

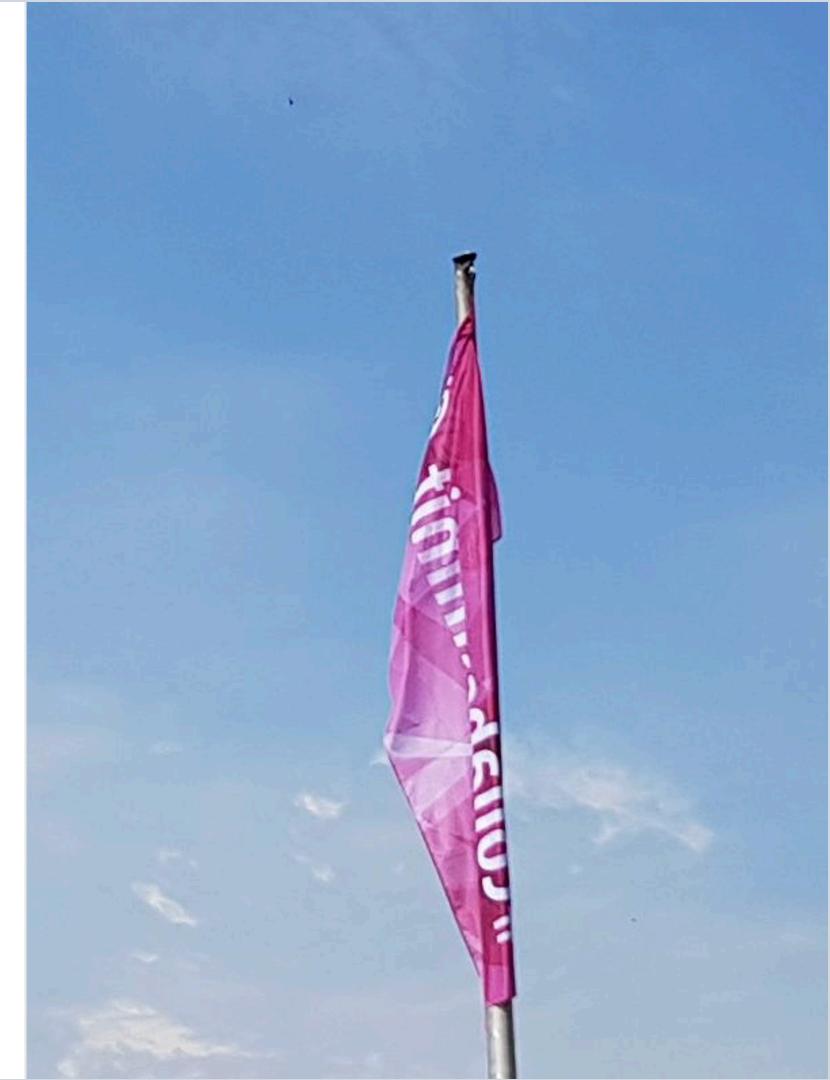


ECS 2025

Integrate Playwright tests for Power Apps into your DevOps process

Luise Freese | MVP

Elio Struyf | GDE | GitHub Star | MVP



Premium Partner**Premium Sponsor****Technology Partner****Diamond Sponsor**

ShareGate:

Platinum Sponsor**Gold Sponsor****Silver Sponsor****Bronze Sponsor****Startup Silver****Startup Bronze****Startup Sponsor**



Hello!

Elio Struyf

Struyf Consulting

#Stickerpreneur @ pyod.shop
Demo Time / Front Matter CMS

eliostruyf.com  & 

elio@struyfconsulting.be



Low code, but very real bugs



**IF YOU COULD TEST YOUR
CODE**

**THAT WOULD BE
GREAT**

Why do we need to test Power Apps?

