Xebia | Xpirit

Automate your cloud application testing with Playwright

Randy Pagels
DevOps Architect | Xpirit USA

Microsoft Solutions Partner

GitHub Verified Partner

Let's connect

Randy-Pagels

@RandyPagels

PagelsR

https://xpirit.com

SCAN ME

https://qrco.de/beHAED

Let's connect!

Randy Pagels

DevOps Architect

rpagels@xpirit.com

Agenda

- Testing Challenges
- Testing Goals
- Playwright Overview
- Getting Started
- Test Generation
- Test Reporting
- Trace Viewer
- Time Travel Experience
- Continuous Integration Testing

Xebia | Xpirit

UI Testing Challenges

Consistent design across browsers! Dynamic content and maintaining performance!

- Testing is **hard**
- Testing takes **time to learn**
- Testing takes **time to build**
- Tests are **slow** – they take too long to run!
- Tests are **brittle** – they break whenever the app changes!
- Tests are **flaky** – they crash all the time!
- Tests **don't make sense** – they are complicated and unreadable!
- Tests require **changing context**

https://automationpanda.com/2023/07/28/modern-web-testing-with-playwright/

Modern Testing Goals

Major goals for modern web testing

- Make testing **easy**
- Make testing **fun**
- Focus on **fast feedback loops** rather than certain types of tests.
- Make test development as **painless** as possible.
- Choose test tooling that naturally **complements dev workflows**.

https://automationpanda.com/2023/07/28/modern-web-testing-with-playwright/

Xebia | Xpirit

Playwright Overview

AN INTRODUCTION TO PLAYWRIGHT

@playwrightweb
https://playwright.dev

READY TO START YOUR PLAYWRIGHT RUN ?

5 RESOURCES TO HELP YOU JUMPSTART YOUR TEST ADVENTURE

1 READ THE DOCS
- https://aka.ms/playwright
- https://aka.ms/playwright/quickstart
- Install Playwright
- Run your first Test
- Explore Tools & Libraries

2 EXPLORE THE API
https://aka.ms/playwright/api
JS, Java, Python, .NET
- Playwright Test Runner
- Test Reporter
- Testing + Experimental

3 VISIT THE REPO
https://aka.ms/playwright/github
- Playwright Framework source
- Examples + Release docs
- Discussions

4 FOLLOW @playwrightweb
- Framework release updates
- Content + event announcements
- Community interactions

5 CHECKOUT DEMO CODE
https://aka.ms/playwright/demo
- Test script examples
- Github Actions integration
- Test Reporting outputs

WHAT IS PLAYWRIGHT?

0 AN OPEN SOURCE TESTING FRAMEWORK
- ENABLES RELIABLE END-TO-END TESTING AND AUTOMATION
- FOR MODERN WEB APPLICATIONS

https://github.com/microsoft/playwright
- TOOLING
- AUTOMATION
- API

CHECK OUT THE SOURCE

WHY SHOULD I BE USING PLAYWRIGHT?

1. ONE API, MULTI-EVERYTHING
2. RESILIENT, NO FLAKY TESTS
3. LIMITLESS, NO TRADEOFFS
4. ISOLATED, FAST EXECUTION
5. TOOLING, CREATE-DEBUG-ANALYZE

LET'S TALK ABOUT EACH OF THESE!

1 ONE API / MULTI EVERYTHING

- ANY BROWSER ANY PLATFORM
  Chromium, WebKit, Firefox
  MacOS, Linux, Windows
  LOCAL, HEADLESS, HEADED
- CROSS LANGUAGE
  JavaScript, TypeScript, Python, Java, .NET — libraries.
- TEST MOBILE WEB
  Native emulation
  - Chrome on Android
  - Mobile Safari on iOS
  - Same Rendering Engine on Cloud and Desktop.

PLAYWRIGHT – the Grey –
CHROMIUM WEBKIT FIREFOX
ONE API TO RULE THEM ALL
CROSS BROWSER TESTING
TEST AUTOMATION AT CLOUD SCALE

2 RESILIENT! NO FLAKY TESTS...

TESTING ENTRY
ALMOST READY FOR YOU .. HANG ON .. LET ME CHECK AGAIN ...
ARE YOU EXPECT-ing US?
TEST ACTIONS
REDUCE TEST COMPLEXITY INCREASE READABILITY + RELIABILITY
WEB PAGE

1 AUTO-WAIT
No more artificial timeouts to manage
- waits for elements to be ACTIONABLE before execution of test steps
- + RICH introspection events available!

