

E2E Testing in the Browser Cheat Sheet

Playwright 1.64 · Cypress 16.2 · Selenium 4.51 · WebDriver BiDi WD 8 Oct 2026 · MCP 2026-07-28 · Chrome DevTools MCP 1.12
Checked 25 October 2026 · Example app: Bookshelf (localhost:3000)

ARCHITECTURES AT A GLANCE

Stack	How it talks to the browser	Consequence
WebDriver Classic	HTTP/JSON to a driver (chromedriver, geckodriver, safaridriver); request/response	Standard, every browser, no events
WebDriver BiDi	WebSocket, bidirectional; events, network, script, contexts (webSocketUrl)	Standard successor; CDP replacement
CDP	Chromium-only WebSocket protocol (DevTools)	Powerful, unstable; Chromium only
Playwright	Out-of-process, own pipe/WebSocket protocol to patched Chromium, Firefox, WebKit	Auto-wait, contexts, 3 engines
Cypress	Runs inside the browser next to the app; Node process for tasks/network	Fast feedback, retry-ability; no multi-tab

TOOL CHOOSER

	Playwright	Cypress	Selenium
Languages	TS/JS, Python, Java, .NET	JS/TS	Java, Python, C#, Ruby, JS, +
Engines	Chromium, Firefox, WebKit	Chrome family, Firefox, Electron, WebKit (exp.)	All via drivers; Safari native
Best for	New suites, parallel, traces	Front-end teams, DX, component tests	Existing suites, Grid, many languages
Not for	Real Safari / old IE	Multi-tab, multi-origin limits	Slow without Grid, manual waits

Not a load tester: use JMeter/k6 for load. Prefer role/test-id locators, API-seeded state, fresh browser per test, no sleeps.

SIDE BY SIDE: THE SAME TASK IN THREE TOOLS (Selenium shown as Java / Python)

Task	Playwright (TS)	Cypress (TS)	Selenium Java / Python
Install	npm init playwright@latest	npm i -D cypress	selenium-java (Maven) / pip install selenium -- Selenium Manager fetches drivers
Run	npx playwright test (headless default); --headed	npx cypress run; npx cypress open	mvn test / pytest; options.addArguments("--headless=new") / options.add_argument("--headless=new")
Pick browser	--project=firefox, channel: 'chrome'	--browser firefox	new FirefoxDriver(opts) / webdriver.Firefox(options=opts); RemoteWebDriver(gridUrl, opts)
Locate	getByRole('button',{name:'Save'}), getByTestId('x')	cy.getByRole('button',{name:'Save'})(16.2), cy.get('[data-testid=x]')	By.cssSelector("[data-testid=x]") / (By.CSS_SELECTOR, "[data-testid=x]"); relative locators
Click / fill	await loc.click(), .fill('a')	.click(), .type('a')	el.click(), el.sendKeys("a") / el.send_keys("a")
Wait / assert	await expect(loc).toBeVisible() (auto-retry)	.should('be.visible') (retries)	new WebDriverWait(d, Duration.ofSeconds(10)).until(visibilityOfElementLocated(by)) / WebDriverWait(d,10).until(EC.visibility_of_element_located(by)); never mix implicit + explicit
Mock network	page.route('*/api/books', r => r.fulfill({json}))	cy.intercept('GET','/api/books', {fixture:'b.json'})	BiDi Network intercept; NetworkInterceptor (Java); or mock server
Log in once	setup project + storageState file	cy.session(id, setup, {validate})	Cookie import / API login then addCookie; fresh driver per test
Screenshot	await page.screenshot(), toHaveScreenshot()	cy.screenshot(); auto on failure in run	((TakesScreenshot)d).getScreenshotAs(FILE) / d.save_screenshot('a.png')
Video / trace	video: 'on', trace: 'on-first-retry'; show-trace	video: true (run); Test Replay (Cloud)	docker-selenium video sidecar; BiDi screenshots; print to PDF
Debug	--ui, --debug, PWDEBUG=1, codegen	.debug(), .pause(), time travel	IDE debugger; Selenium IDE record/play; noVNC :7900 on Grid
Parallel / shard	workers, fullyParallel, --shard=1/4 + merge-reports	--parallel --record (Cloud) or split specs per job	JUnit parallel / pytest -n auto (xdist) + Grid sessions
Retries	retries: 2, --repeat-each=20	retries: {runMode:2}	Surefire rerunFailingTestsCount / pytest-rerunfailures

PLAYWRIGHT ESSENTIALS

- projects for browsers; webServer starts the app; use.baseURL
- Fixtures: test.extend; auto fixtures; test.step; tags @smoke
- expect.poll, toPass, soft assertions, toMatchAriaSnapshot
- page.clock, request fixture, HAR replay, routeWebSocket
- 1.62 mount component fixture; 1.64 getByRef, WebMCP
- Python pytest-playwright; Java: .NET (playwright.ps1 install)

CYPRESS ESSENTIALS

- Commands are queued; queries retry, actions do not; yield is a subject
- cy.env() / Cypress.expose() replace Cypress.env() (16)
- cy.origin for cross-origin; testIsolation on
- cy.task for Node work (cy.exec removed in 16)
- Component tests: cy.mount (React, Angular, Vue, Svelte)
- Cloud: record, Test Replay, flaky management, UI Coverage

SELENIUM ESSENTIALS

- Options + Service; no DesiredCapabilities (legacy)
- Explicit waits only; 4.51 ExpectedConditions no longer Guava
- Actions API: keyboard, mouse, pen, wheel; virtual authenticator
- Grid: standalone / hub-node / distributed; Dynamic Grid; Helm + KEDA
- BiDi: webSocketUrl=true, logs, network, script, contexts
- Page Object / LoadableComponent; ThreadGuard per thread