WORKS
ON MY
MACHINE
〔ツ〕





Functionality, Accessibility, Performance, and Security

Functionality, Accessibility, Performance, and Security

T, R, U,
S, T,

**What options do we have for testing
this app?**

Feature / Tool	Manual Testing	Test Studio	Power Apps Test Engine (Preview)	Playwright
Effort to Run	High (repetitive, manual)	Medium (semi-automated)	Low (automated via CLI)	Low (automated via CLI)
Repeatability	Low	Medium	High	High
UI Coverage	Full (but inconsistent)	Limited to basic controls	Simulates control interactions	Full UI automation (DOM-based)
Test Authoring	N/A	Point-and-click	YAML + Power Fx (structured)	JavaScript/TypeScript scripts
Version Control Friendly	✗	✗	✓ (tests stored in source)	✓
Suitable for CI/CD	✗	✗ (no CLI support)	✓ (via Power Platform CLI)	✓ (easily with any pipeline)
Debuggability	Manual observation	Very limited logs	Logs, screenshots, trace files	Screenshots, logs, video, traces
Learning Curve	None	Low	Medium (Power Fx + YAML)	Medium (JS + browser concepts)
Cross-Platform/Browser	✓ (human)	✗	✓ (Chromium, Firefox, WebKit)	✓ (Chromium, Firefox, WebKit)
Production Readiness	✓	✓	✗ (Preview; not for production)	✓

Why Playwright?

Why Playwright?

- Ready for production (not just a preview)
- Uses popular programming languages (JavaScript/TypeScript)
- Widely adopted for web testing and automation
- Works on all major browsers and operating systems
- Gives you full control and easy debugging
- Simple to add to your CI/CD pipelines
- No vendor lock-in—use it anywhere
- Large, active community for support

Starting with Playwright

Using Playwright

Navigation

```
await page.goto('https://eliostruyf.com');
```

Get an element

```
const title = page.locator('header h2');
```

Test assertion

```
await expect(title).toHaveText('Elio  
Struyf');
```

SOLUTIONS ARCHITECT

Hi, I'm Elio Struyf — an Solutions Architect passionate about creating innovative solutions and sharing knowledge with the developer community. With expertise in Microsoft 365, Azure, End-to-End testing, and Developer Tooling, and Productivity. I help teams build better software and deliver exceptional results.

**Elio Struyf**

Solutions Architect & Developer Expert



Latest articles

May 25, 2025

Clean up old GitHub Actions artifacts with a script

Automatically clean up old GitHub Actions artifacts to stay within storage quotas and keep your CI/CD pipelines running smoothly.

[Read article →](#)

April 24, 2025

Monitor your GitHub Actions storage usage with a script

Effortlessly monitor your GitHub Actions storage usage with this script, ensuring you stay within quotas and avoid unexpected costs.

[Read article →](#)

April 23, 2025

Turn your presentations into playgrounds with Demo Time

Transform your presentations into interactive experiences with Demo Time, empowering your audience to learn and explore at their own pace.

[Read article →](#)

April 14, 2025

Reviewing the Shokz OpenRun Pro 2 for cyclists

Discover the Shokz OpenRun Pro 2 headphones for cyclists—enhanced sound, comfort, and safety for your rides. Perfect for music and awareness!

[Read article →](#)