2 WEB-FIRST ASSERTIONS
tailored for the dynamic web!
ASYNC AWAIT EXPECT
- convenience async matchers for assertions
- separate timeout (from test) for web first assert with default = 5 secs

3 TRACING
configurable strategy
retry? record traces? capture screenshot?
- checks conditions till met

3 NO TRADE-OFFS / LIMITS

TESTING API
TEST RUNNER API
TEST CONFIG OPTIONS

1 ALIGNED WITH MODERN WEB ARCHITECTURES
- RUN TESTS OUT OF PROCESS .. FREED FROM IN-PROCESS TEST RUNNER CONSTRAINT

2 MULTIPLE EVERYTHING
MULTIPLE ORIGINS
MULTIPLE TABS
MULTIPLE USERS
TEST ALL THE THINGS
ONE TEST MANY CONTEXTS
WRITE MORE COMPREHENSIVE TEST ACTIONS & SCRIPTS!
MORE REALISTIC USER ACTION FLOWS

3 TRUSTED EVENTS
I'M IN YOUR BROWSER – ACTING LIKE ANY USER!
- TEST HOVER ELEMENTS
- INTERACT WITH DYNAMIC CONTROLS
- USE REAL BROWSER INPUT PIPELINES

INDISTINGUISHABLE FROM REAL USERS!

4 TEST FRAMES
PIERCE SHADOW DOM WITH PLAYWRIGHT SELECTORS!

4 FULL ISOLATION / FAST EXECUTION

1 BROWSER CONTEXT
TEST SCRIPT
LIKE HAVING A BRAND NEW BROWSER PROFILE EACH TIME
- CREATES A BROWSER CONTEXT PER TEST IN JUST ms.
  - FULL TEST ISOLATION
  - ZERO OVERHEADS
  - LESS COMPLEXITY FOR TESTER

2 LOG IN ONCE!
TEST SCRIPT
1 STORE AUTH STATE
2 LOAD AUTH STATE (from stored location)
TEST NEW THING
- SAVE AUTHENTICATION STATE (tokens, cookies) FROM FIRST RUN – & REUSE IN ALL SUBSEQUENT TESTS
- BYPASS REPETITIVE LOGIN OPERATIONS (many third party sites don't want automated logins)
- STILL FULL ISOLATION FOR INDEPENDENT TESTS

5 POWERFUL TOOLING!

1 CODEGEN
INTERACTIVE SESSION
AUTO CAPTURED
- GENERATE TESTS BY RECORDING USER ACTION (FLOW)
- SAVE IN ANY SUPPORTED LANG. (Java, JS/TS, Python etc.)

2 INSPECTOR
TEST – DEBUG
INSPECTOR
- INSPECT PAGE, LOGS
- GENERATE SELECTOR
- STEP THROUGH EXEC
- SEE CLICK POINTS

3 TRACE VIEWER
- CAPTURE TEST PROGRESS (STATE, FAILURES, TIMING)
- SCREENCASTS, SNAPSHOTS, ACTION EXPLORER, TEST SOURCE (POST-MORTEM ANALYSIS)

ILLUSTRATED BY @SKETCHTHEDOCS

Playwright Overview

Open-source web test automation library on Node.js, which makes test automation easier for browsers based on Chromium, Firefox, and Webkit through a single API.

- ✓ **Cross-Browser**
  - Chromium, Firefox, and WebKit
- ✓ **Cross-language**
  - JavaScript, Python, and C#
- ✓ **Cross-platform**
  - Windows, Linux, and macOS
- ✓ Modern Asynchronous API
  - async/await
- ✓ Rich Element Interaction
  - wide range of built-in functions
- ✓ **Screenshots and Video Recording**
  - visual evidence of issues
- ✓ **Full Isolation – Fast Execution**
  - separate browser contexts

