



Testing web applications with Playwright

It's time to play your tests wright

Debbie O'Brien

Senior Program Manager at Microsoft

 Playwright Advocate

International Speaker

Teacher

YouTuber

Open Source Contributor

Love Sport: running, cycling, skiing, padel,
4th degree black belt in Taekwondo

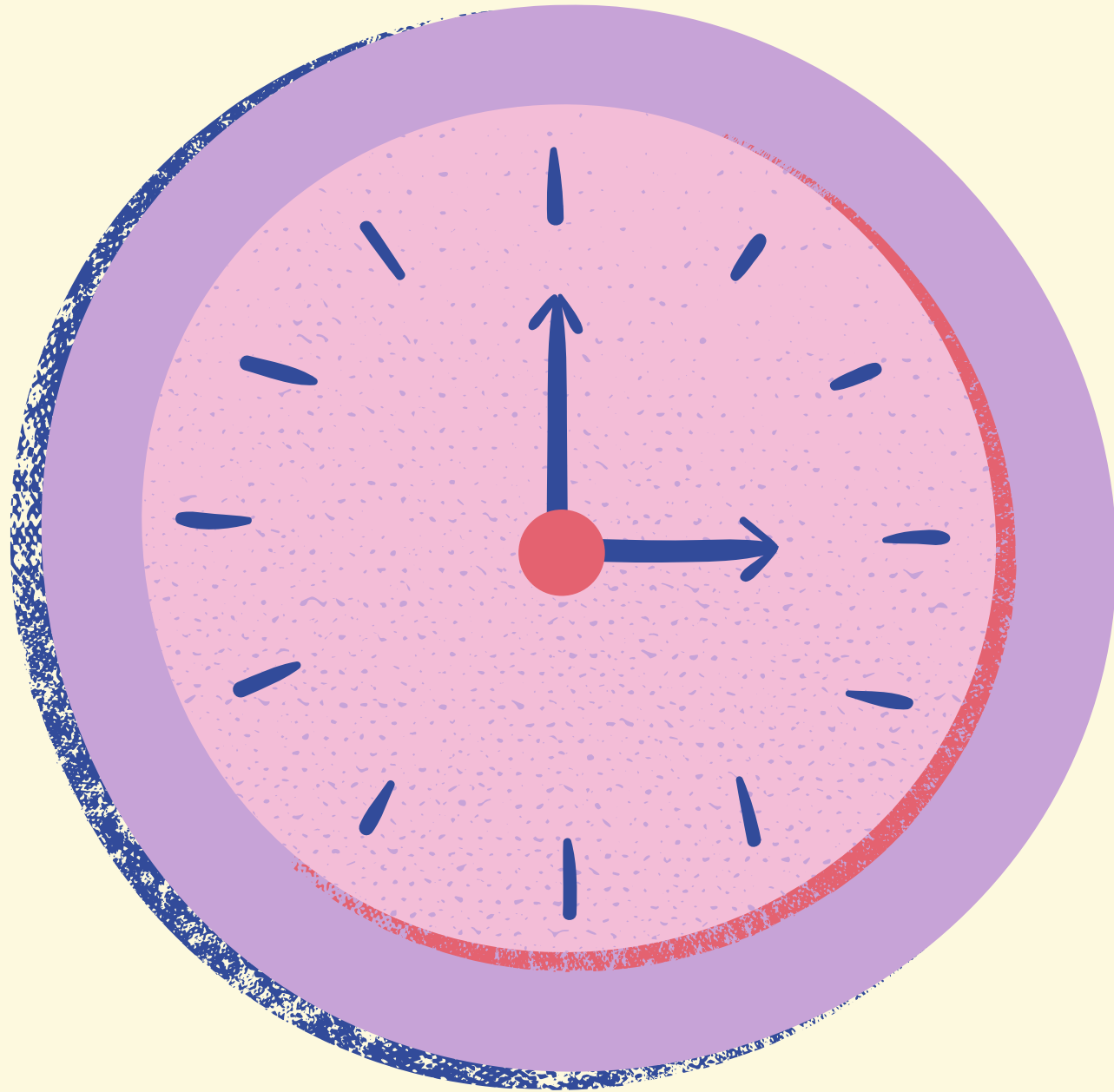


Experts



debs_obrien

Testing



is hard

takes time to learn

takes time to build

time === money

culture

How can we get more developers to
write tests?

Playwright Testing

Reliable end-to-end testing for modern web apps

Any browser • Any platform • One API

Full isolation • Fast execution

Powerful Tooling



Getting Started

```
● ● ●  
  
# Create project  
dotnet new console -n PlaywrightDemo  
cd PlaywrightDemo  
  
# Add project dependency  
dotnet add package Microsoft.Playwright  
# Build the project  
dotnet build  
# Install required browsers - replace netX with actual output folder name, f.ex. net6.0.  
pwsh bin\Debug\netX\playwright.ps1 install
```