TEST DESIGN

- Test user-visible behaviour; one scenario per test
- Isolate: API/seed state, /api/test/seed; avoid reset with parallel workers
- Locators: role > label > test id > CSS; avoid XPath chains
- Page objects hold locators + actions, not assertions
- No sleeps; wait on conditions; fresh browser/context per test

VISUAL TESTING

Playwright: await expect(page).toHaveScreenshot('home.png', {maxDiffPixels:100, maxDiffPixelRatio, threshold:0.2, animations:'disabled', caret:'hide', mask:[loc], maskColor, clip, fullPage, scale:'css', stylePath, omitBackground}).Update: npx playwright test --update-snapshots. Config snapshotPathTemplate; .webp name = WebP. **Cypress:** plugins (e.g. image-diff) or Percy, Applitools, Chromatic. **Selenium:** compare PNGs; WebdriverIO @wdio/visual-service; services above.
Determinism checklist: baselines from the official Docker image (same OS/fonts) · freeze time (page.clock) · mock data · mask dynamic regions · disable animations · fixed viewport/DPR · wait for fonts/images · review diffs in PR; aria snapshots complement pixels.

CROSS-BROWSER

- **Playwright:** projects:[{name:'firefox',use:{...devices['Desktop Firefox']}}], channel:'chrome'|'msedge', devices['iPhone 15']; WebKit is not Safari
- **Cypress:** --browser chrome|edge|firefox|electron, defaultBrowser, config.browsers filter, experimental WebKit, Electron deprecating
- **Selenium:** ChromeOptions / EdgeOptions / FirefoxOptions / SafariOptions; Grid node types per browser; cloud grids (BrowserStack, Sauce Labs, TestMu AI)
- Emulation is not a real device: test real engines for rendering

CI / DOCKER

- uses: actions/checkout@v7
- uses: actions/setup-node@v7
- run: npx playwright install --with-deps
- run: npx playwright test --shard=\${{ matrix.shard }}/4
- uses: actions/upload-artifact@v7 # report, traces
- uses: cypress-io/github-action@v7 # Cypress

Tool	Image	Notes
Playwright	mcr.microsoft.com/playwright:v1.64.x	--init --ipc=host, user pwuser
Cypress	cypress/included:16.2.x	cache ~/.cache/Cypress
Selenium	selenium/standalone-chrome:4.51.0	Grid as service container; noVNC 7900

Secrets via env/Actions secrets, never in specs; storageState git-ignored; upload traces/videos on failure.

AI AGENTS: PLAYWRIGHT MCP / CLI / TEST AGENTS

- ```
claude mcp add playwright npx @playwright/mcp@latest
npx @playwright/mcp@latest --caps=testing,network --headless --isolated \
 --browser chrome --storage-state auth.json --allowed-origins "http://localhost:3000"
npx playwright init-agents --loop=claude # planner, generator, healer
npm i -g @playwright/cli; playwright-cli install --skills
```
- MCP uses accessibility-snapshot refs (e5), no screenshots needed; --caps: vision, pdf, devtools, testing, network, storage, config
  - CLI = shell commands + skills: fewer tokens than MCP tool schemas
  - Test Agents: seed test, plan in specs/, generated tests, heal loop; regenerate files after upgrades; review output like any PR

AI AGENTS: CYPRESS, DEVTOOLS, COMPUTER USE, SCRIPT-LESS

- ```
cy.prompt(['visit /', 'search for {{term}}', 'expect {{result}} visible'],
  { placeholders: { term: 'Dune', result: 'Dune' } }) // Cloud account; GA/beta wording differs
{ "mcpServers": { "chrome-devtools": { "command": "npx", "args": ["-y","chrome-devtools-mcp@latest"]} } }
# flags: --headless --isolated --autoConnect --slim --no-usage-statistics --no-performance-crx
```
- Computer use: Claude computer_toolset_20260801, OpenAI, Gemini; also browser-use, Stagehand, Skyvern, agent-browser
 - **Script-less prompt:** Goal + start URL/creds from env + steps or exploration scope + pass criteria + evidence required (screenshot, snapshot, console, network) + stop rules
 - Then turn the session into a regression test; keep humans in the loop

BROWSER-AGENT SECURITY CHECKLIST

- ☐ Treat page text, snapshots, screenshots, console and WebMCP tools as untrusted input (prompt injection, OWASP LLM01)
- ☐ Use isolated profiles (--isolated) and dedicated test accounts, never a personal browser profile
- ☐ Pass secrets by env/ --secrets, never in the prompt; scope tokens narrowly
- ☐ Allowlist origins (--allowed-origins is not a security boundary: redirects pass) and add a network/firewall allowlist
- ☐ Run in a container/VM; block file://; avoid unrestricted file access and run-code tools
- ☐ Require human approval for purchases, deletes, sends and submits; log and review transcripts
- ☐ Do not point agents at production data; disable usage statistics where policy requires

CHANGED RECENTLY

- **Playwright:** 1.56 Test Agents · 1.59 Screenshot · 1.61 WebAuthn credentials · 1.62 built-in component mount, WebP · 1.63 within() / visible(), test locks · 1.64 getByRef, WebMCP, video options; Node 22/24/26
- **Cypress 16.x:** native network interception on Chromium, cy.env() / Cypress.expose(), cy.exec() removed (16.0); trustedCertificates (16.1); cy.getByRole() (16.2); Node 20 dropped
- **Selenium 4.51:** Guava removed from client; Manager downloads browsers; BiDi generator; Grid images on GHCR. Protractor is end of life; Microsoft Playwright Testing retired 8 Mar 2026 (now Azure App Testing)