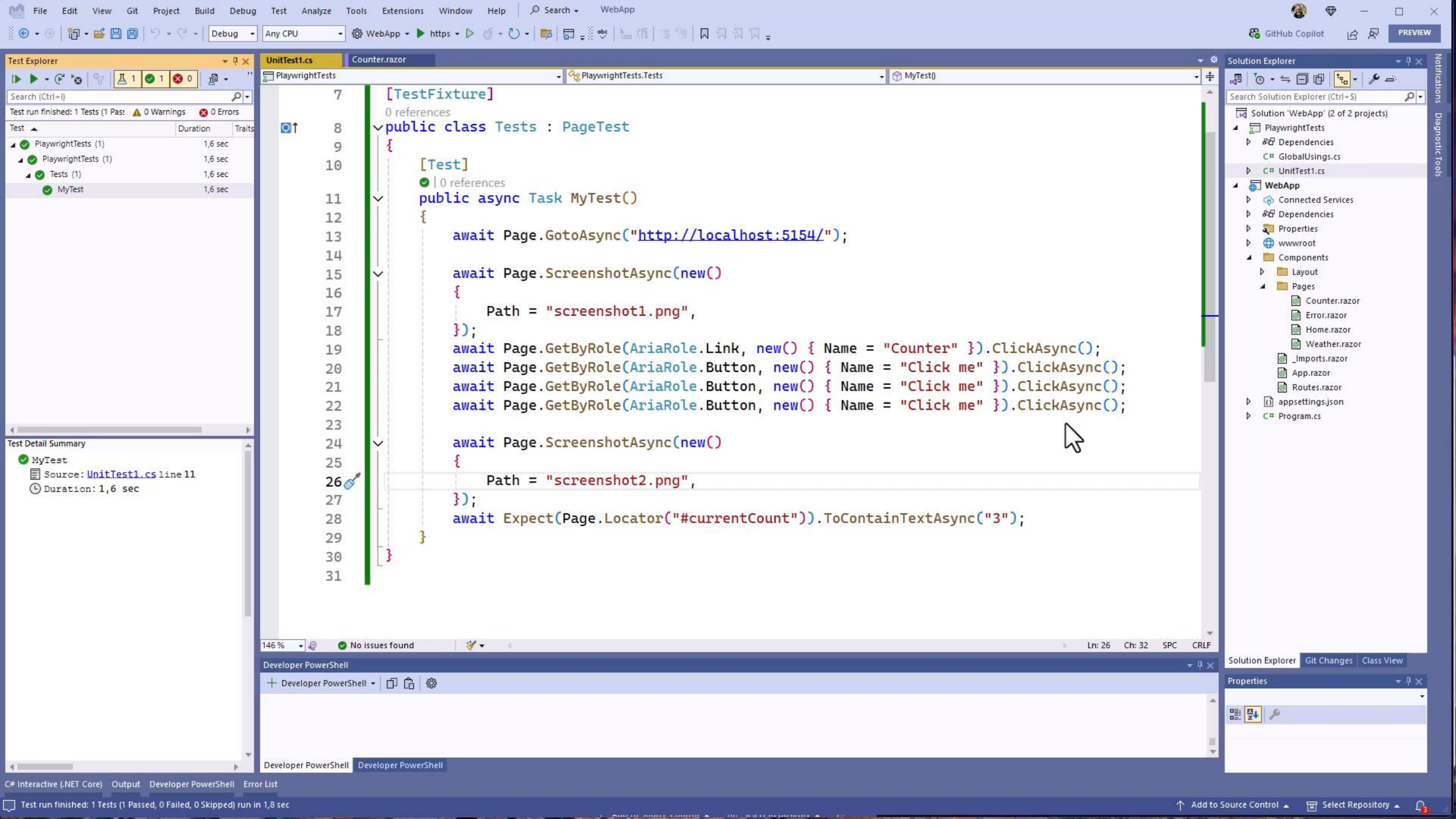

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
WebApp
Debug
Any CPU
WebApp
https
GitHub Copilot
PREVIEW
Test Explorer
UnitTest1.cs
Counter.razor
PlaywrightTests
PlaywrightTests.Tests
MyTest()
namespace PlaywrightTests;
Solution Explorer
PowerShell 7 (x64)
PS D:\VSO\WebApp\Test>
Dependencies
GlobalUsings.cs
UnitTest1.cs
WebApp
Connected Services
Dependencies
Properties
wwwroot
Components
Layout
Pages
Counter.razor
Error.razor
Home.razor
Weather.razor
_Imports.razor
App.razor
Routes.razor
appsettings.json
Program.cs
Git Changes
Class View
Ready
Select Repository

UN SELFIE ?