Reliable end-to-end testing for modern web apps

…by Microsoft

…and the people who made Puppeteer

https://playwright.dev/

https://github.com/microsoft/playwright

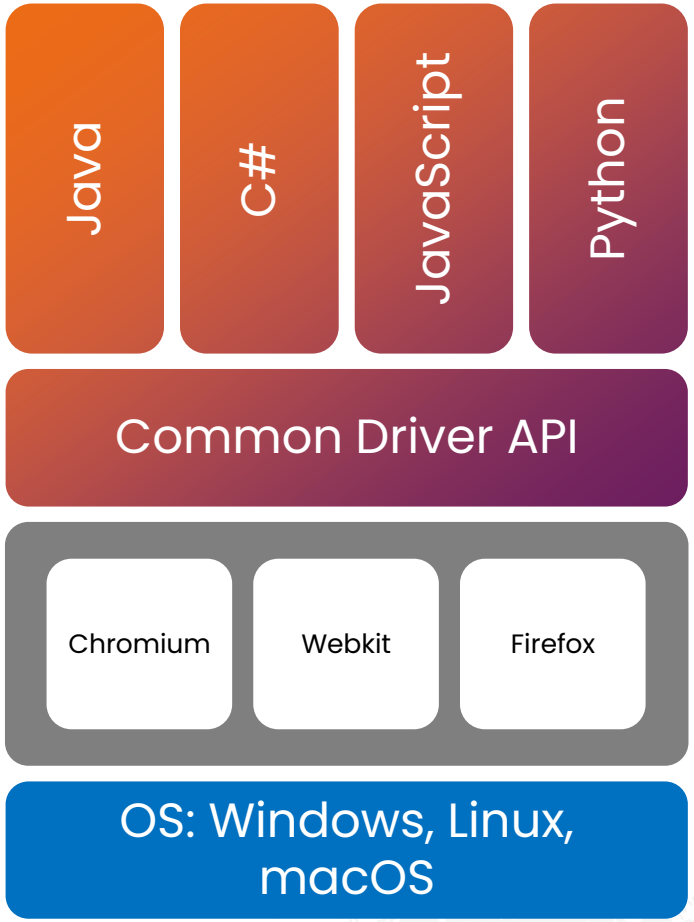
Playwright Architecture

- Language Bindings

- Single Automation Protocol
- Unify all debugging protocols

- (Native) browser debugging protocols,

e.g. ChromeDevTools Protocol, (Safari) Web Inspector



Getting Started

Setting up using VS Code

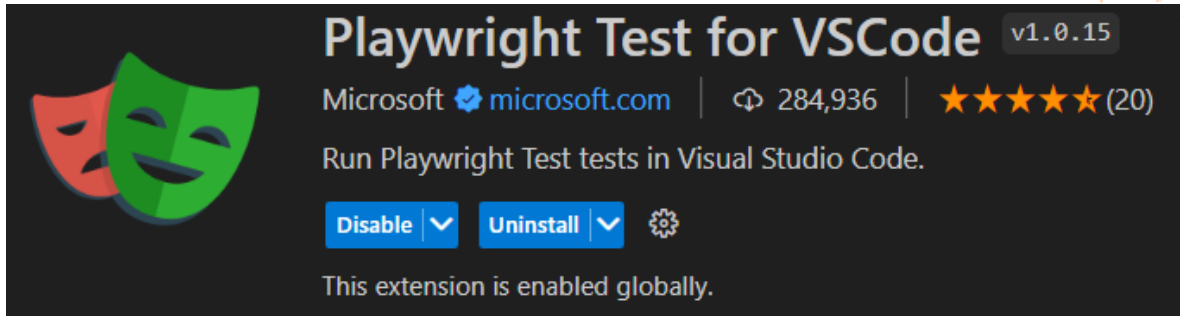
Install Extension

Install Playwright
- npx init playwright@latest

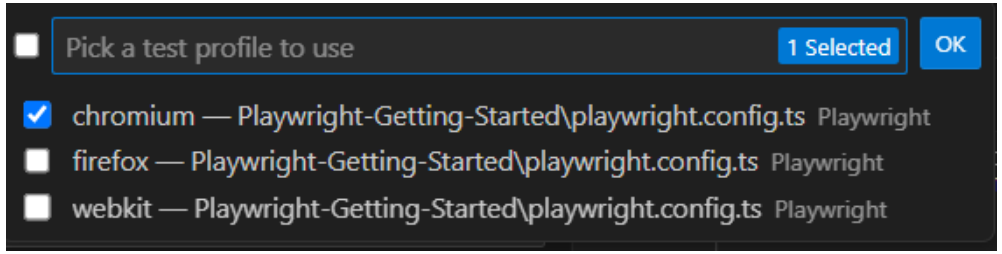
Run Tests
- npx playwright test
- npx playwright test --ui