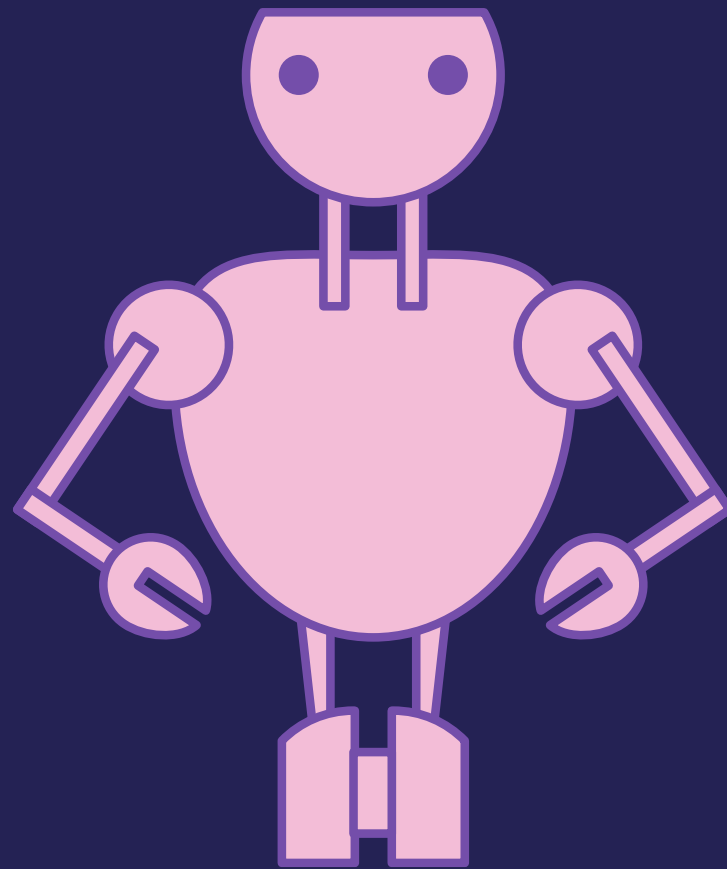



After Install

Install NUnit + Microsoft.Playwright.NUnit

Create a test file


Run the test



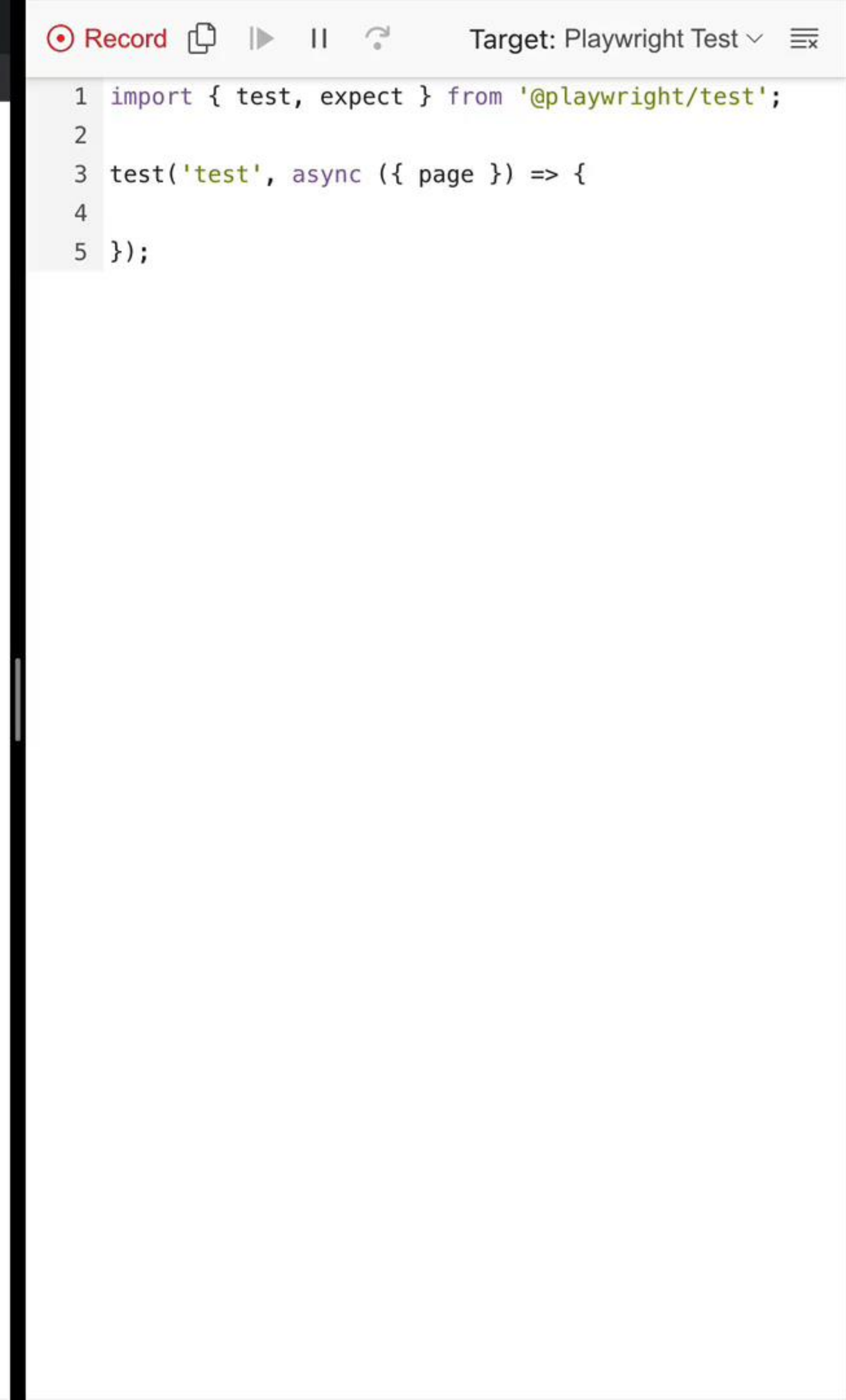
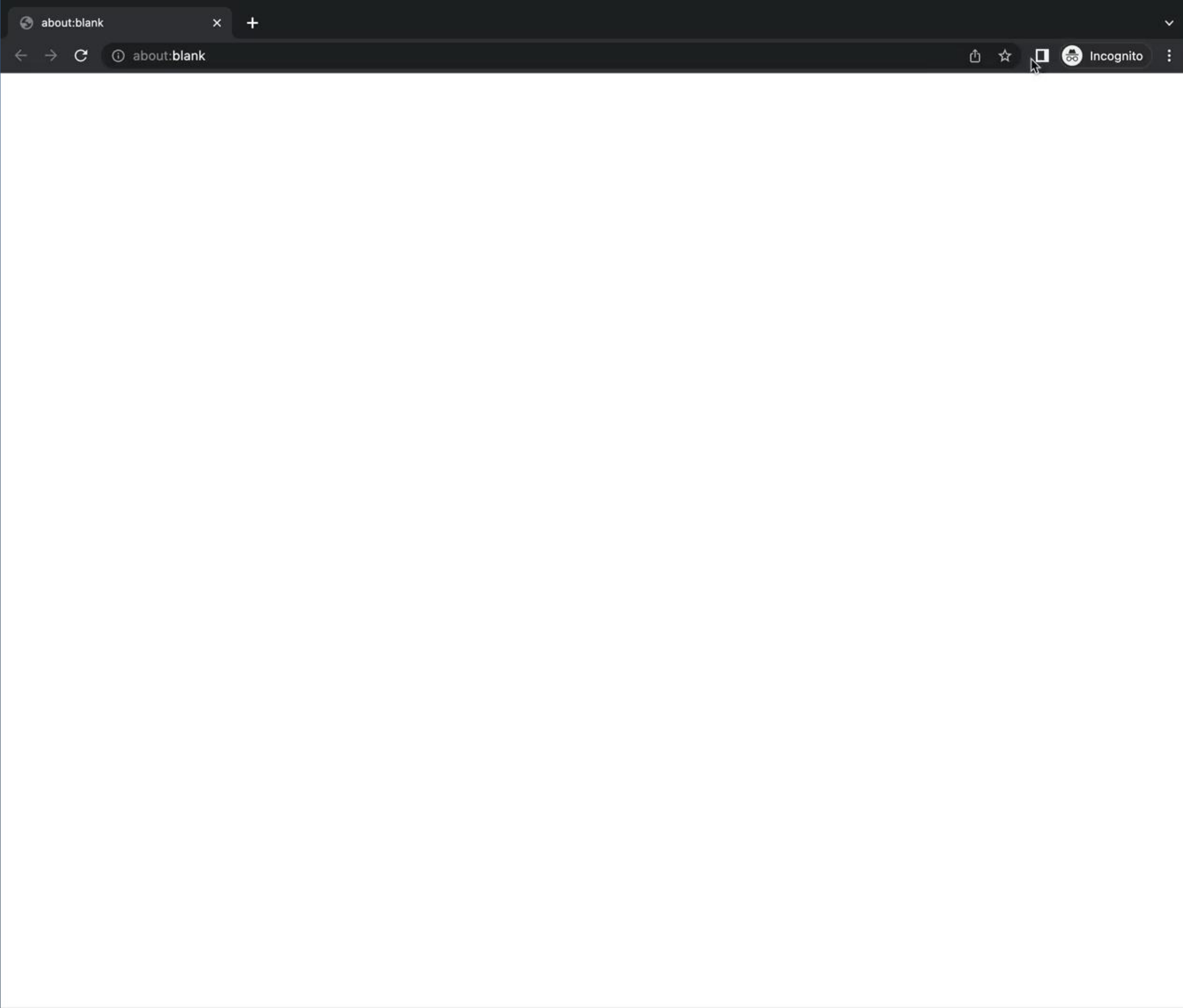
Codegen