(AMAZING FEATURES)


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
WebApp
Debug
Any CPU
WebApp
https
GitHub Copilot
PREVIEW
Test Explorer
Search (Ctrl+I)
Test run finished: 1 Tests (1 Pass) 0 Warnings 0 Errors
Test
Duration
Traits
PlaywrightTests (1) 1,8 sec
PlaywrightTests (1) 1,8 sec
Tests (1) 1,8 sec
MyTest 1,8 sec
Test Detail Summary
MyTest
Source: UnitTest1.cs line 11
Duration: 1,8 sec
UnitTest1.cs
Counter.razor
PlaywrightTests
PlaywrightTests.Tests
MyTest()
[TestFixture]
0 references
public class Tests : PageTest
{
[Test]
0 references
public async Task MyTest()
{
await Page.GotoAsync("http://localhost:5154/");
await Page.GetByRole(AriaRole.Link, new() { Name = "Counter" }).ClickAsync();
await Page.GetByRole(AriaRole.Button, new() { Name = "Click me" }).ClickAsync();
await Page.GetByRole(AriaRole.Button, new() { Name = "Click me" }).ClickAsync();
await Page.GetByRole(AriaRole.Button, new() { Name = "Click me" }).ClickAsync();
await Expect(Page.Locator("#currentCount")).ToContainTextAsync("3");
}
}
146 %
No issues found
Ln: 13 Ch: 56 SPC CRLF
Developer PowerShell
Developer PowerShell
Content root path: D:\VSO\WebApp\WebApp
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
Failed to determine the https port for redirect.
Solution Explorer
Search Solution Explorer (Ctrl+$)
Solution 'WebApp' (2 of 2 projects)
PlaywrightTests
Dependencies
GlobalUsings.cs
UnitTest1.cs
WebApp
Connected Services
Dependencies
Properties
wwwroot
Components
Layout
Pages
Counter.razor
Error.razor
Home.razor
Weather.razor
_Imports.razor
App.razor
Routes.razor
appsettings.json
Program.cs
Solution Explorer
Git Changes
Class View
Properties
Notifications
C# Interactive (.NET Core)
Output
Developer PowerShell
Error List
Ready
Add to Source Control
Select Repository

LE DEBUG ?


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
WebApp
Debug
Any CPU
WebApp
https
GitHub Copilot
PREVIEW
Test Explorer
Search (Ctrl+I)
Test run finished: 1 Tests (1 Pass 0 Warnings 0 Errors
Test
Duration
Traits
PlaywrightTests (1) 1,6 sec
PlaywrightTests (1) 1,6 sec
Tests (1) 1,6 sec
MyTest 1,6 sec
Test Detail Summary
MyTest
Source: UnitTest1.cs line 11
Duration: 1,6 sec
UnitTest1.cs
Counter.razor
PlaywrightTests
PlaywrightTests.Tests
MyTest()
[TestFixture]
0 references
public class Tests : PageTest
{
[Test]
0 references
public async Task MyTest()
{
await Page.GotoAsync("http://localhost:5154/");
await Page.ScreenshotAsync(new()
{
Path = "screenshot1.png",
});
await Page.GetByRole(AriaRole.Link, new() { Name = "Counter" }).ClickAsync();
await Page.GetByRole(AriaRole.Button, new() { Name = "Click me" }).ClickAsync();
await Page.GetByRole(AriaRole.Button, new() { Name = "Click me" }).ClickAsync();
await Page.GetByRole(AriaRole.Button, new() { Name = "Click me" }).ClickAsync();
await Page.ScreenshotAsync(new()
{
Path = "screenshot2.png",
});
await Expect(Page.Locator("#currentCount")).ToContainTextAsync("3");
}
}
146 %
No issues found
Ln: 26 Ch: 32 SPC CRLF
Developer PowerShell
Developer PowerShell
Developer PowerShell
Developer PowerShell
Solution Explorer
Search Solution Explorer (Ctrl+S)
Solution 'WebApp' (2 of 2 projects)
PlaywrightTests
Dependencies
GlobalUsings.cs
UnitTest1.cs
WebApp
Connected Services
Dependencies
Properties
wwwroot
Components
Layout
Pages
Counter.razor
Error.razor
Home.razor
Weather.razor
_Imports.razor
App.razor
Routes.razor
appsettings.json
Program.cs
Solution Explorer
Git Changes
Class View
Properties
Notifications
Diagnostic Tools
C# Interactive (.NET Core)
Output
Developer PowerShell
Error List
Test run finished: 1 Tests (1 Passed, 0 Failed, 0 Skipped) run in 1,8 sec
Add to Source Control
Select Repository