Test Reports
- npx playwright show-report

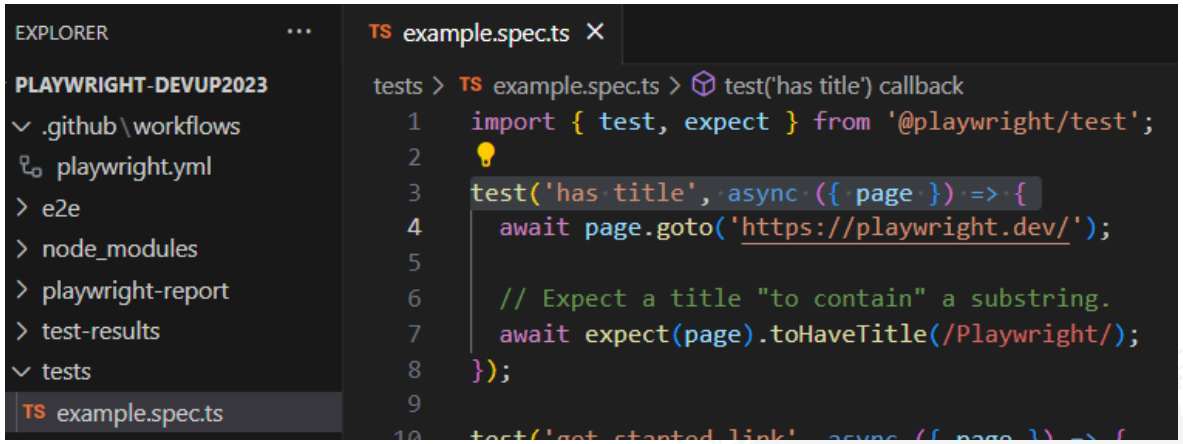
1



2



3



Test Generator

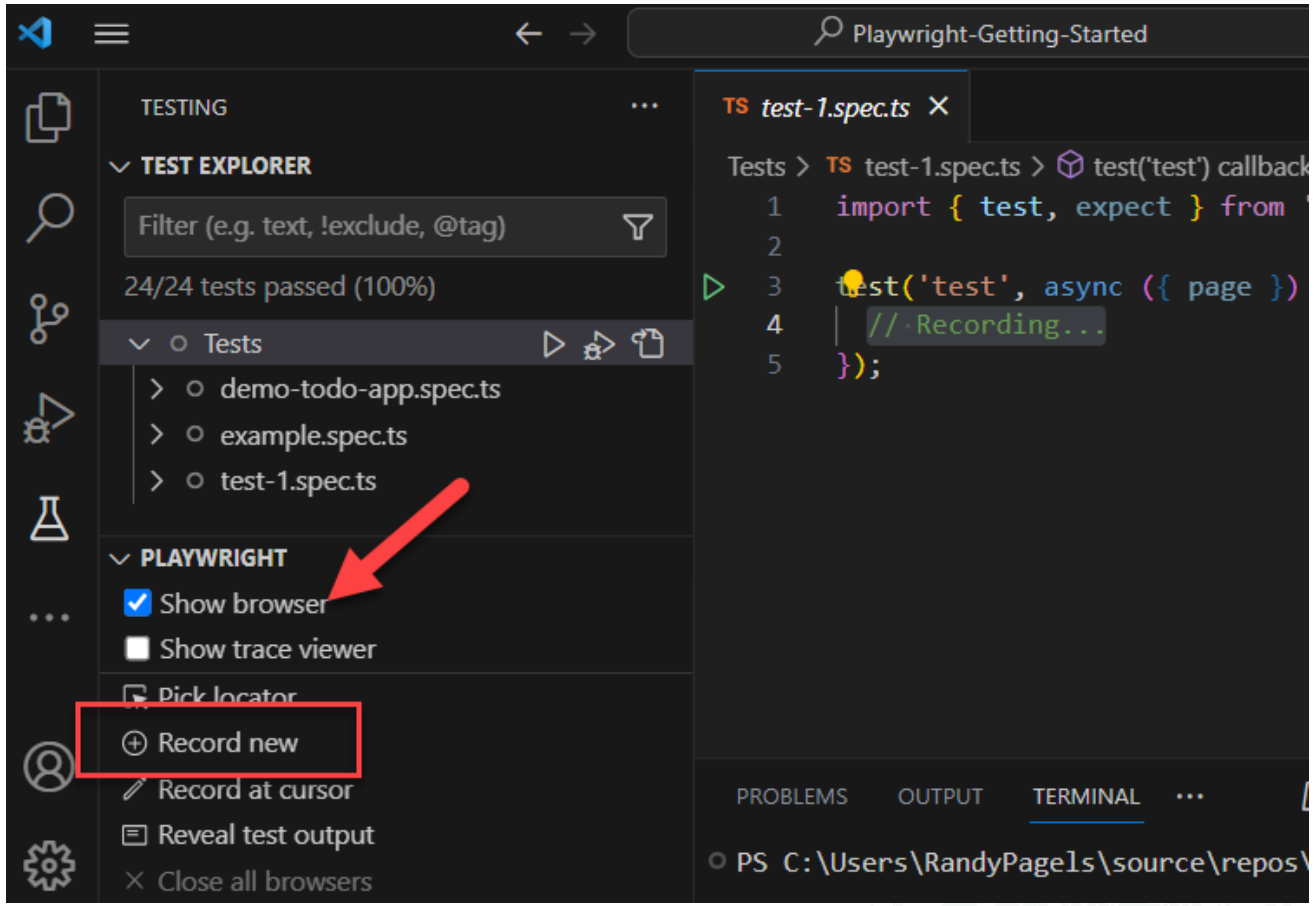
Playwright Test Generator is a GUI tool that helps you explore recorded Playwright traces after the script has ran.

CLI

- npx playwright codegen

VS Code

- Use Plugin, “Record Now”



Xebia | Xpirit

Trace Viewer

Playwright Trace Viewer is a GUI tool that helps you explore recorded Playwright traces after the script has ran.

