Playwright

@WebKit Con 2023

State of Modern E2E Testing

Agenda

- Playwright
- Trends
- Lessons learned
- Internals
- Challenges
- Resources
- Q&A

Playwright

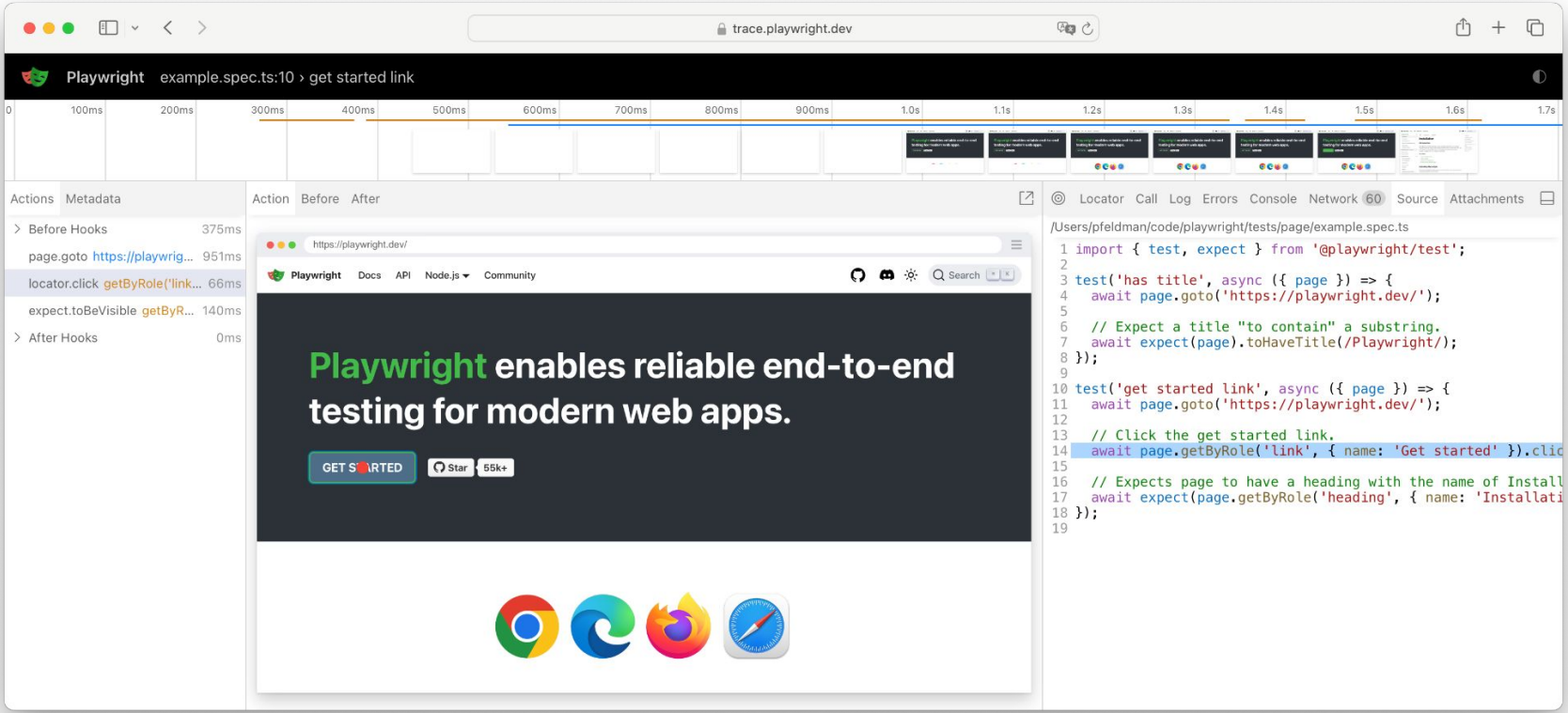
Playwright: **History**

- 2009 - 2013: WebKit Web Inspector
- 2013 - 2019: Chrome DevTools
- 2011: CDP - chrome remote protocol
- 2015: Node.js debugging tools
- 2017: Puppeteer - automation
- 2020: Playwright - cross-browser testing

Playwright: **Test**

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

test('has title', async ({ page }) => {
  await page.goto('https://playwright.dev/');

  // Expect a title "to contain" a substring.
  await expect(page).toHaveTitle(/Playwright/);
});

test('get started link', async ({ page }) => {
  await page.goto('https://playwright.dev/');

  // Click the get started link.
  await page.getByRole('link', { name: 'Get started' }).click();