REACT LOCATORS

```
await page.Locator("_react=BookItem").ClickAsync();
```

match by **component**:	`_react=BookItem`
match by **component** and **exact property value**, **case-sensitive**:	`_react=BookItem[author = "Steven King"]`
match by **property value only**, **case-insensitive**:	`_react=[author = "steven king" i]`
match by **component** and **truthy property value**:	`_react=MyButton[enabled]`
match by **component** and **boolean value**:	`_react=MyButton[enabled = false]`
match by **property value substring**:	`_react=[author *= "King"]`
match by **component** and **multiple properties**:	`_react=BookItem[author *= "king" i][year = 1990]`
match by **nested property value**:	`_react=[some.nested.value = 12]`
match by **component** and **property value prefix**:	`_react=BookItem[author ^= "Steven"]`
match by **component** and **property value suffix**:	`_react=BookItem[author $= "Steven"]`
match by **component** and **key**:	`_react=BookItem[key = '2']`
match by **property value regex**:	`_react=[author = /Steven(\\s+King)?/i]`

OTHER LOCATORS

- CSS locator
- N-th element locator
- Parent element locator
- Combining two alternative locators
- Locating only visible elements
- Vue locator
- React locator
- XPath locatorLabel to form control retargeting
- Legacy text locator
- id, data-testid, data-test-id, data-test selectors
- Chaining selectors

API TESTING

API testing | Playwright .NET

```
public class TestGitHubAPI : PlaywrightTest
{
    static string API_TOKEN = Environment.GetEnvironmentVariable("GITHUB_A

    private IAPIRequestContext Request = null;

    [SetUp]
    public async Task SetUpAPITesting()
    {
        await CreateAPIRequestContext();
    }

    private async Task CreateAPIRequestContext()
    {
        var headers = new Dictionary<string, string>();
        // We set this header per GitHub guidelines.
        headers.Add("Accept", "application/vnd.github.v3+json");
        // Add authorization token to all requests.
        // Assuming personal access token available in the environment.
        headers.Add("Authorization", "token " + API_TOKEN);

        Request = await this.Playwright.APIRequest.NewContextAsync(new() {
            // All requests we send go to this API endpoint.
            BaseURL = "https://api.github.com",
            ExtraHTTPHeaders = headers,
        });
    }

    [TearDown]
    public async Task TearDownAPITesting()
    {
        await Request.DisposeAsync();
    }
}
```

