Day - 8

13.02.2025

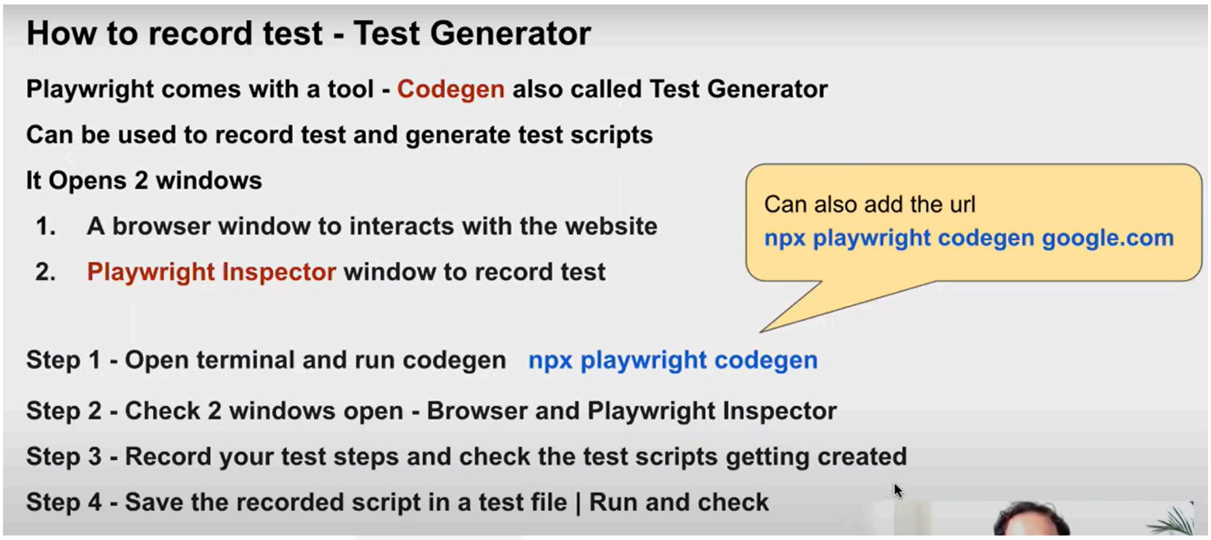
Introduction

- There is no need to wait for anything prior to performing an action: Playwright automatically waits for the wide range of actionability checks to pass prior to performing each action.
- There is also no need to deal with the race conditions when performing the checks - Playwright assertions are designed in a way that they describe the expectations that need to be eventually met.
- That's it! These design choices allow Playwright users to forget about flaky timeouts and racy checks in their tests altogether.

```
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();

});
```

Test generator

Playwright comes with the ability to generate tests for you as you perform actions in the browser and is a great way to quickly get started with testing. The playwright will look at your page and figure out the best locator, prioritizing role, text and test id locators. If the generator finds multiple elements matching the locator, it will improve the locator to make it resilient and uniquely identify the target element.



How to record test - Test Generator

Playwright comes with a tool - Codegen also called Test Generator

Can be used to record test and generate test scripts

It Opens 2 windows

1. **A browser window to interacts with the website**
2. **Playwright Inspector window to record test**

