On CI you can’t run the **Safari app**—you run **WebKit** (the Safari engine) via Playwright. Make sure your workflow installs WebKit *with its Linux deps* and that your Playwright project targets browserName: 'webkit' (not "safari"). ([Playwright](https://playwright.dev/docs/browsers?utm_source=chatgpt.com))

Here’s a minimal, known-good GitHub Actions setup that runs Chromium/Firefox/WebKit on ubuntu-latest:

name: e2e

on: [push, pull\_request]

jobs:

test:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-node@v4

with: { node-version: '20' }

**# (Option A) Official action that takes care of browsers & deps**

- uses: microsoft/playwright-github-action@v1

with:

version: 1.x

# (Option B) If you prefer manual install, replace the step above with:

# - run: npm ci

# - run: npx playwright install --with-deps

# # or just WebKit: npx playwright install --with-deps webkit

- run: npm ci

- run: npx playwright test --reporter=line

The equivalent Playwright config should explicitly include a WebKit project:

// playwright.config.ts

import { defineConfig, devices } from '@playwright/test';

export default defineConfig({

projects: [

{ name: 'chromium', use: { ...devices['Desktop Chrome'] } },

{ name: 'firefox', use: { ...devices['Desktop Firefox'] } },

{ name: 'webkit', use: { ...devices['Desktop Safari'] } }, // Safari engine

],

});

**Why it fails on “Safari”**

* **Safari vs WebKit:** Playwright *doesn’t* automate Apple’s branded Safari. It ships its own cross-platform **WebKit** build; use browserName: 'webkit'. If you try to call “Safari” directly, or rely on Safari-only UI, it will fail. ([Playwright](https://playwright.dev/docs/browsers?utm_source=chatgpt.com))
* **Missing system libraries:** On Linux runners, WebKit needs extra packages. Use npx playwright install --with-deps (or the official action above) to install both the browser and OS deps. Errors like “host system is missing dependencies to run browsers” mean that step didn’t happen or ran in a different cache layer. ([Playwright](https://playwright.dev/docs/ci?utm_source=chatgpt.com))

**Quick checklist (fix most CI WebKit issues)**

1. **Install with deps in CI:** npx playwright install --with-deps [webkit] (or use microsoft/playwright-github-action@v1). ([Playwright](https://playwright.dev/docs/ci?utm_source=chatgpt.com))
2. **Target webkit** in your Playwright projects; don’t use "safari". ([Playwright](https://playwright.dev/docs/browsers?utm_source=chatgpt.com))
3. **Keep PW versions aligned:** the Playwright npm package and the downloaded browsers should match (avoid mixing cached old browsers with a new npm version). ([GitHub](https://github.com/microsoft/playwright/issues/20481?utm_source=chatgpt.com))
4. **Avoid Chromium-only assumptions:** features/flags/APIs that exist in Chromium may differ in WebKit (timing, focus behavior, video codecs, etc.). Add small waits around animations, rely on getByRole/getByLabel, and avoid vendor-specific CSS/JS. (General best practice; no special CI step.)
5. **Headed vs headless:** Default is headless and fine on CI. If you forced headed runs, add xvfb-run -a npx playwright test on Linux. (Only if you really need headed runs.)

**“I really need *Safari*”**

If a requirement means “the actual Safari app on macOS/iOS,” Playwright can still only run **WebKit**, not the branded Safari UI. For true Safari/iOS coverage, use a cloud grid with real devices (e.g., BrowserStack/Sauce), which provides Mobile Safari—Playwright will connect but you’re running on their infra. Apple’s restrictions prevent local automation of the Safari app or iOS Safari directly from CI. ([BrowserStack](https://www.browserstack.com/guide/playwright-ios-automation?utm_source=chatgpt.com" \o "Playwright iOS Automation Testing on Real Devices))