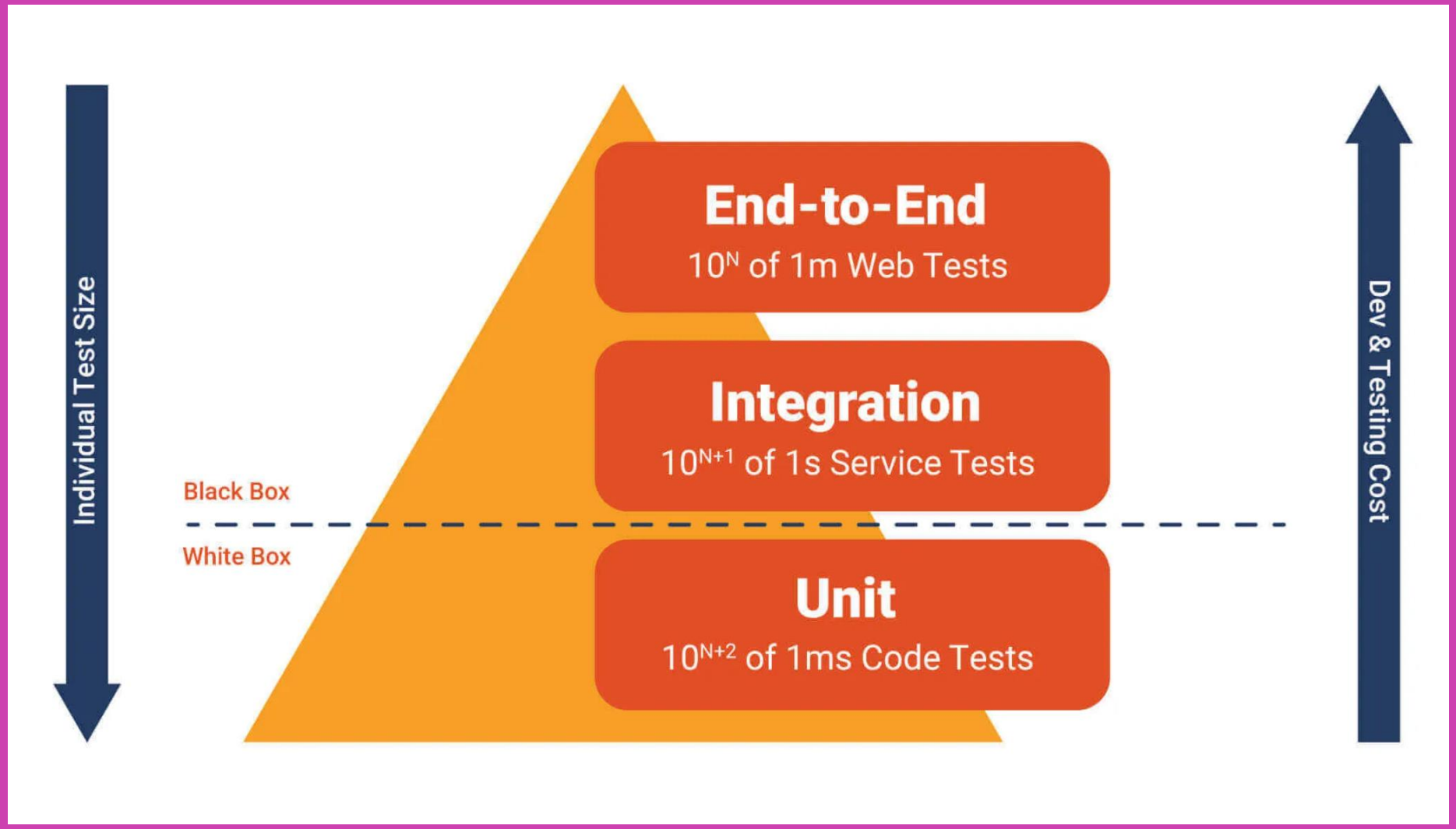
ACCESS-IBIL-ITY TESTING

https://npmjs.org/@axe-core/playwright

```
const { AxeBuilder } = require('@axe-core/playwright');
const playwright = require('playwright');

(async () => {
  const browser = await playwright.chromium.launch({ headless: true });
  const context = await browser.newContext();
  const page = await context.newPage();
  await page.goto('https://dequeuniversity.com/demo/mars/');

  try {
    const results = await new AxeBuilder({ page }).analyze();
    console.log(results);
  } catch (e) {
    // do something with the error
  }

  await browser.close();
})();
```