Generate tests by recording your actions

Codegen



```
pwsh bin\Debug\netX\playwright.ps1 codegen
```





Multiple Languages

Java

JavaScript

TypeScript

Python

C#



debs_obrien

.NET Podcasts - Discover


dotnetpodcasts.azurewebsites.net/discover

Search here


See all categories

Microsoft Mobile Desktop Community M365


What's new



.future
Microsoft / Gimlet Creative



Hello World (Audio) -...
Microsoft



The .NET MAUI Podcast
Xamarin

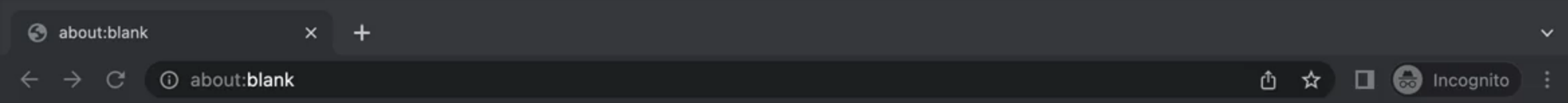
Specially for you

Record

Target: C#

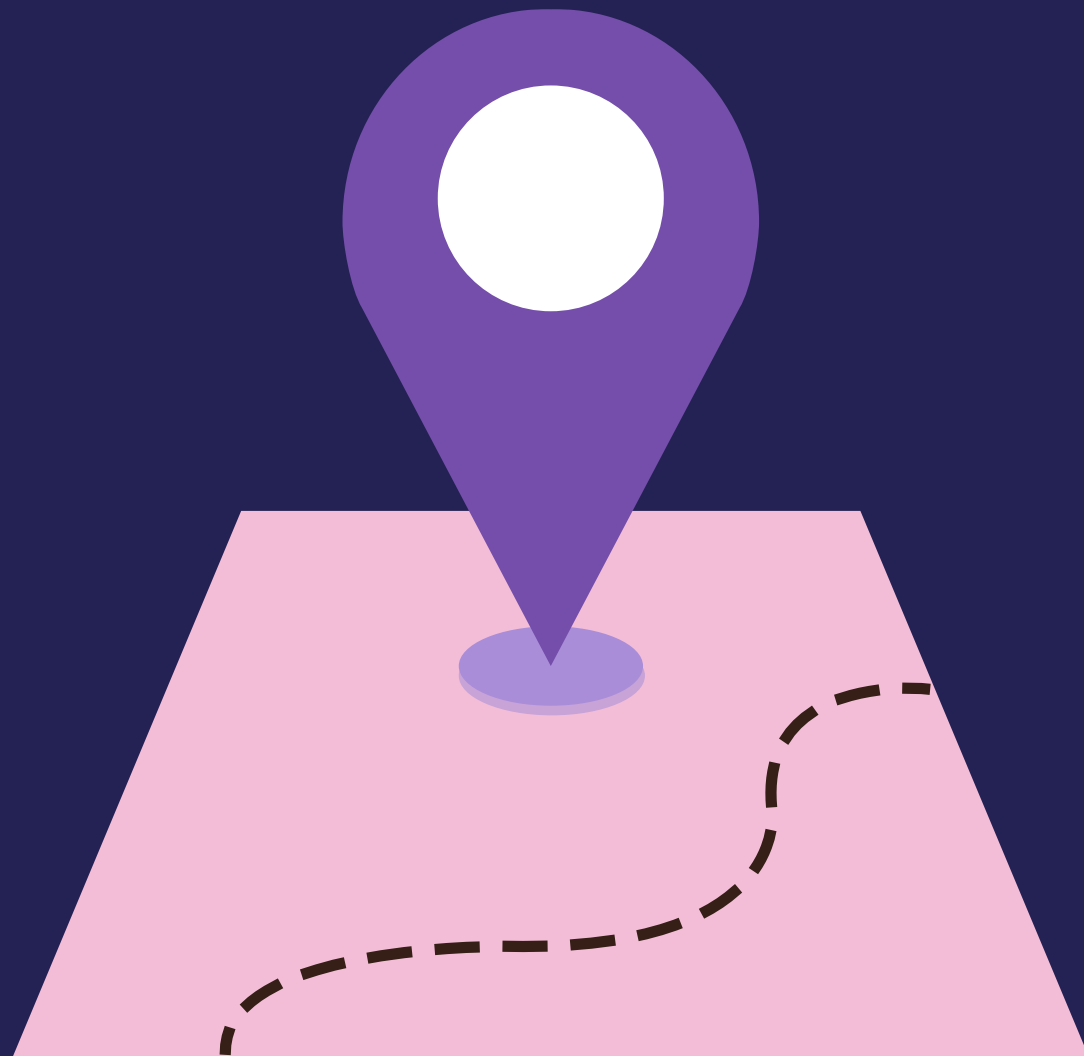
```
1 using Microsoft.Playwright;  
2 using System;  
3 using System.Threading.Tasks;  
4  
5 class Program  
6 {  
7     public static async Task Main()  
8     {  
9         using var playwright = await Playwright.Crea  
10        await using var browser = await playwright.C  
11        {  
12            Headless = false,  
13        });  
14        var context = await browser.NewContextAsync(  
15    }  
16 }  
17
```

Explore Playwright Selector



Record [copy] [play] [pause] [refresh] Target: C# [dropdown] [menu]

```
1 using Microsoft.Playwright;
2 using System;
3 using System.Threading.Tasks;
4
5 class Program
6 {
7     public static async Task Main()
8     {
9         using var playwright = await Playwright.CreateAsync();
10        await using var browser = await playwright.Chromium.La
11        {
12            Headless = false,
13        });
14        var context = await browser.NewContextAsync();
15
16        // Open new page
17        var page = await context.NewPageAsync();
18    }
19 }
20
```

Locators

a way to find element(s) on the page at any moment with built in auto-waiting and retry-ability

Web First Assertions

Created with the *page.locator(selector[, options])* method

strict by default

```
await Expect(page.Locator('header >> nav'))  
  .toHaveTextAsync(['Features', 'Featured Podcasts', 'Available on', 'Powered by .NET']);
```

Must be awaited

Re-queries given selectors

waits until it has four elements with
given texts

Web First Assertions

locators for everything

- `Expect(Locator).toBeCheckedAsync()`
- `Expect(Locator).toBeDisabledAsync()`
- `Expect(Locator).toBeEditableAsync()`
- `Expect(Locator).toBeEmptyAsync()`
- `Expect(Locator).toBeEnabledAsync()`
- `Expect(Locator).toBeFocusedAsync()`
- `Expect(Locator).toBeHiddenAsync()`
- `Expect(Locator).toBeVisibleAsync()`
- `Expect(Locator).toContainTextAsync(text)`
- `Expect(Locator).toHaveAttributeAsync(name)`

- `Expect(Locator).toHaveClassAsync(expected)`
- `Expect(Locator).toHaveCountAsync(count)`
- `Expect(Locator).toHaveCSSAsync(name, value)`
- `Expect(Locator).toHaveIdAsync(id)`
- `Expect(Locator).toHaveJSPropertyAsync(name, value)`
- `Expect(Locator).toHaveTextAsync(expected)`
- `Expect(Locator).toHaveTitleAsync(title)`
- `Expect(Locator).toHaveURLAsync(url)`
- `Expect(Locator).toHaveValueAsync(value)`

Locators

CSS selectors

```
await page
  .Locator('div:nth-child(1) .card .items-actions .button')
  .clickAsync();
```