CLI

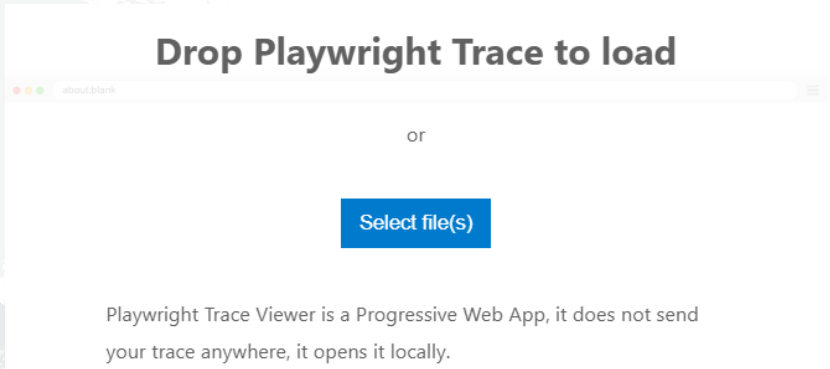
- npx playwright test test.ts --trace on

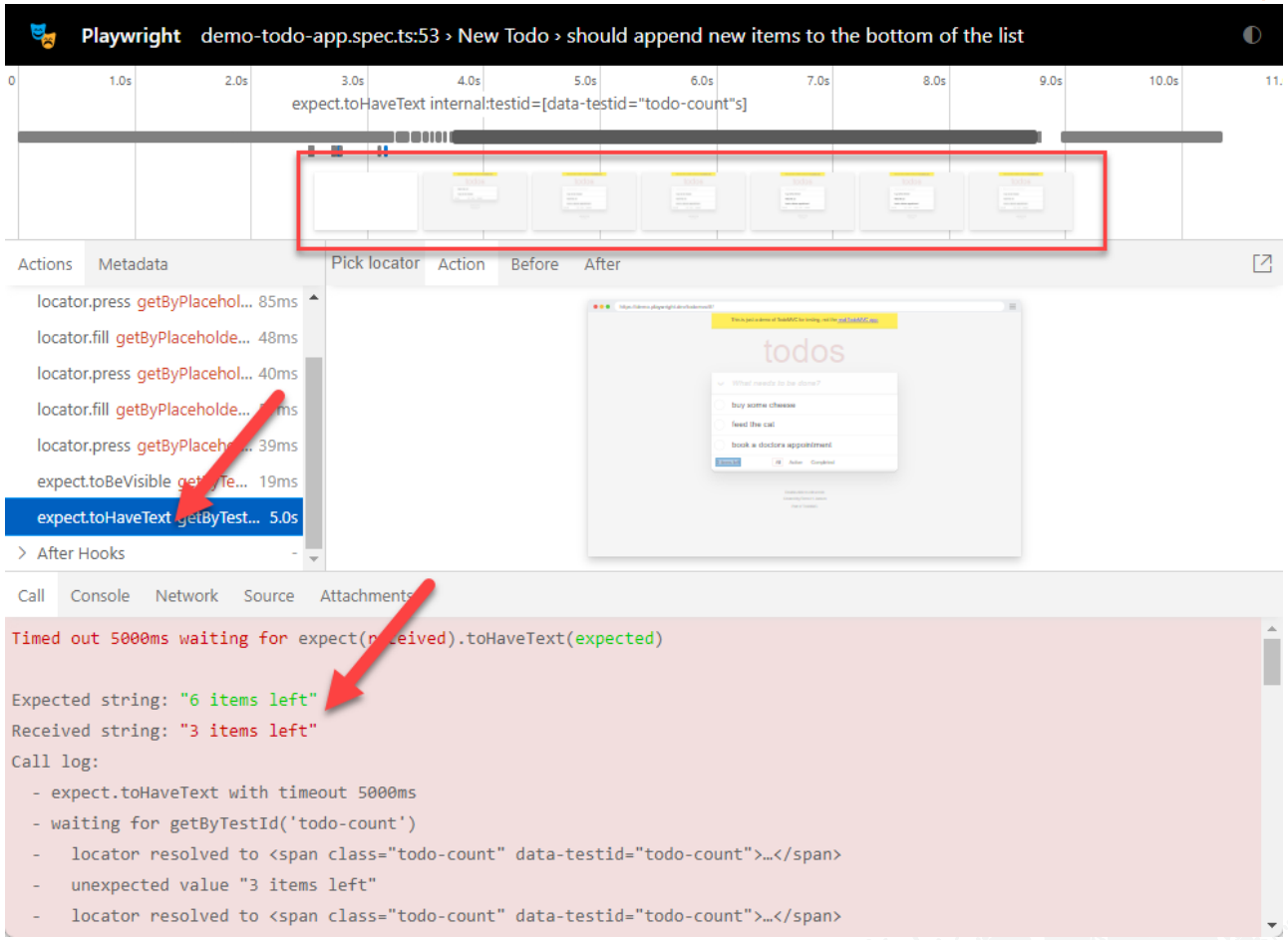
VS Code

- Use Plugin

Hosted Trace File Viewer

- https://trace.playwright.dev





Xebia | Xpirit

Report List

Playwright Test comes with a few built-in reporter.