TESTING ? TEST IS DOUBT

TEST PYRAMID



GET FEEDBACKS

- Trace
- Screenshots
- Videos

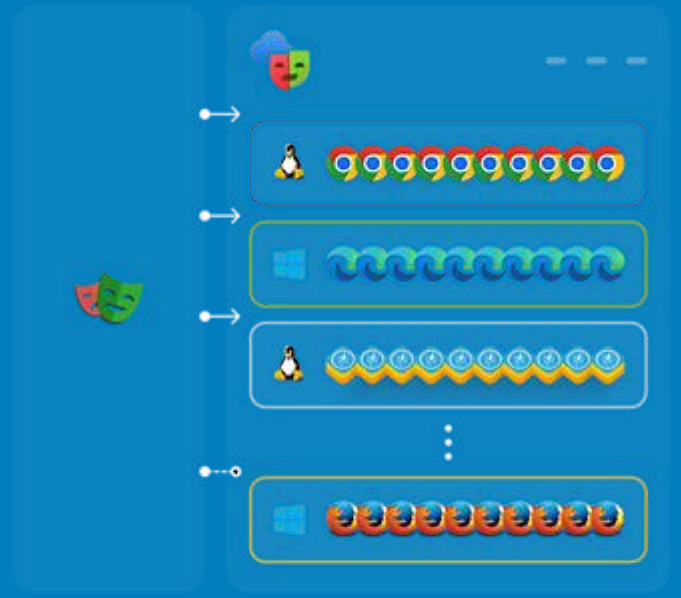
PLAYWRIGHT TESTING SERVICE

Exécuter des tests Playwright depuis Azure avec différentes combinaisons de navigateurs et d'OS

RUN PLAYWRIGHT TESTS AT SCALE

Exécuter les suites de tests plus rapidement

Distribuer les tests sur les navigateurs avec une parallélisation plus élevée dans CI pour exécuter les suites de tests plus rapidement.



EXÉCUTER LES TESTS PLAYWRIGHT DE MANIÈRE FLEXIBLE

Parallel browsers

Distribuez les tests pour qu'ils s'exécutent en parallèle sur un maximum de 50 navigateurs et dépassez les ressources locales pour accélérer les tests.

OS and browser testing combinations

Utilisez n'importe quel appareil pour tester les systèmes d'exploitation Linux et Windows et les navigateurs modernes tels que Chromium, WebKit et Firefox.

Endpoint testing

Testez des applications accessibles au public et en privé ou exécutez des tests sur un serveur de développement local.