VISITORS

797 / 1,663,368

Powered by visitorbadge.io

Struyf Consulting

Let's build together

Front Matter CMS ★ 2216

Manage content in VS Code

Demo Time ★ 88

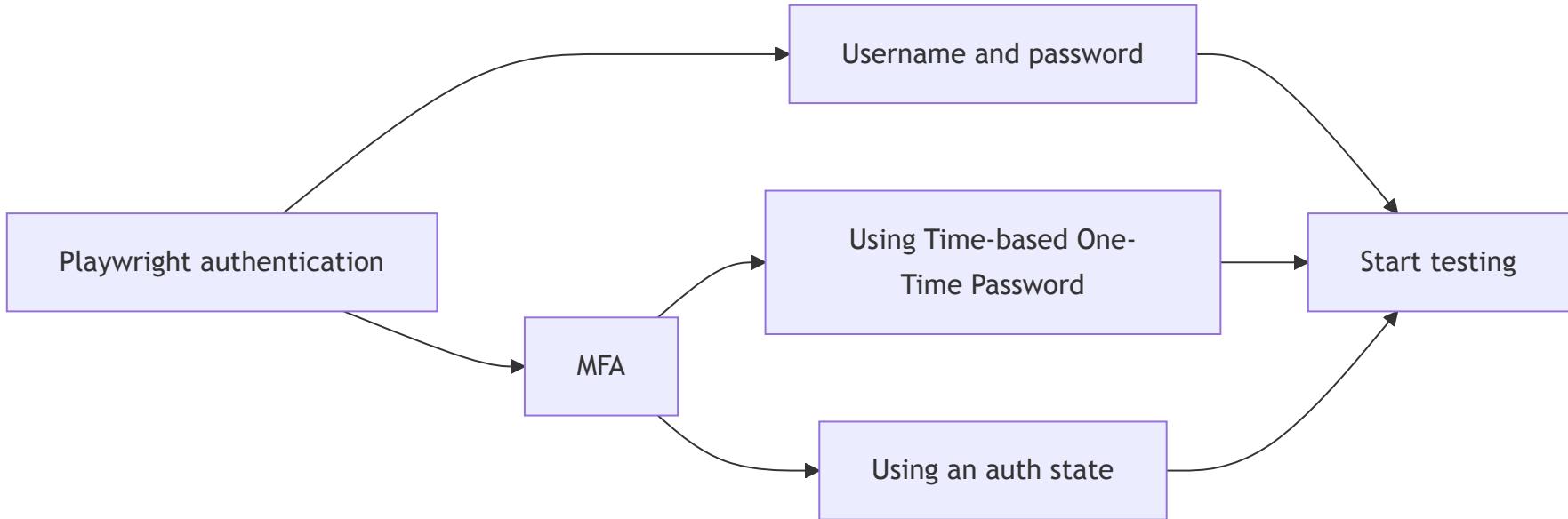
Present from VS Code

Support**Buy me a coffee**



AUTH

The options



Power Apps is a weird beast



It uses magic



Control testing

Text label

Text input

Button

```
<div class="appmagic-content-control-name appmagic-control-view canvasContentDiv _vst" data-control-id="7" data-control-name="TextInput1" style="position: absolute; top: 196px; left: 37px; width: 560px; height: 46px; z-index: 2;"></div>
<div class="appmagic-borderfill-container" style="background-color: rgb(255, 255, 255); border-style: solid; border-width: 2px; border-color: rgb(0, 18, 107); border-radius: 5px; margin: -1px;">
<div class="appmagic-border-inner" style="margin: -1px;">
<div class="react-knockout-control" spellcheck="false" style="position: absolute; inset: 0px;">
<div class="appmagic-textbox" touch-action="pan-x pan-y" appmagic-control="TextInput1"><flex>
<!-- ko if: (mode() === "multiline") -->
<!-- /ko -->
<!-- ko if: (mode() !== "multiline") -->
<!-- /ko -->
<input appmagic-control="TextInput1textbox" class="appmagic-text mousetrap block-undo-redo" data-bind="attr: {
    type: mode() === 'password' ? 'password' : 'text',
    title: properties.Tooltip() || null,
    'aria-label': properties.AccessibleLabel() || null,
    placeholder: properties.HintText,
    readonly: viewState.displayMode() === AppMagic.Constants.DisplayMode.View,
    'data-control-part': properties.Clear() ? 'text clearable' : 'text',
    inputmode: keyboardMode,
    maxlength: properties.MaxLength() < 0 ? 0 : properties.MaxLength()
},
css: {
    underline: properties.Underline,
    strikethrough: properties.Strikethrough,
    readonly: viewState.displayMode() === AppMagic.Constants.DisplayMode.View
},
value: text,
event: {
    click: handleClick,
    change: handleOnChange
},
style: {
    fontFamily: properties.Font,
    fontSize: properties.Size,
    color: autoProperties.Color,
    fontWeight: properties.FontWeight,
    fontStyle: properties.Italic,
    textAlign: properties.Align,
    lineHeight: properties.LineHeight,
    paddingTop: properties.PaddingTop,
    paddingRight: properties.PaddingRight,
    paddingBottom: properties.PaddingBottom,
    paddingLeft: properties.PaddingLeft
},
disable: viewState.displayMode() === AppMagic.Constants.DisplayMode.Disabled" type="text" placeholder data-control-part="text"
inputmode="text" tabindex="0" style="font-family: 'Open Sans', sans-serif; font-size: 21pt; color: rgb(0, 0, 0); font-weight: normal; font-style: normal; text-align: left; line-height: 1.2; padding: 5px 5px 5px 12px;">
<div class="appmagic-text-clear-container"></div></flex>
<!-- /ko -->
<!-- ko if: properties.Format() === 'number' && viewState.displayMode() === AppMagic.Constants.DisplayMode.Edit -->
<!-- /ko -->
</div>
</div>
</div>
</div>
```

