



Test generator

Introduction

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. 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 that uniquely identify the target element.

Generate tests with the Playwright Inspector

When running the `codegen` command two windows will be opened, a browser window where you interact with the website you wish to test and the Playwright Inspector window where you can record your tests and then copy them into your editor.

Running Codegen

Use the `codegen` command to run the test generator followed by the URL of the website you want to generate tests for. The URL is optional and you can always run the command without it and then add the URL directly into the browser window instead.

```
playwright codegen demo.playwright.dev/todomvc
```

Recording a test

Run the `codegen` command and perform actions in the browser window. Playwright will generate the code for the user interactions which you can see in the Playwright Inspector window. Once you have finished recording your test stop the recording and press the **copy** button to copy your generated test into your editor.

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.

```
playwright codegen --viewport-size=800,600 playwright.dev
```

Emulate devices

Record scripts and tests while emulating a mobile device using the `--device` option which sets the viewport size and user agent among others.

```
playwright codegen --device="iPhone 13" playwright.dev
```

Emulate color scheme

Record scripts and tests while emulating the color scheme with the `--color-scheme` option.

```
playwright codegen --color-scheme=dark playwright.dev
```

Emulate geolocation, language and timezone

Record scripts and tests while emulating timezone, language & location using the `--timezone`, `--geolocation` and `--lang` options. Once the page opens:

1. Accept the cookies
2. On the top right click on the locate me button to see geolocation in action.

```
playwright codegen --timezone="Europe/Rome" --geolocation="41.890221,12.492348" --lang="it-IT" bing.com/maps
```

Preserve authenticated state

Run `codegen` with `--save-storage` to save `cookies` and `localStorage` at the end of the session. This is useful to separately record an authentication step and reuse it later when recording more tests.

```
playwright codegen github.com/microsoft/playwright --save-storage=auth.json
```

Login

After performing authentication and closing the browser, `auth.json` will contain the storage state which you can then reuse in your tests.

Make sure you only use the `auth.json` locally as it contains sensitive information. Add it to your `.gitignore` or delete it once you have finished generating your tests.

Load authenticated state

Run with `--load-storage` to consume the previously loaded storage from the `auth.json`. This way, all `cookies` and `localStorage` will be restored, bringing most web apps to the authenticated state without the need to login again. This means you can continue generating tests from the logged in state.

```
playwright codegen --load-storage=auth.json github.com/microsoft/playwright
```

Record using custom setup

If you would like to use codegen in some non-standard setup (for example, use `browser_context.route()`), it is possible to call `page.pause()` that will open a separate window with codegen controls.

Sync **Async**

```
from playwright.sync_api import sync_playwright

with sync_playwright() as p:
    # Make sure to run headed.
    browser = p.chromium.launch(headless=False)
```

```
# Setup context however you like.
context = browser.new_context() # Pass any options
context.route('**/*', lambda route: route.continue_())

# Pause the page, and start recording manually.
page = context.new_page()
page.pause()
```