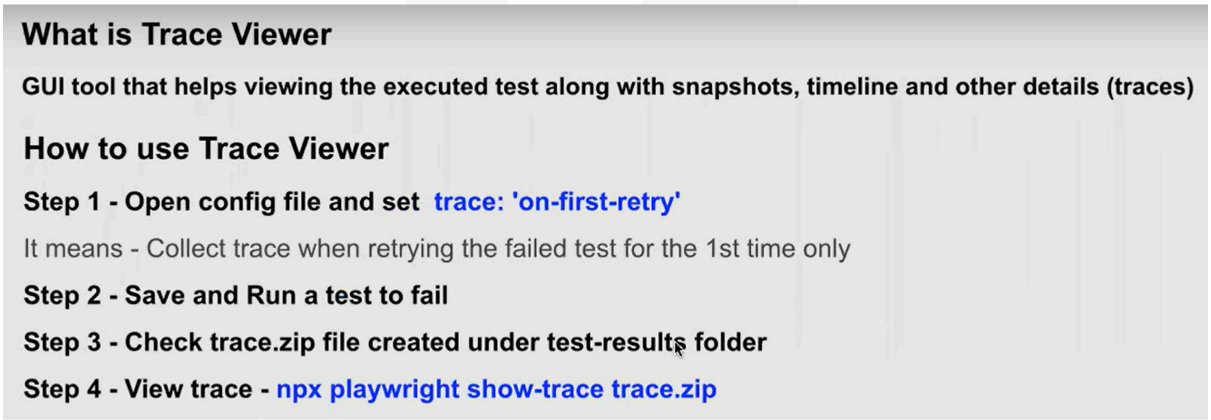
Can also add the url
npx playwright codegen google.com

Step 1 - Open terminal and run codegen npx playwright codegen

Step 2 - Check 2 windows open - Browser and Playwright Inspector

Step 3 - Record your test steps and check the test scripts getting created

Step 4 - Save the recorded script in a test file | Run and check



What is Trace Viewer

GUI tool that helps viewing the executed test along with snapshots, timeline and other details (traces)

How to use Trace Viewer

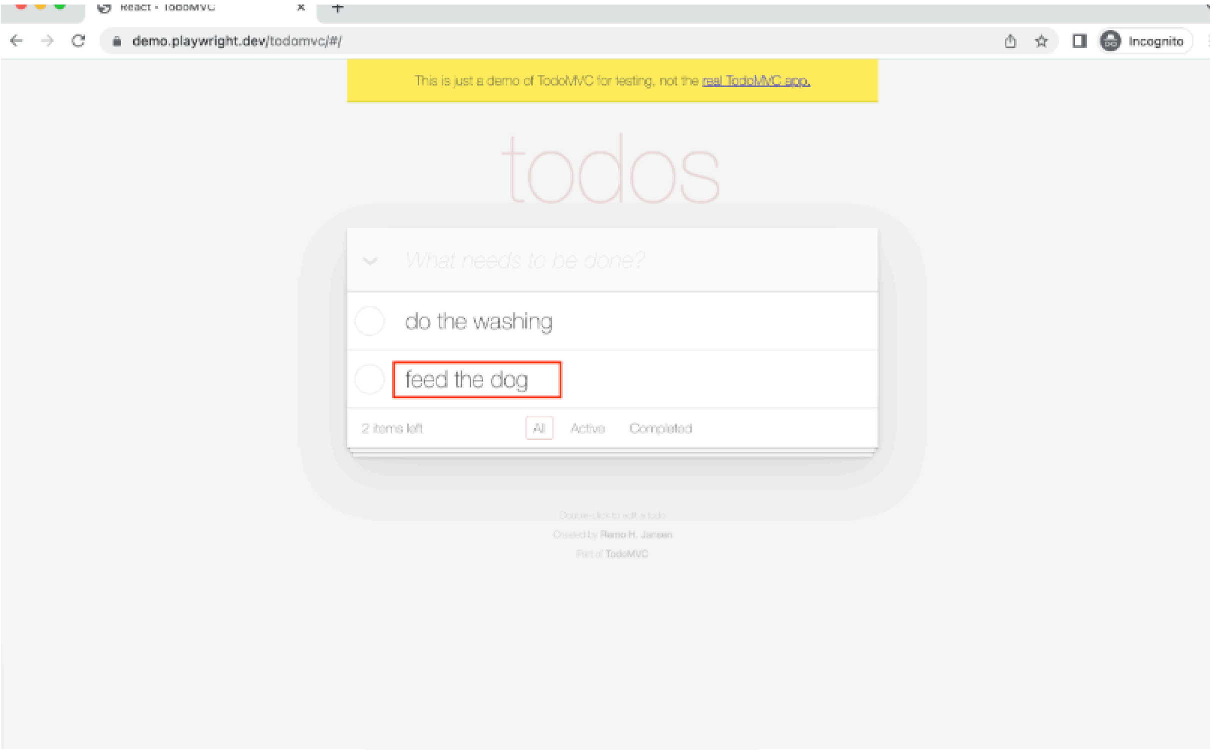
Step 1 - Open config file and set trace: 'on-first-retry'