HTML Report

- npx playwright test --reporter=**html**

List Report

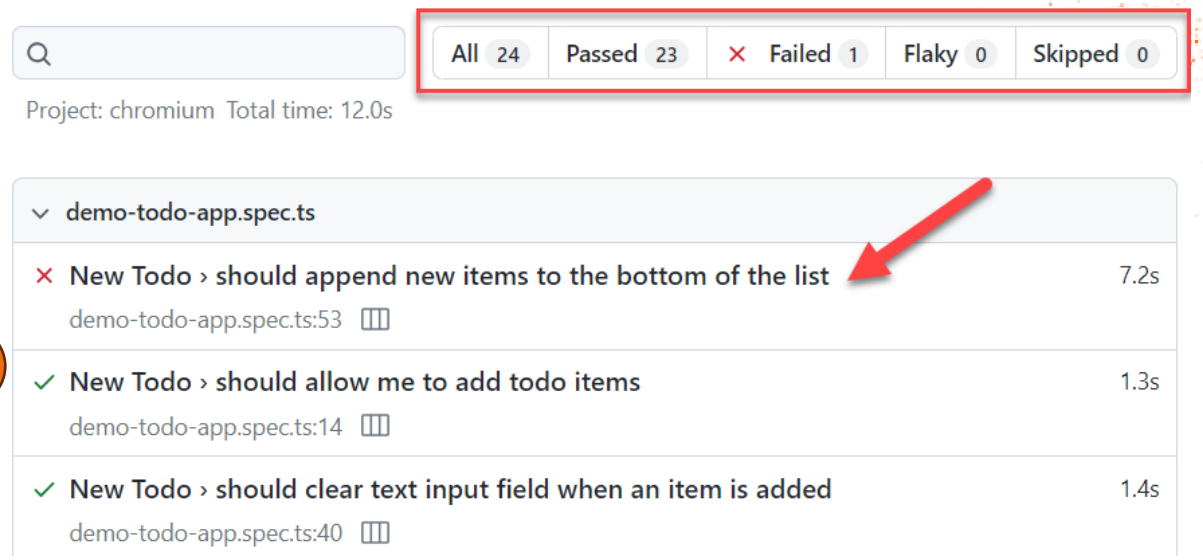
- npx playwright test --reporter=**list**

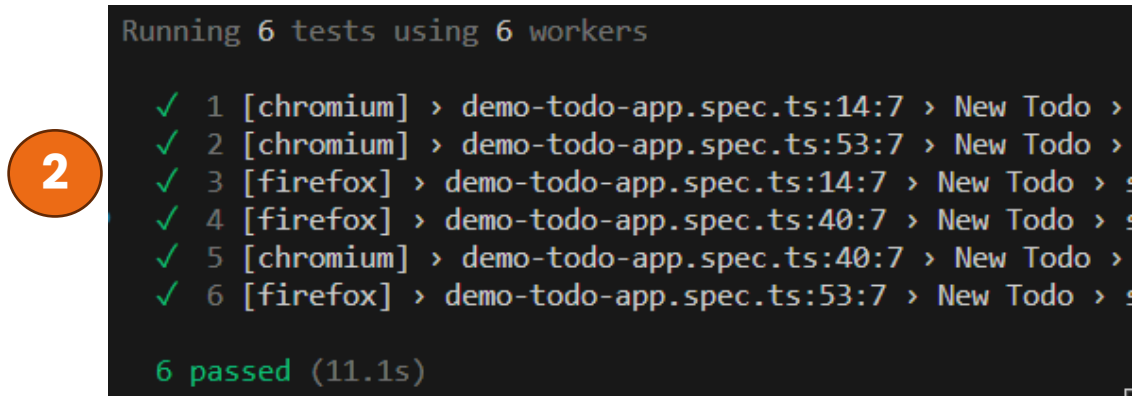
Line Report

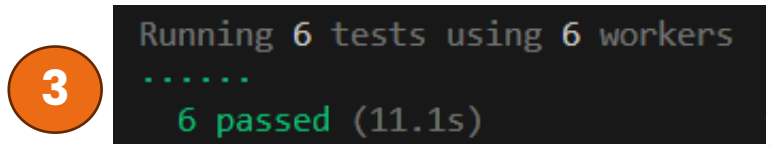
- npx playwright test --reporter=**line**

Dot Report

- npx playwright test --reporter=**dot**







Xebia | Xpirit

Playwright UI Mode

Explore, run and debug tests with a time travel experience complete with watch mode

CLI

npx playwright test testname.ts --ui

DOM Inspection

- page structure, inspect elements, view attributes.

Console Logs

- JavaScript-generated logs, errors, and warnings.

Network Monitoring

- Requests, timings, inspect headers, payloads.