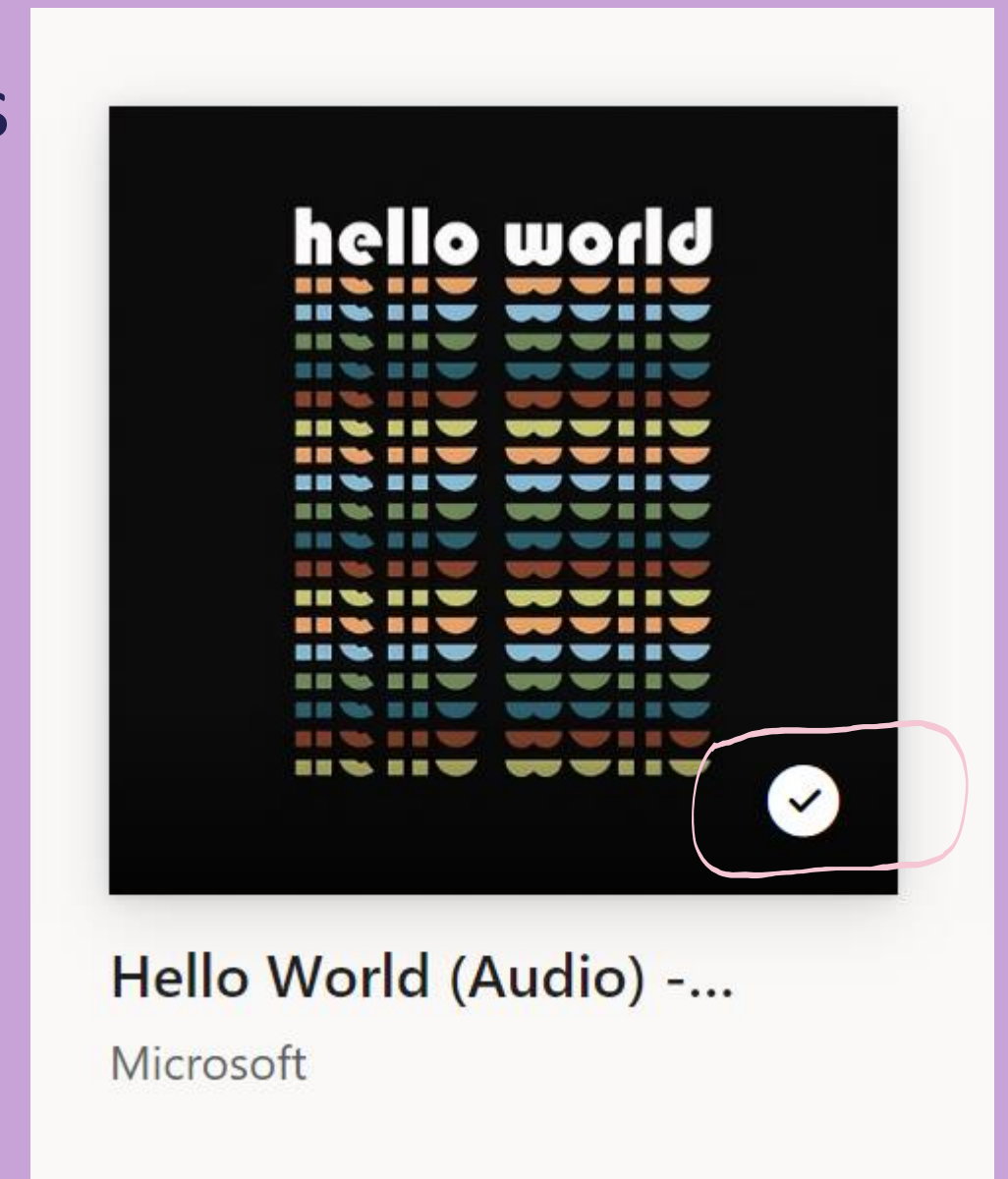
Hello World (Audio) -...


Microsoft

Locators + hasText

Case insensitive substring match,
accepts both strings and regexes

```
await page
  .Locator('.card', {
    hasText: 'Hello World'
  })
  .Locator('button')
  .clickAsync();
```



 **Discover** **Subscriptions** **Listen Later** **Listen Together** **Settings**[< Back](#)

Episodes



Episode 110: RELEASE CANDIDATE!

Apr 15, 2022

19 min

Show Notes

.NET MAUI RC is here. I don't need to say any more.
New releases...



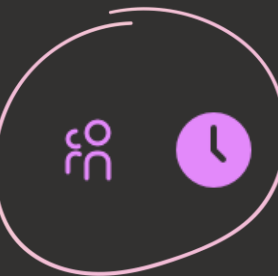
Episode 109: .NET MAUI (preview 14) is Released!

Mar 18, 2022

43 min

Show Notes

Another month, another step closer to the .NET MAUI GA! Join James, David and Matt to hear about the latest bits being added to .NET MAUI....



Episode 108: Buddy App

Mar 11, 2022

34 min

Another customer success story - this time a Xamarin app that helps those in the construction industry with mental health.
Follow Us:...



Locators + has:

```
await page
  .Locator('.row', {
    has: page.Locator('text=.NET MAUI')
  })
  .Locator('[aria-label="Listen Later"]')
  .clickAsync();
```

Episode 110: RELEASE CANDIDATE!

Apr 15, 2022

Show Notes

.NET MAUI RC is here. I don't need to say any more.

New releases...



Episode 109: .NET MAUI (preview 14) is Released!

Mar 18, 2022

Show Notes

Another month, another step closer to the .NET MAUI GA! Join James, David and Matt to hear about the latest bits being added to .NET MAUI....



Episode 108: Buddy App

Mar 11, 2022

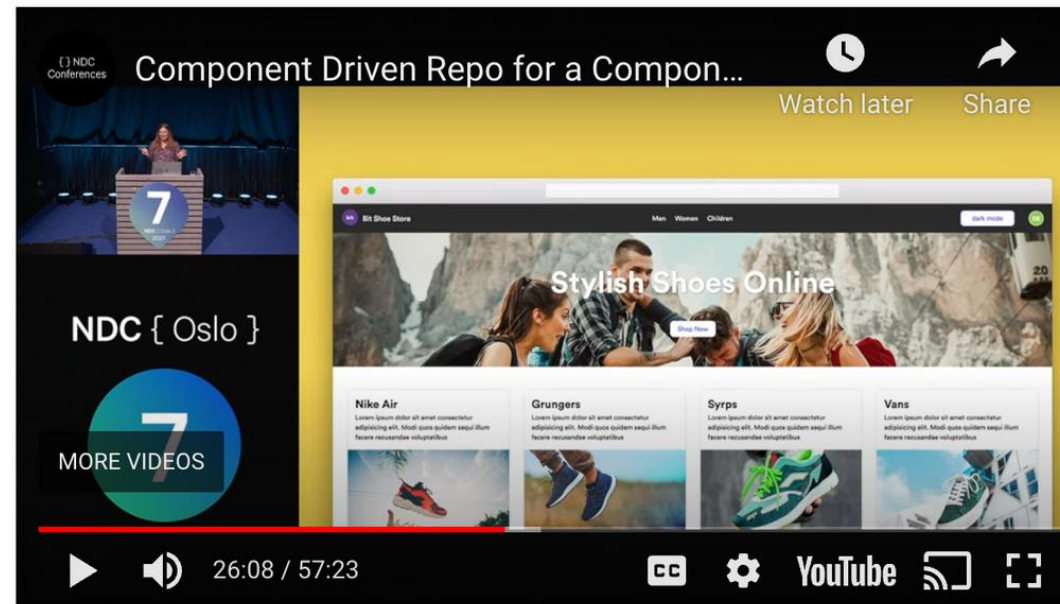
Another customer success story - this time a Xamarin app that helps those in the construction industry with mental health.

