

Playwrightの紹介



Press Space for next page →





自己紹介

- 📝 飯野陽平 ([wheatandcat](https://www.wheatandcat.me/))
- 🏢 個人事業主 → 法人設立 (合同会社UNICORN 代表社員)
- 💻 Work: シェアフル株式会社CTO
- 📖 Blog: <https://www.wheatandcat.me/>
- 🛠️ 今までに作ったもの
 - [memoir](#)
 - [ペペロミア](#)
 - [Atomic Design Check List](#)

Playwrightとは？

- E2EテストのためのNode.jsのライブラリ
- Chromium、Firefox、および WebKit ブラウザをサポート
- テストコードをブラウザ操作から自動生成できるのが強み
- Microsoftが作成
- 開発に元Puppeteerの開発者がいるため、インタフェースが似ている

導入

以下のコマンドで実行で導入可能

```
npm init playwright@latest
```

コード

以下のコードで実行できる

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

test.describe("check website", () => {
  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 the URL to contain intro.
    await expect(page).toHaveURL(/.*intro/);
  });
});
```

実行①

以下のコマンドで実行

```
playwright test
```

実行②

テスト結果は以下のように確認できる

🔍	All 2	Passed 1	✖ Failed 1	Flaky 0	Skipped 0
▼	todo.spec.ts				6.7s
✖	check website › has title				5.7s
	todo.spec.ts:4				
✔	check website › get started link				1.1s
	todo.spec.ts:11				

実行③

対象のテストをクリックで詳細も確認できる

```

  Errors

Error: expect(received).toHaveTitle(expected)

Expected pattern: /Praywright/
Received string: "Fast and reliable end-to-end testing for modern web apps | Playwright"
Call log:
- expect.toHaveTitle with timeout 5000ms
- waiting for locator(':root')
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- waiting for locator(':root')
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"
- locator resolved to <html lang="en" dir="ltr" data-theme="light" data-rh="…>…</html>
- unexpected value "Fast and reliable end-to-end testing for modern web apps | Playwright"

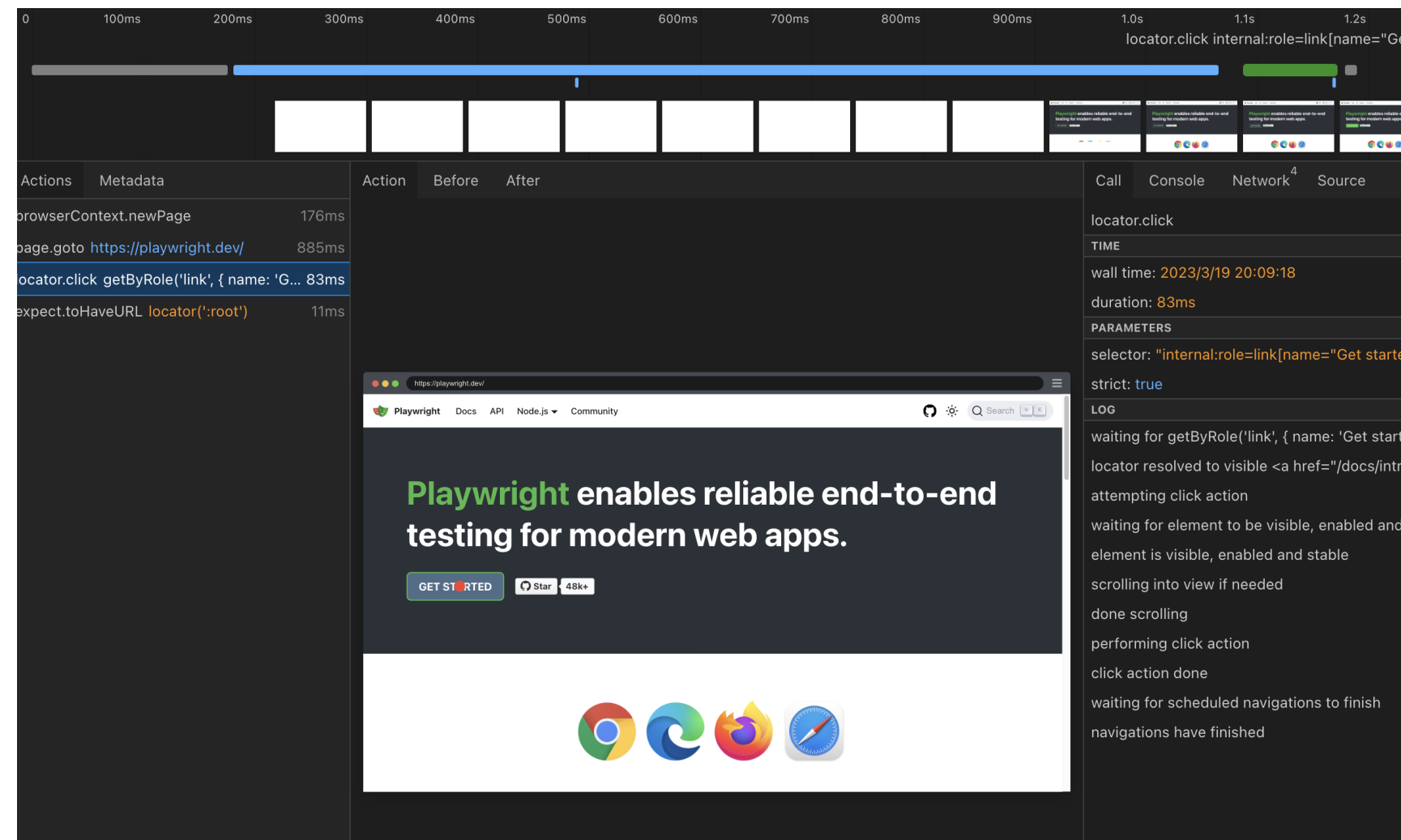
6 |
7 | // Expect a title "to contain" a substring.
> 8 | await expect(page).toHaveTitle(/Praywright/);
    |                   ^
9 | });
10 |
11 | test("get started link", async ({ page }) => {

at file:///Users/iinoyouhei/go/src/github.com/wheatandcat/todo/e2e/todo.spec.ts:8:24
```


実行④

Trace機能を使用して確認