Interactivity

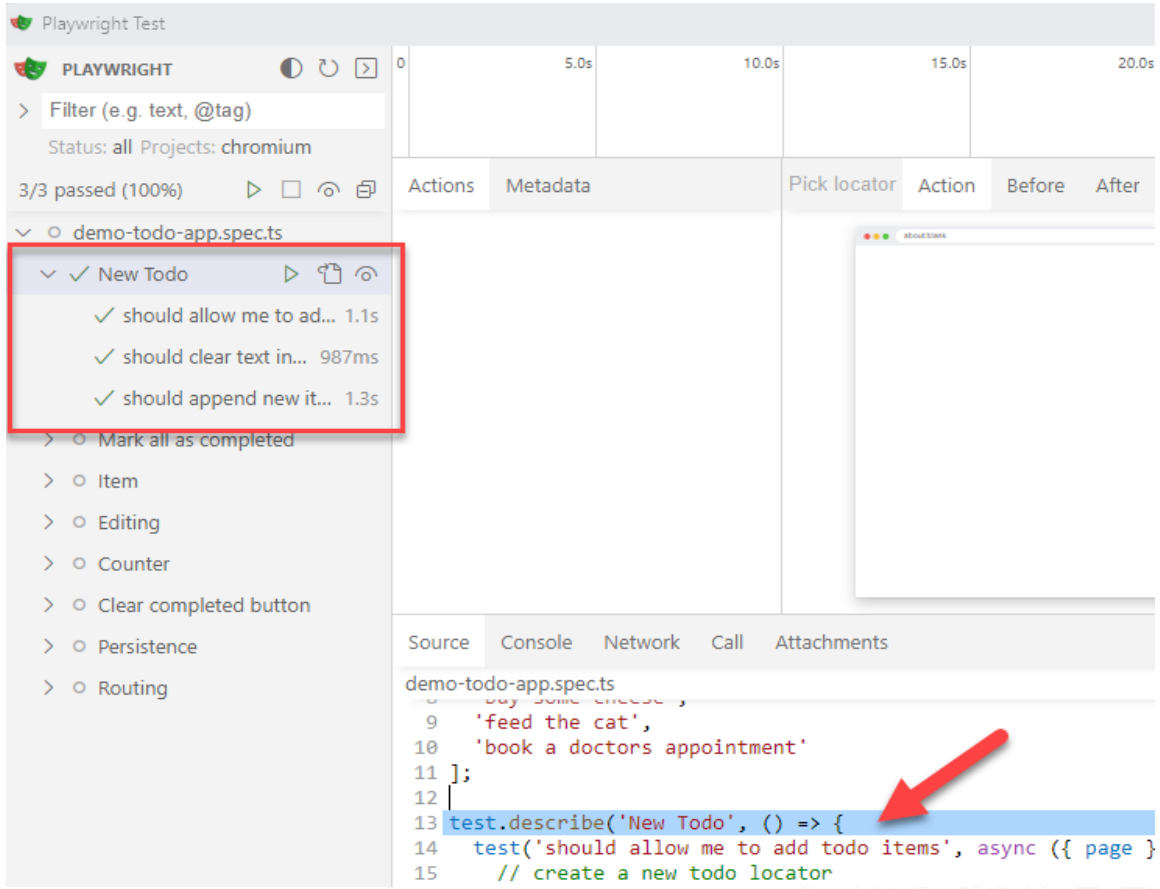
- Trigger events, clicks, real-time page responses.

Element Highlighting

- Hover over elements to highlight them.

Source Code Access

- View HTML, CSS, JS source for debugging.