Follow Us:...



Frame Locators

Autowait iframes to appear



Elements Console Recorder Sources Network Performance Memory Application Security Lighthouse Media

▼<div class="layout_layout__23k9L shoe-store_layout__1Khu"> grid

▶<header class="header_header__2Nct">...</header> flex

▼<div class="layout_layout__23k9L"> grid

...▶<iframe width="560" height="315" src="https://www.youtube.com/embed/nQA48vH2CCc" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen style="margin: 0 auto;">...</iframe> == \$0

▶<div class="shoes-card-grid_grid__23-xe home_products__1zbph" style="

Styles Computed Layout Event Listeners >>

Filter :hov .cls + [▶]

element.style {

margin: ▶ 0 auto;

}

iframe[Attributes Style] {

width: 560px;

height: 315px;

padding -

- - 560x315 -

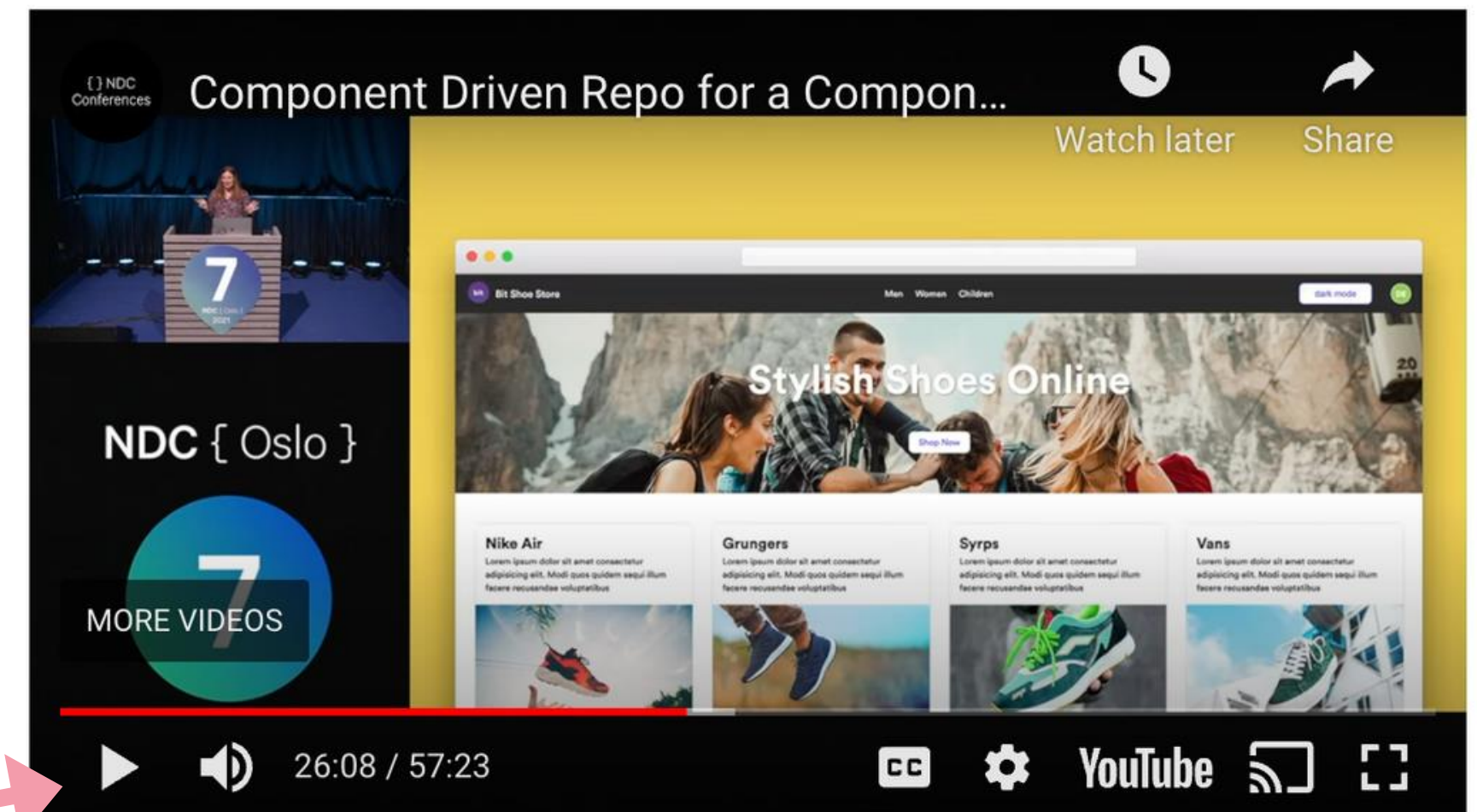
-

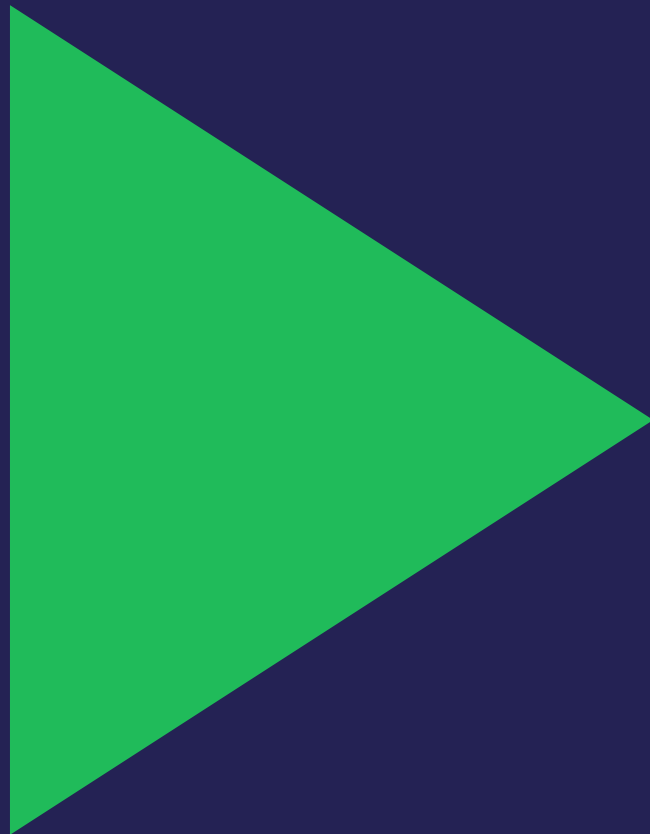
Frame Locators

Autowait iframes to appear

```
var pauseButton = page
  .FrameLocator('iframe')
  .Locator('[aria-label="Pause"]')

await pauseButton.clickAsync();
```





Running Tests


Run all tests, set of tests, single test

Tests run in parallel

Super fast....