It means - Collect trace when retrying the failed test for the 1st time only

Step 2 - Save and Run a test to fail

Step 3 - Check trace.zip file created under test-results folder

Step 4 - View trace - npx playwright show-trace trace.zip



This is just a demo of TodoMVC for testing, not the real TodoMVC app.

Generating locators

You can generate locators with the test generator.

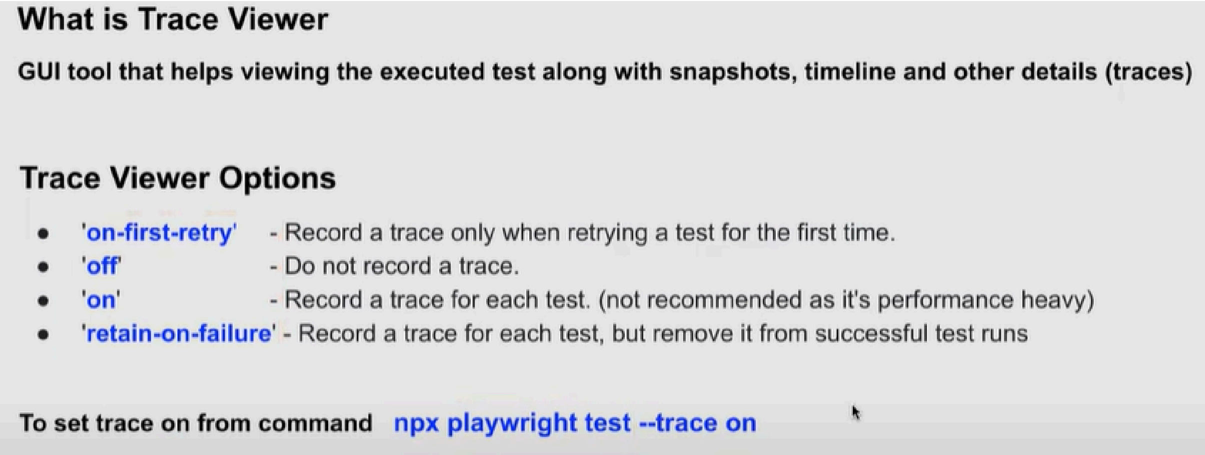
- Press the 'Record' button to stop the recording and the 'Pick Locator' button will appear.
- Click on the 'Pick Locator' button and then hover over elements in the browser window to see the locator highlighted underneath each element.
- To choose a locator, click on the element you would like to locate and the code for that locator will appear in the field next to the Pick Locator button.
- You can then edit the locator in this field to fine tune it or use the copy button to copy it and paste it into your code.

Emulation

You can use the test generator to generate tests using emulation so as to generate a test for a specific viewport, device, color scheme, as well as emulate the geolocation, language or timezone. The test generator can also generate a test while preserving authenticated state.

Emulate viewport size

Playwright opens a browser window with its viewport set to a specific width and height and is not responsive as tests need to be run under the same conditions. Use the --viewport option to generate tests with a different viewport size.



What is Trace Viewer

GUI tool that helps viewing the executed test along with snapshots, timeline and other details (traces)

Trace Viewer Options

- 'on-first-retry' - Record a trace only when retrying a test for the first time.
- 'off' - Do not record a trace.
- 'on' - Record a trace for each test. (not recommended as it's performance heavy)
- 'retain-on-failure' - Record a trace for each test, but remove it from successful test runs

To set trace on from command `npx playwright test --trace on`