



# Automated testing with Playwright

- a practical introduction

# Demo time!





# What do you need?

- VS Code
- Node.js
- Playwright VS Code extension
- Demo project from MS

# About Playwright

- About the tool
- Brief history



# Why Playwright for Power Platform?

- Microsoft-backed
- Handles complex SPAs
- Reliable ways to find elements
- Authentication support
- CI/CD integration



# Playwright vs. other tools



- Selenium (with EasyRepro)
- Cypress
- TestComplete (linked to Jira/Zephyr)





# When and how to use Playwright?

- Unit test/API test/UI tests
- Test business critical flows, not all
- Ensure that dev team catches problems before the end users



# Playwright structure

- Project structure
- Selectors vs. locators
- Assertions





# Some tips

- “Flaky” tests
- Pick Locator
- Record tests
- Set timeout

# Timeout



## Set test timeout in the config

```
playwright.config.ts

import { defineConfig } from '@playwright/test';

export default defineConfig({
  timeout: 120_000,
});
```

API reference: [testConfig.timeout](#).

## Set timeout for a single test

```
example.spec.ts

import { test, expect } from '@playwright/test';

test('slow test', async ({ page }) => {
  test.slow(); // Easy way to triple the default timeout
  // ...
});

test('very slow test', async ({ page }) => {
  test.setTimeout(120_000);
  // ...
});
```





# Some tips

- “Flaky” tests
- Pick Locator
- Record tests
- Set timeout
- @Tag to classify
- Test project in source control



# Great resources to get started

- MS Learn page:  
[Set up test automation with Playwright](#)
- Official Playwright docs  
[Getting started - VS Code | Playwright](#)
- Artem Bondar YouTube videos  
[Install Playwright in 7 Minutes \(Even If You're New to Testing\)](#)





Fredrik Neiderud

---

Power Platform / Dynamics 365 Freelancer  
Solution architect, Independent Advisor & MCT



# Thanks!

---

