Playwright tooling

Mit den richtigen Werkzeugen geht es besser

CLI

Versucht mal folgendes:



Wichtig für lokale Projekte & CI/CD

VSCode Extension

```
TEST
                              ··· A navigation_intro.test.ts ×
                                   02_einstieg_in_playwright > example > tests > 09_navigation > 🖍 navigation_intro.test.ts > 😚 test('navigation intro') callback

√ TEST-EXPLORER

                                       1 import { expect, test } from '@playwright/test';
 Filtern (z. B. Text, !exclude, @tag)
0 0/0
                                           const homeTitle = 'Bewerben Sie sich hier für ein Konto';

∨ ○ 02_einstieg_in_playwright

∨ ○ example

                                           const savingsTitle = 'Wir nehmen ihr Geld und lassen es arbeiten!'

∨ ○ tests

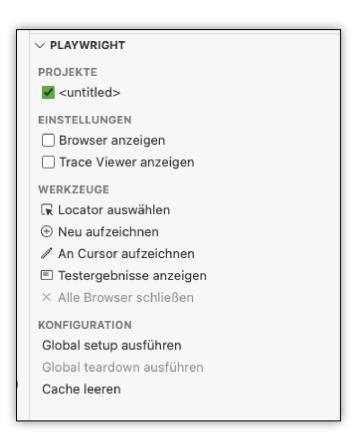
  > 0 01_basic
   > 0 02_fixtures
                                           test('navigation intro', async({ page }) => {
  > 0 03_expect
   > 0 04_async_await
                                              await page.goto('/');
                                        8
   > 0 05_locators
   > 0 06_filters
                                        9
   > 0 07_multi_matches
                                      10
                                              await page.goto('/savings.html');
   > 0 08_more_locators
                                              await expect(page).toHaveTitle(savingsTitle);
                                      11

∨ ○ 09_navigation

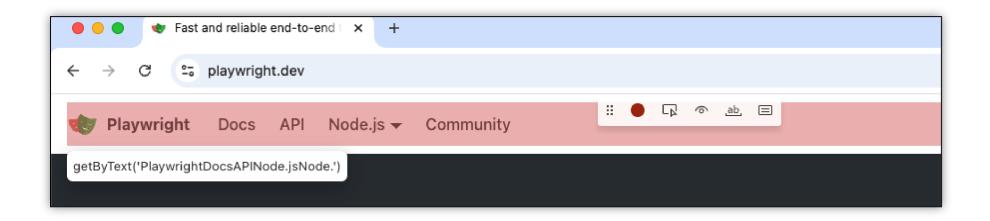
∨ ○ navigation_intro.test.ts

                                      12
    o navigation intro ▷ ♣> 🖰 🔿
                                      13
                                              await page.goBack();
   > o navigation_performance.test.ts
                                              await expect(page).toHaveTitle(homeTitle);
   > o navigation_timeout.test.ts
                                      14
   > 0 10_fill
                                      15
   > 0 11_checking
                                      16
                                              await page.goForward();
   > 0 12_selecting
   > 0 13_assertions_mixed
                                              await expect(page).toHaveTitle(savingsTitle);
                                      17
   > 0 14_soft
                                      18
   > 0 15_dialogs
                                      19
                                              await page.reload();
   > 0 16_events
   > 0 17_cookies
                                              await expect(page).toHaveTitle(savingsTitle);
                                      20
   > 0 18_local_storage_evaluate
                                      21
   > 0 19 downloads
                                      22 });
   > 0 20_uploads
   > 0 21_screenshots
```

VSCode Extension



VSCode Extension - Codegen



Trace Viewer

Traces:

- Actions
- Screenshots
- Logs
- Console messages
- HTTP traffic
- Other metadata

Trace Viewer

```
use: {
8
9
      trace: 'on'
10
      baseURL:
11
      headless: □ on-all-retries
      12
     },
13
               □ retain-on-failure
14
               □ retain-on-first-failure
15
```

on-first-retry ist die empfohlene Wahl.

Trace Viewer

Der Trace Viewer bietet uns viele Optionen, aber lasst es uns erleben:

```
1 cd 02_einstieg_in_playwright/example/tests/09_navigation
2 npx playwright test navigation_intro.test.ts --trace on
```

UI Mode

Der Trace Viewer bietet uns viele Optionen, aber lasst es uns erleben:

```
1 cd 02_einstieg_in_playwright/example/tests/09_navigation
2 npx playwright test --ui
```

Azure + Playwright



https://azure.microsoft.com/de-de/products/playwright-testing

Abschluss



Ende

Das war alles für dieses Kapitel