  // Expects page to have a heading with the name of Installation.
  await expect(page.getByRole('heading', { name: 'Installation' })).toBeVisible();
});
```

Playwright: **Trace**

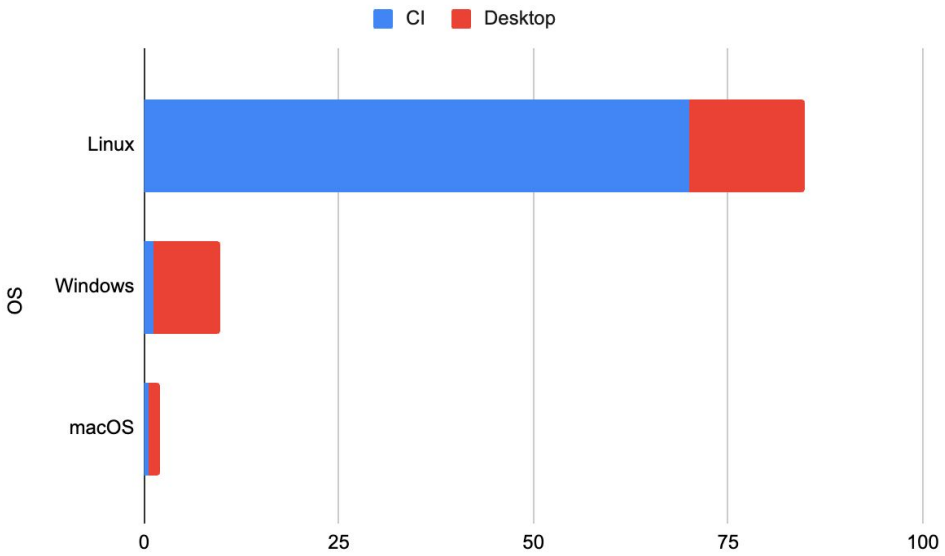


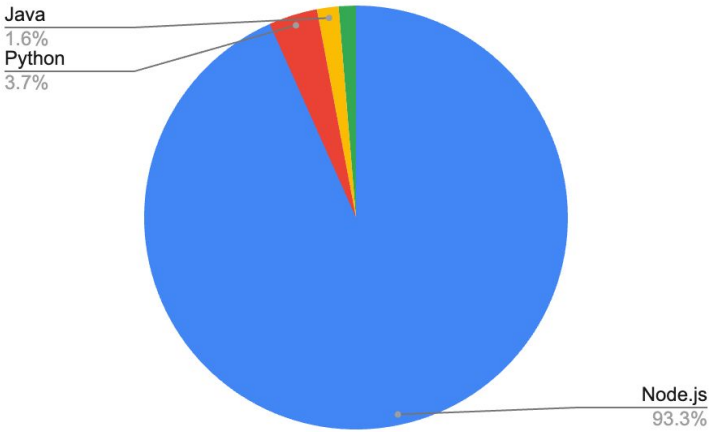
Playwright: **Features**

- **Accessible:** zero configuration, all platforms, all clouds, all containers, all browsers.
- **Capable:** network, emulation, javascript, security, workers, service workers, oopifs, etc.
- **Simple:** resilient role-based locators, auto-waiting and action retries.
- **Reliable:** in-memory browser contexts, removed flakiness.
- **Fast:** parallel headless execution.
- **Complete:** devX, test runner, trace viewer, reporting, recorder, ide

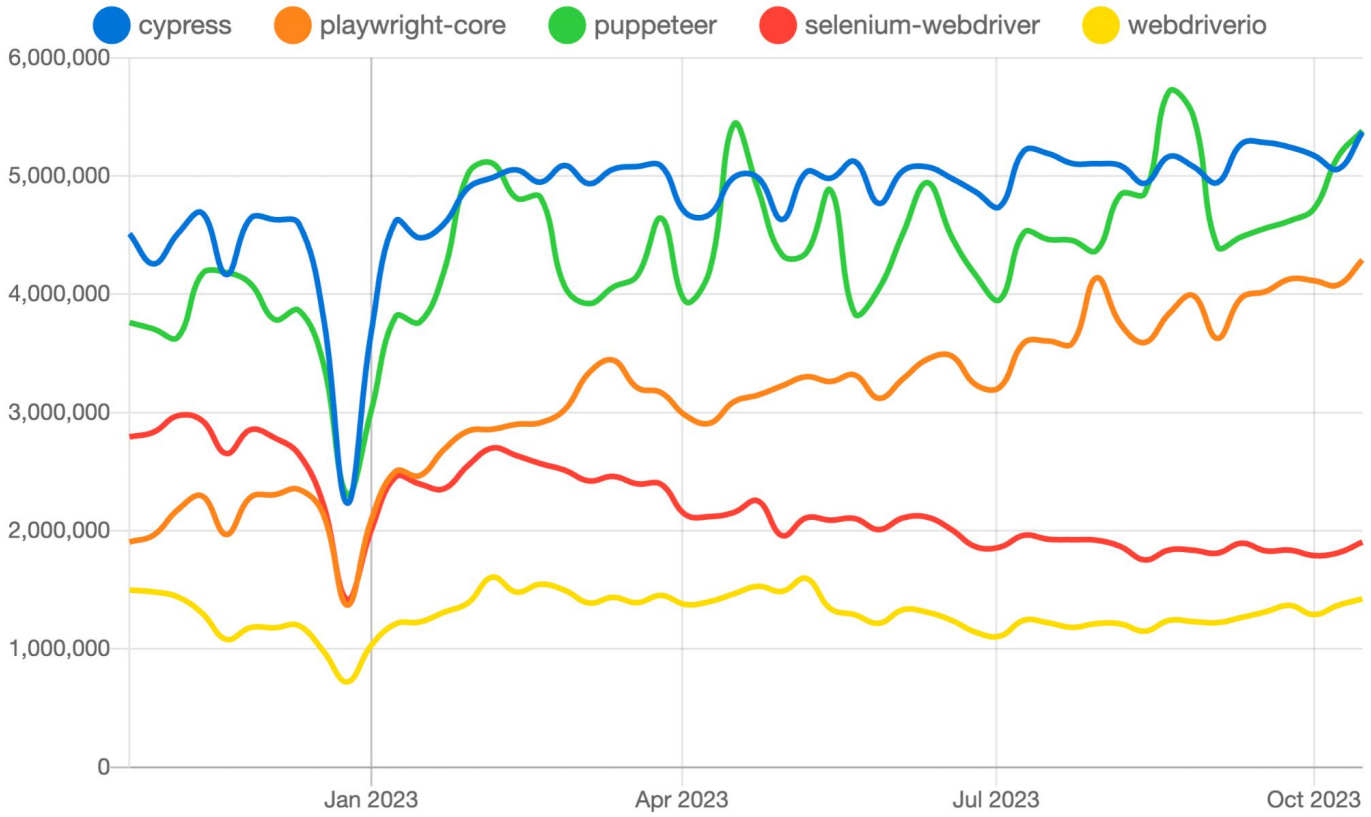
Trends

Trends: **Platform** & **OS**

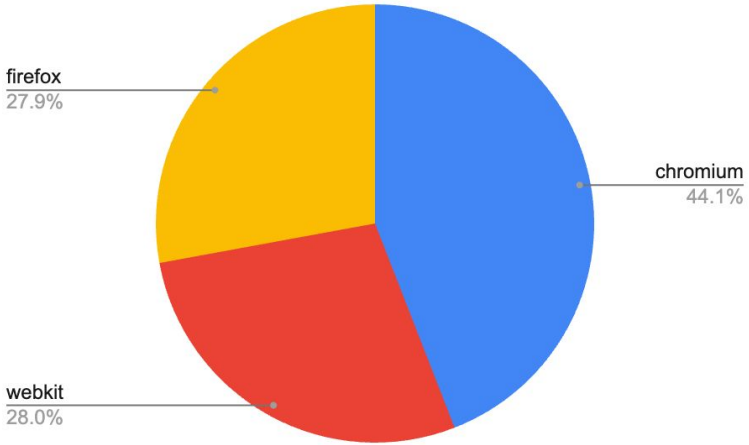




Trends: **NPM downloads**



Trends: **Browsers**



Lessons, challenges

Lessons: **What makes it work**

1. **Integration**: from the **occlusion** detection in the **trusted click** all the way to the **trace viewer** experience.

2. **Product approach:** Playwright is **responsible** for the bugs and regressions, we can't afford **click** does not work in iframe in a certain driver implementation.

3. **Cadence**: Monthly release cycle allows fixing bugs as they are reported (upstreamed where possible).

4. **Quality:** Extensive test coverage.

Internals: **Architecture (WebKit)**

- macOS, Windows, Linux
  - headed and headless embedders
- Communication over **Web Inspector Protocol (WebCore)**
  - **navigation** - cross-process, same document
  - **clicks** - occlusion
  - **emulation** - fixed layout, dpr
  - **network** - interception
  - **javascript** - run in execution context, bootstrap
  - **frames** - inspection
  - **workers** - inspection
  - ...
- Target management: **contexts, targets, pipe**.

Challenges

- Emulation: touch code is missing in macOS & GTK
- Fixed layout assumes iOS
- Screenshots across platforms - consistent compositing
- Network stack inconsistencies (oh well)
- No great upstream avenue - Web Inspector Protocol is not accessible to the embedder

Resources

- Cross-browser Web Testing and Automation Framework
- Documentation: https://playwright.dev
- Source / Issues: https://github.com/microsoft/playwright
- Social:
  - https://aka.ms/playwright/discord
  - https://aka.ms/playwright/twitter
  - https://aka.ms/playwright/youtube

Q&A