Xebia | Xpirit

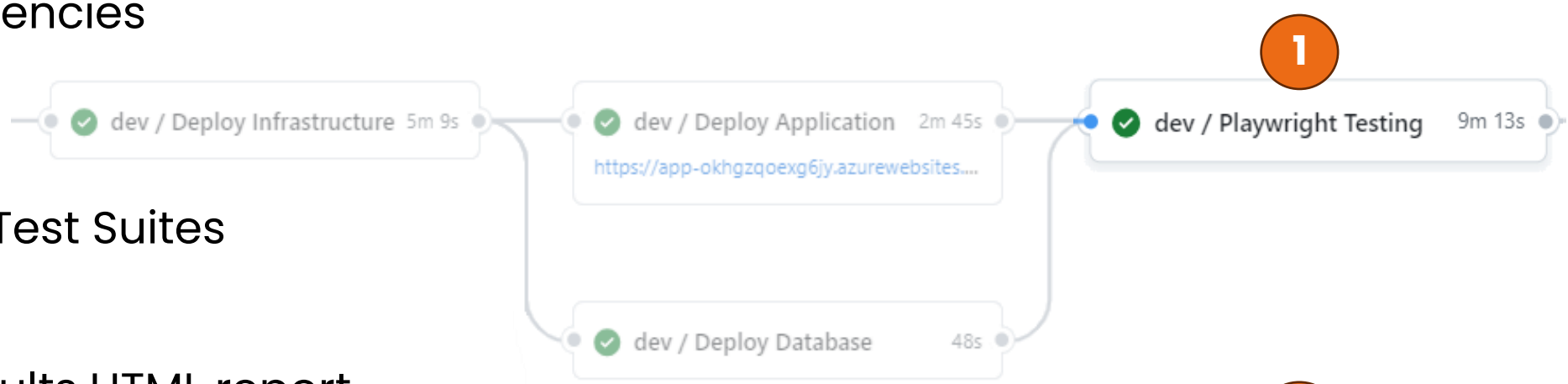
Xebia | Xpirit

Continuous Integration Testing

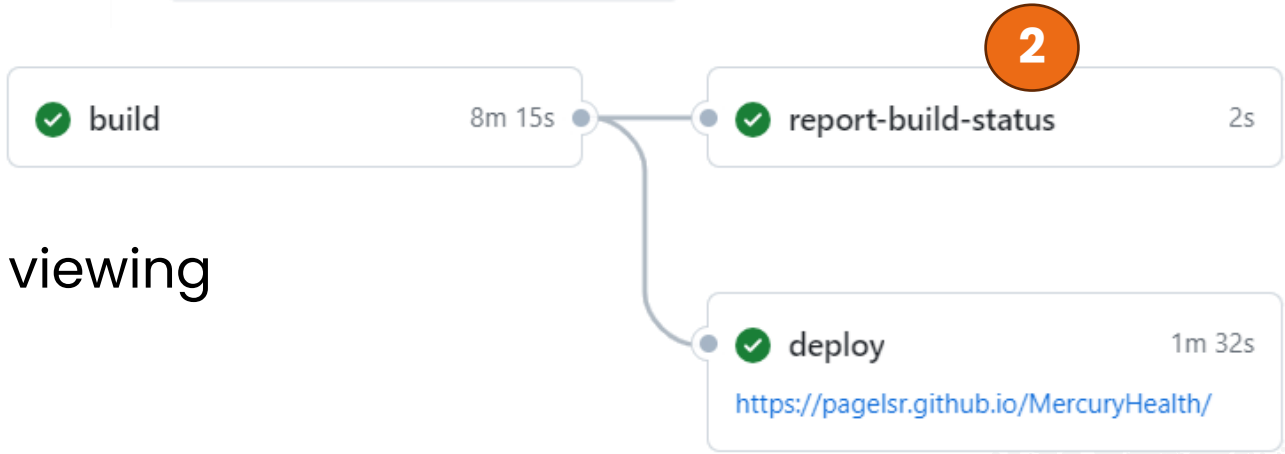
Continuous Testing

Tests run on each push and pull request into the main/master branch

- Installs all dependencies
- Installs Playwright
- Installs Browsers
- Run all the tests
- Parallelization for Test Suites



- Generate Test Results HTML report
- Generate Screenshots
- Generate Video Recordings
- Generate Debug Trace file
- Deploy static files to GitHub Pages for viewing



Xebia | Xpirit

Demos

Xebia | Xpirit

Azure AD Authentication

- Store the credentials of test accounts using environment variables.

```
test(`example test`, async ({ page }) => {
  // ...
  await page.getByLabel('User Name').fill(process.env.USERNAME);
  await page.getByLabel('Password').fill(process.env.PASSWORD);
});
```

- Set the values of these variables to use your CI system's secret management or Azure Key Vault.

```
testE2E:
  name: Run Playwright E2E tests
  runs-on: ubuntu-latest
  container: mcr.microsoft.com/playwright:v1.27.0-jammy
  env:
    USERNAME: ${{secrets.USERNAME}}
    PASSWORD: ${{secrets.PASSWORD}}
```

- Local Development? Use .env files and add to .gitignore.

```
# .env file
STAGING=0
USERNAME=me
PASSWORD=secret
```

Note: MFA cannot be fully automated and requires manual intervention.

Tool Comparison

Playwright is not the only browser automation tool out there

Selenium	Cypress	Playwright
Popular among testers	Popular among developers	So hot right now
C#, Java, JavaScript, Ruby, Python	JavaScript only	C#, Java, JavaScript, Python
All major full browsers	Chrome, Edge, Firefox, Electron	Chromium+, Firefox, WebKit
WebDriver protocol	In-browser JavaScript	Debug protocols
Just a tool – requires a framework	Full, modern framework	Full, modern framework
Can be slow/flaky if waiting is poor	Visual execution but can be slow	Typically the fastest execution
Open source, standards, & gov	Open source from Cypress	Open source from Microsoft

https://automationpanda.com/2023/07/28/modern-web-testing-with-playwright/

Microsoft Playwright Testing

Join the waitlist for Microsoft Playwright Testing private preview

- New Azure service for running Playwright tests.
- Cloud enabled to run Playwright tests at scale.
- High parallelization across operating system-browser combinations.
- Get tests done much faster.
- Speed up delivery of features without sacrificing quality.

Private Preview

To learn more about Microsoft Playwright Testing, refer to **https://aka.ms/mpt/private-preview-blog**.

Xebia | Xpirit

Thank you!