And it keeps a dinosaur alive 

The canvas (read iframe)

```
▼<div id="playerShellContent" class="player-shell-content"> (flex)
  ▼<div id="playerAppContainer" class="player-app-container" role="main" aria-hidden="false"> (flex)
    ▼<iframe class="player-app-frame" width="100%" height="100%" tabindex="0" id="fullscreen-app-host" scrolling="no" frameborder="0" name="fullscreen-app-host" src="https://run
      time-app.powerplatform.com/publishedapp/e/default-b32d3268...%22%20%22staticContentUrl%22%3A%22https%3A%2F%2Fstatic.powerapps.com%22%7D" allow="geolocation; microphone; c
      amera; fullscreen; clipboard-write" referrerpolicy="strict-origin-when-cross-origin" style="outline: none;" title="application"> = $0
    ▼#document (https://run
      time-app.powerplatform.com/publishedapp/e/default-b32d3268-2794...%22%20%22staticContentUrl%22%3A%22https%3A%2F%2Fstatic.powerapps.com%22%7D)
        <!DOCTYPE html>
        ▼<html lang="en">
          ▶<head>@@</head>
          ▼<body class="browser notwindows">
            ▼<div id="publishedCanvasContainer" style="position: relative; overflow: auto; "> (scroll)
              ▶<style>@@</style>
              ▼<div id="responsiveScalingContainer" style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; z-index: 1; >
                ▼<div id="publishedCanvas" dir="ltr" style="width: 640px; height: 1136px; transform: translate(632.5px, 415px) scale(1.73063); ">
```

Playwright tests for Power Platform

Retrieving an element

```
// Get the iframe canvas
const iframe = page.frameLocator("iframe#fullscreen-app-host");
const publishedCanvas = iframe.locator("#publishedCanvas");

// Retrieve the control to test/use
const textInputControl = publishedCanvas.locator(`div[data-control-name='TextInput1']`);
const textInput = textInputControl.locator("input");

// Use the control
await textInput.fill("Hello World");
```