Test scripts

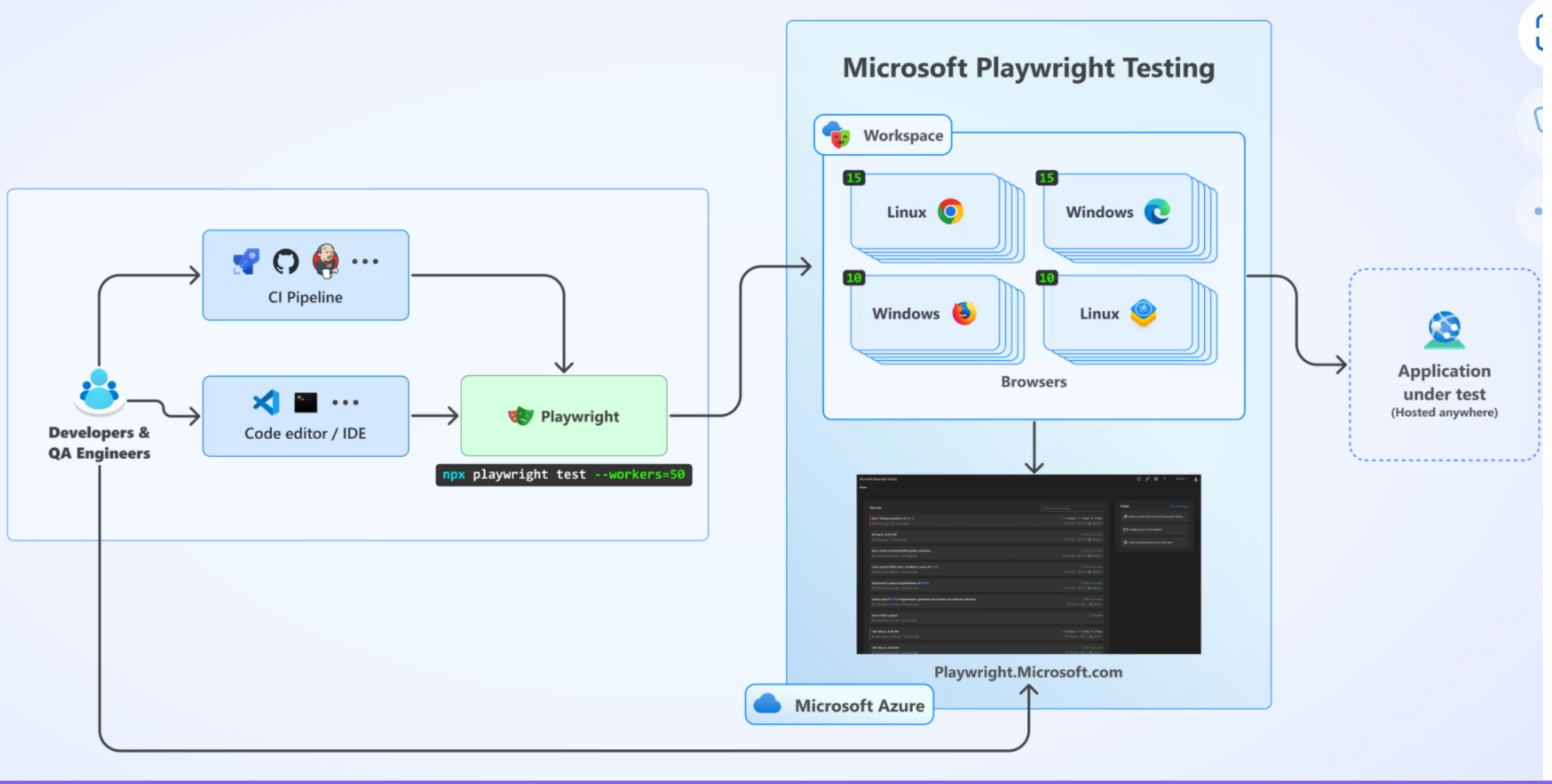
Intégrez de manière transparente votre suite de tests existante sans modifier le code de test.

Consistent regional performance

Exécutez vos tests automatiquement sur des navigateurs hébergés dans la région Azure la plus proche.

TARIFICATION

Essai de 30 jours	**Linux** **Système d'exploitation du navigateur cloud**	**Windows** **Système d'exploitation du navigateur cloud**
Les 100 premières minutes de test sont gratuites avec un essai de 30 jours.	**0,010 €** / 1 minute d'essai	**0,019 €** / 1 minute d'essai



Azure Playwright Testing Service

GOOD PRACTISES

Trip & Tricks - Page object models

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

namespace BigEcommerceApp.Tests.Models;

public class SearchPage
{
  private readonly IPage _page;
  private readonly ILocator _searchTermInput;

  public SearchPage(IPage page)
  {
    _page = page;
    _searchTermInput = page.Locator("[aria-label='Enter your search term']");
  }

  public async Task GotoAsync()
  {
    await _page.GotoAsync("https://bing.com");
  }

  public async Task SearchAsync(string text)
  {
    await _searchTermInput.FillAsync(text);
    await _searchTermInput.PressAsync("Enter");
  }
}
```

```cs
using BigEcommerceApp.Tests.Models;

// in the test
var page = new SearchPage(await browser.NewPageAsync());
await page.GotoAsync();
await page.SearchAsync("search query");
```

PLAYWRIGHT

reliable end-to-end testing for modern web apps

Any browser – Any platform – One API

Full Isolation – Fast execution

Powerful Tooling – Multi Language

Christophe Peugnet - Adrien Clerbois

MICROSOFT MVP, TECHNICAL ARCHITECT @ SENSE OF TECH
Follow us on social media @tossnet1 @aclerbois