```
playwright test --trace on  
npx playwright show-report
```

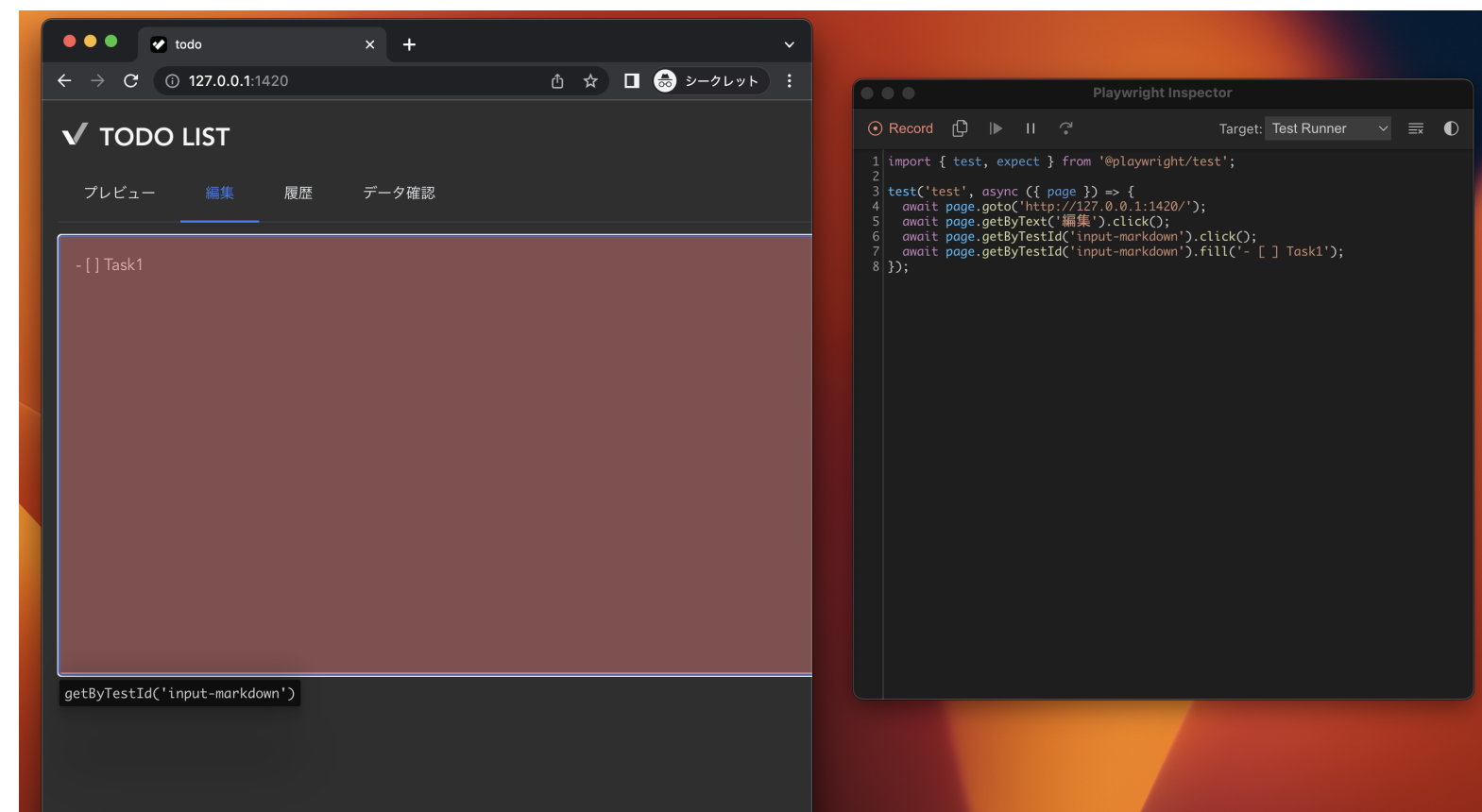


- [Trace Viewer | Playwright](#)

テストコードの自動生成

以下のコマンドで実行

```
npx playwright codegen
```



※デモをやる

- [Test Generator | Playwright](#)

実装

- 自作のアプリに導入してみたので紹介
- PR: [e2eで保存したデータを検証](#)

実装したモチベーション

- アプリで使用している[markdown-to-jsx](#)がユニットテストで動作しない
- Markdownの入力をparseしてTODO管理するアプリなので、`markdown-to-jsx`の入/出力が重要
- なので、`Playwright`で継続的にテストできるように実装

実装内容

※デモしながら紹介（トレースも見せる）

- テストファイルは以下を参照
 - [e2e/todo.spec.ts](#)
 - localStorageのテストは`evaluate`を使用
 - [Evaluating JavaScript | Playwright](#)
- CIで実行は以下を参照
 - [.github/workflows/playwright.yml](#)
 - 実行結果は、[こちら](#)

まとめ

- E2Eテストは保守が大変だが、Playwrightはテストコードの生成機能があるので保守しやすい
- 複数のブラウザをサポートしているので安全性が高い
- トレース機能があるのでCIでコケた時も原因の発見が容易

ご清聴ありがとうございました 🎉