The simplified version with M365 Helpers

```
// Get the iframe canvas
const appFrame = await getAppFrame(page);

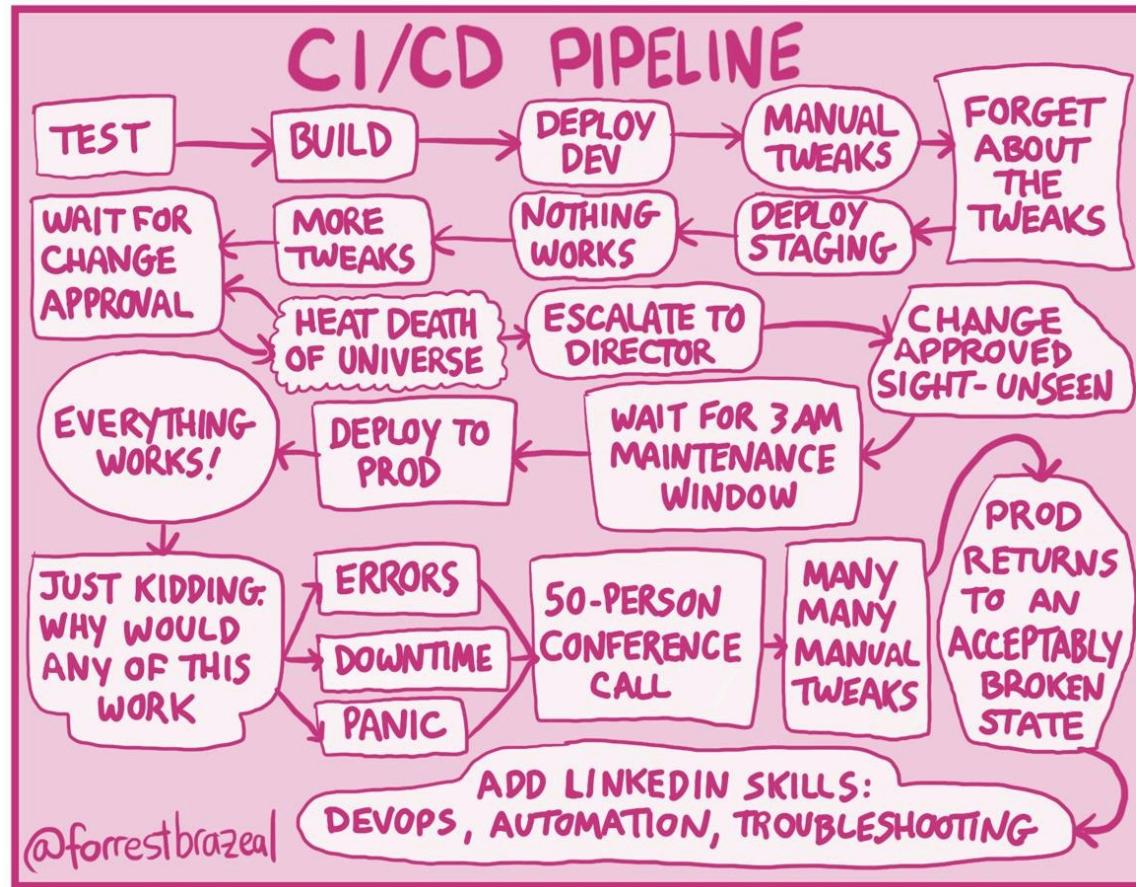
// Retrieve the control to test/use
const textInput = getInput(appFrame, "TextInput1");

// Use the control
await textInput.fill("Hello World");
```

More test ideas with Playwright

- Test your views, forms, and controls
- Test your business logic
- Test your data connections
- Test your security and permissions
- Test accessibility

ALM for Power Apps



© 2018 Forrest Brazeal. All rights reserved.

The options

- GitHub Actions
- Azure DevOps
- Power Platform pipelines

princess

The typical DevOps process



The typical DevOps process



What do you need?

- Microsoft Entra app registration
 - Authenticate our actions for CI/CD against Dataverse
- Power Platform Environments [Dev, (Build), Test, Prod]
- Power Platform application users
 - Connected to the Entra app registration (for every environment)
- CI/CD configuration
 - Power Platform CLI or actions/tasks
 - Service connections

Development

- Export from DEV environment
 - **Managed** and **unmanaged** solutions
- Unpack the solution to have the code
 - Process the .msapp file
 - Check-in the code

Test & Production

Import the **managed** solution

Last slide



THANK YOU, YOU ARE AWESOME ❤

PLEASE RATE THIS SESSION
IN THE MOBILE APP.

github.com/estruyf/powerapp-release-testing

linkedin.com/in/luisefreese
luisefreese.bsky.social 

struyfconsulting.be
elio@struyfconsulting.be
eliostruyf.com 