Running Tests



```
dotnet test
```



Running Tests in headed mode

```

// Chromium, Firefox, or Webkit
await using var browser = await
playwright.Chromium.LaunchAsync(new
BrowserTypeLaunchOptions
{
    Headless = false,
    SlowMo = 100
});

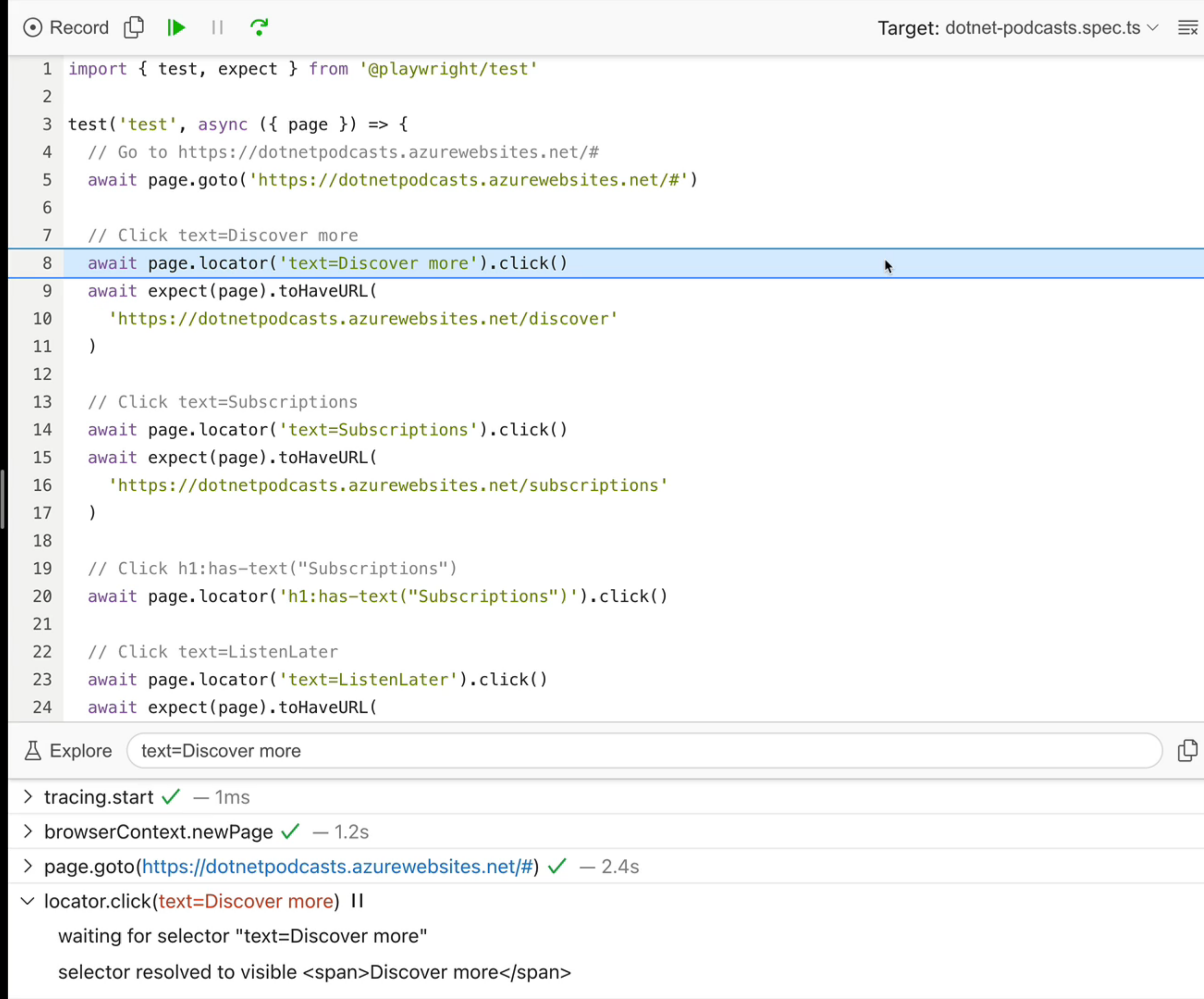
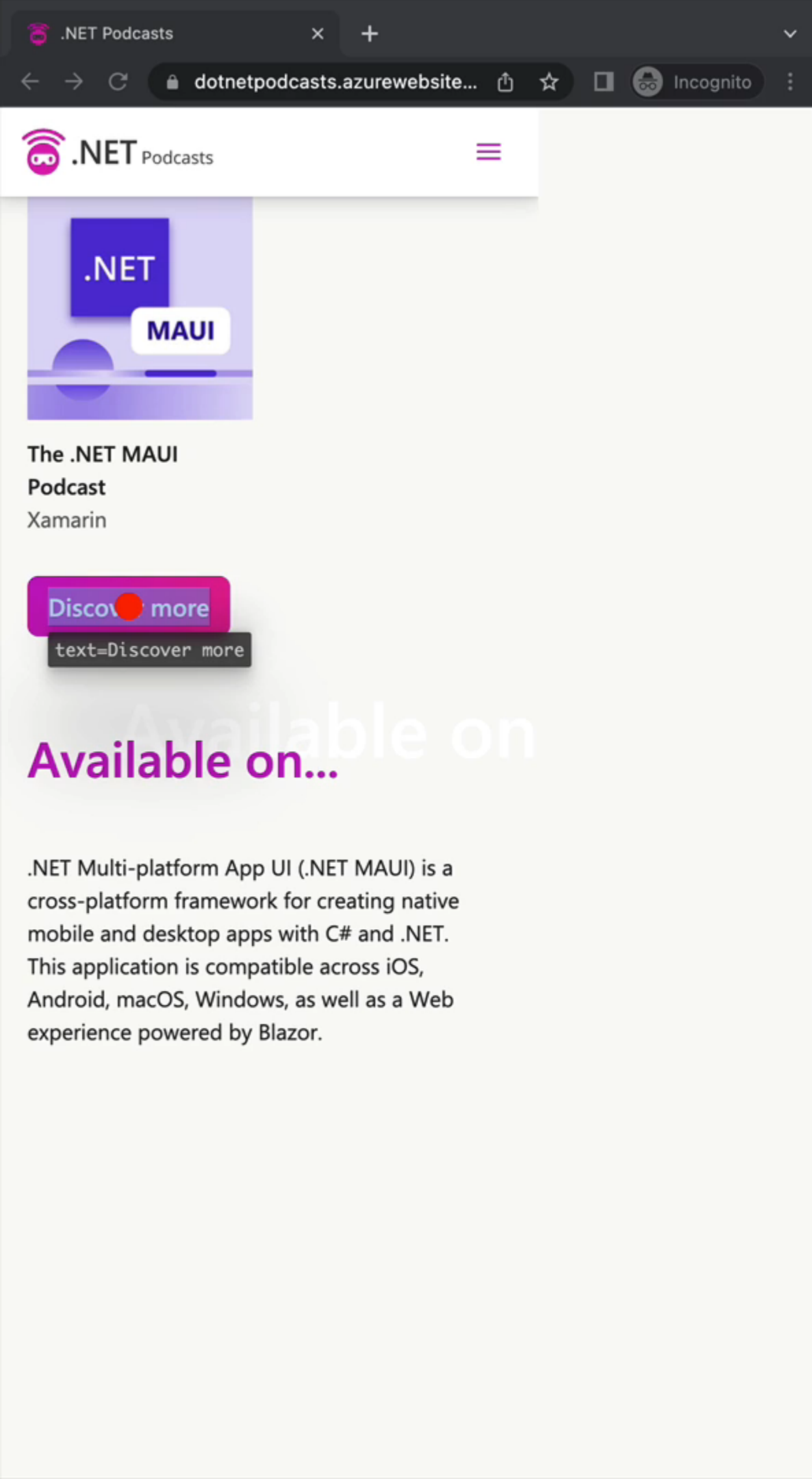
```





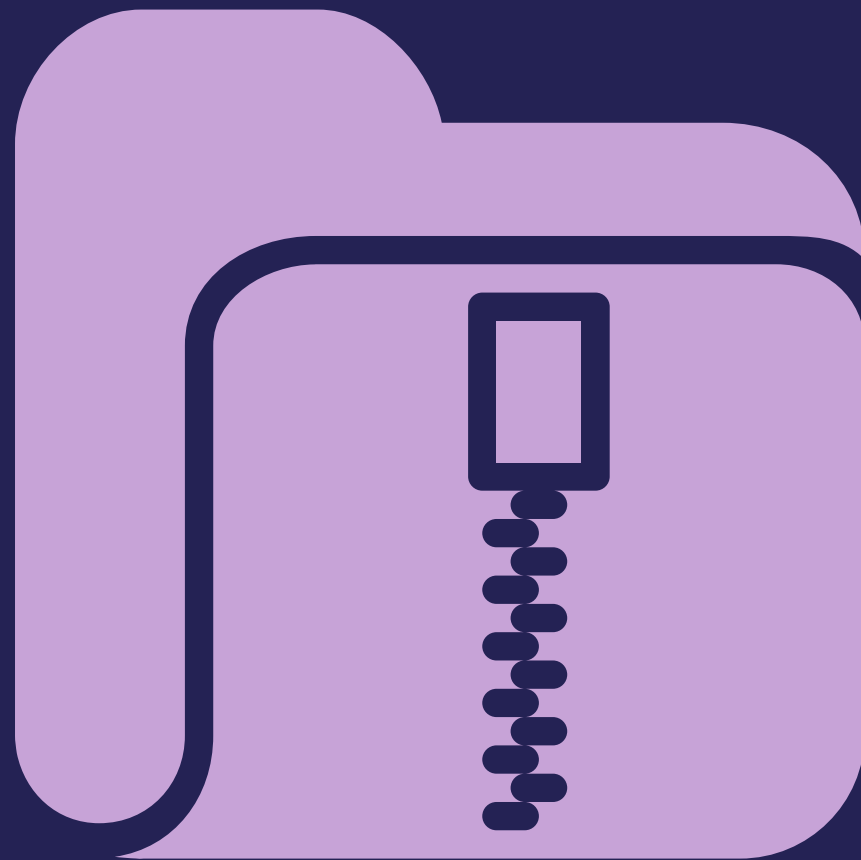
Running debug mode

```
...  
$Env:PWDEBUG=1 dotnet run  
set PWDEBUG=1 dotnet run
```



Trace Files

Record a trace for each test



```
await using var browser =
playwright.Chromium.LaunchAsync();
await using var context = await
browser.NewContextAsync();

// Start tracing before creating / navigating a page.

await context.Tracing.StartAsync(new
TracingStartOptions
{
    Screenshots = true,
    Snapshots = true,
    Sources = true
});

var page = context.NewPageAsync();
await page.GotoAsync("https://playwright.dev");

// Stop tracing and export it into a zip archive.
await context.Tracing.StopAsync(new
TracingStopOptions
{
    Path = "trace.zip"
});
```

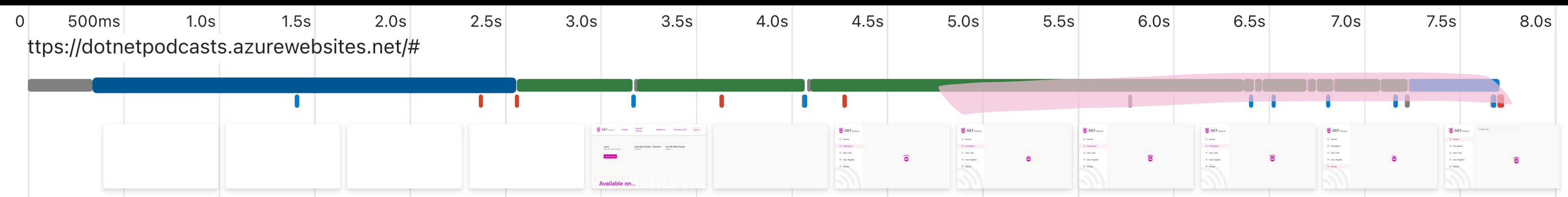

trace.playwright.dev

Drop Playwright Trace to load

or

Select file

Playwright Trace Viewer is a Progressive Web App, it does not send your trace anywhere, it opens it locally.



Actions Metadata

browserContext.newPage	339ms
page.goto https://dotnetpodcasts.azurewebsites.net/#	2.2s
locator.click text=Discover more	607ms
expect.toHaveURL :root	12ms
locator.click text=Subscription	879ms
expect.toHaveURL :root	14ms
locator.click h1:has-text("Subscription")	2.3s
locator.click text=ListenLater	55ms
expect.toHaveURL :root	5ms
locator.click text=Listen later	34ms
locator.click text=ListenTogether	230ms
expect.toHaveURL :root	5ms
locator.click text=I listen together	37ms

Action Before After

https://dotnetpodcasts.azurewebsites.net/#

The first podcast app powered by .NET MAUI & Blazor

Take your favorite tech podcasts with you wherever you are with .NET Podcasts. Native, cross-platform, and powered by .NET across Android, iOS, macOS, Windows, and Web.

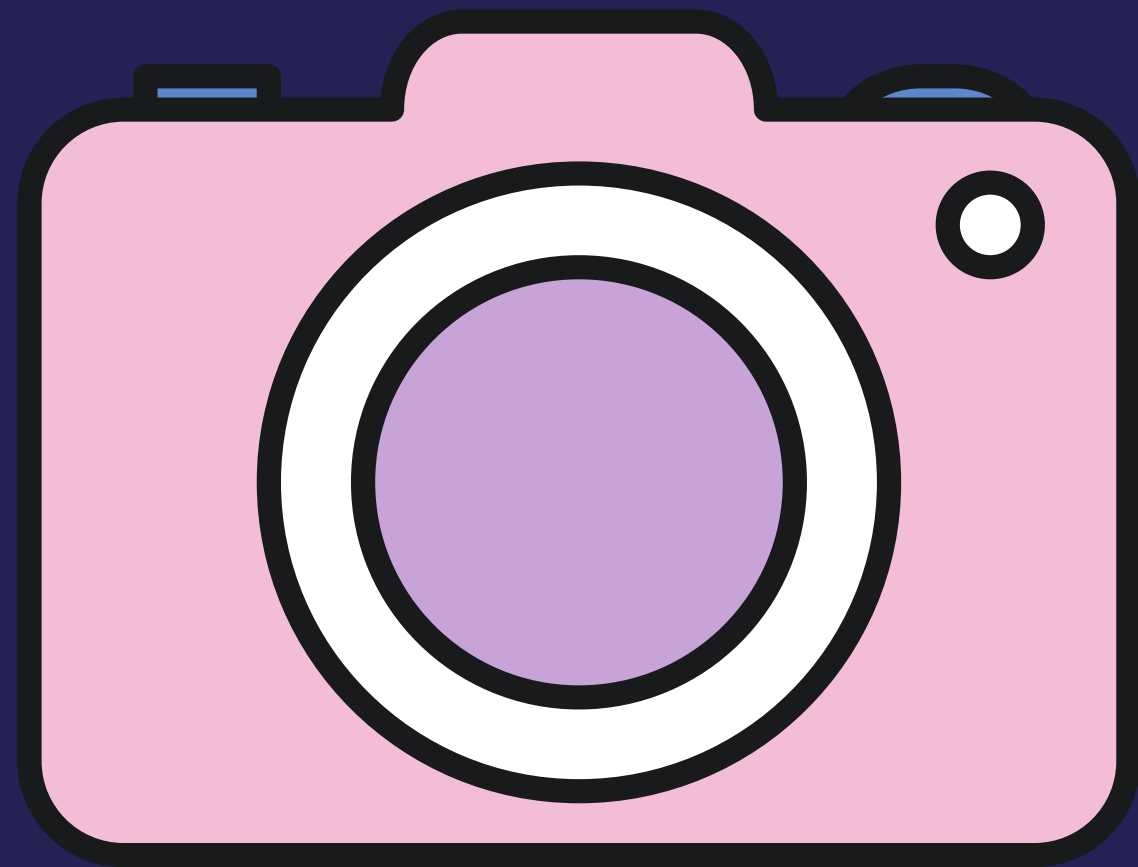
Download on the App Store GET IT ON Google Play Get it from Microsoft

Call Console Network³² Source

page.goto
TIME
wall time: 23/06/2022, 12:22:36
duration: 2.2s
PARAMETERS
url: "https://dotnetpodcasts.azurewebsites.net/#"
waitUntil: "load"
RETURN VALUE
response: <handle>
LOG
navigating to "https://dotnetpodcasts.azurewebsites.net/#"

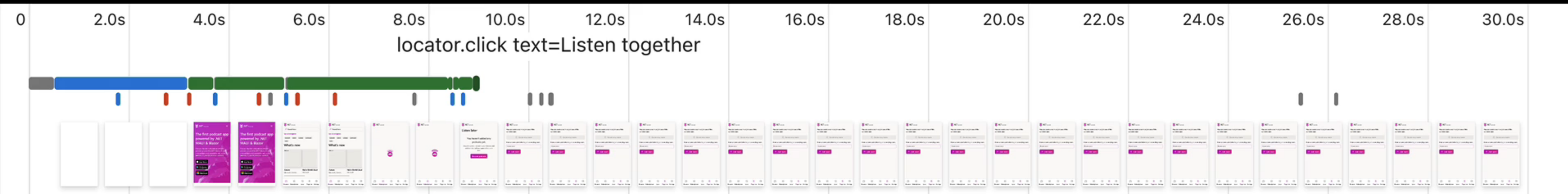
All 4Passed 2Failed 2Flaky 0Skipped 0

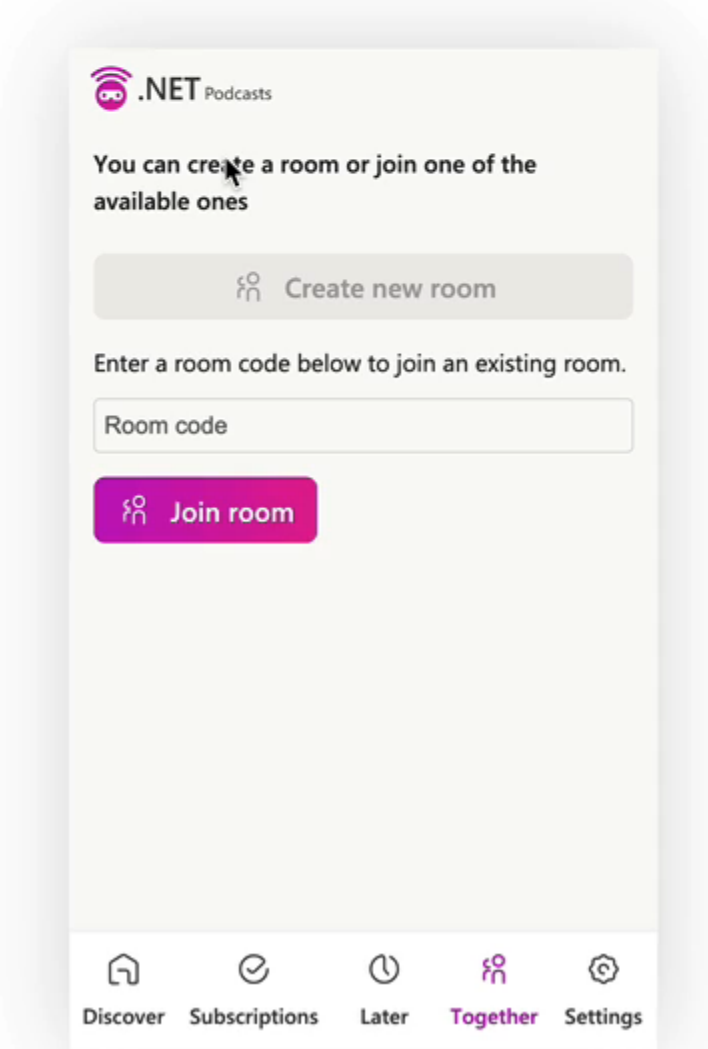
▼ dotnet-podcasts.spec.ts	1.4m
✗ test — dotnet-podcasts.spec.ts:3	Mobile Chrome 30.7s
✗ test — dotnet-podcasts.spec.ts:3	Mobile Safari 30.7s
✓ test — dotnet-podcasts.spec.ts:3	chromium 10.0s
✓ test — dotnet-podcasts.spec.ts:3	webkit 12.8s



DOM Snapshot

You can interact with the DOM,
inspect etc.



Actions	Metadata	Action	Before	After	Call	Console	Network	Source
browserContext.newPage	508ms	https://dotnetpodcasts.azurewebsites.net/listen-together			❗ Action was interrupted			
page.goto https://dotnetpodcasts.azurewebsites.net/listen-together	2.7s				locator.click			
locator.click text=Discover more	501ms				TIME			
expect.toHaveURL :root	17ms				wall time: 23/06/2022, 11:31:29			
locator.click text=Subscriptions	1.4s ❌ 2				duration: Timed Out			
expect.toHaveURL :root	33ms				PARAMETERS			
locator.click h1:has-text("Subscriptions")	3.2s				selector: "text=Listen together"			
locator.click text=ListenLater	91ms				strict: true			
expect.toHaveURL :root	8ms				LOG			
locator.click text=Listen later	102ms				waiting for selector "text=Listen together"			
locator.click text=ListenTogether	271ms				selector resolved to hidden <h1 tabindex=...			
expect.toHaveURL :root	20ms				attempting click action			
locator.click text=Listen to...	Timed Out ❗				waiting for element to be visible, enabled ...			
					element is not visible - waiting...			

Playwright Testing



Codegen: Auto Generate Tests

Run Debug mode

Locators and Frame Locators

Show TraceFiles



debs_obrien

Playwright Testing



Auto wait built in for all actions:

- no need for set timeout calls

Tests run in parallel:

- super fast
- multiple browsers and devices

Intercepting requests:

- follows all re-directs
- bypasses CORS
- manages cookies



Are you ready to Playwright?



Thank You

Docs: playwright.dev

aka.ms/playwright/youtube

aka.ms/playwright/twitter

aka.ms/playwright-slack