

/webapp-testing

Source: `~/ .claude/skills/webapp-testing/SKILL.md`

name: webapp-testing **description:**
Toolkit for interacting with and testing local web applications using Playwright. Supports verifying frontend functionality, debugging UI behavior, capturing browser screenshots, and viewing browser logs. **license:**
Complete terms in LICENSE.txt

Web Application Testing

To test local web applications, write native Python Playwright scripts.

Helper Scripts Available:

- `scripts/with_server.py` - Manages server lifecycle (supports multiple servers)

Always run scripts with `--help` first to see usage. DO NOT read the source until you try running the script first and find that a customized solution is absolutely necessary. These scripts can be very large and thus pollute your context window. They exist to be called directly as black-box scripts

rather than ingested into your context window.

Decision Tree: Choosing Your Approach

User task → Is it static HTML?

- └ Yes → Read HTML file directly to identify selectors
 - | └ Success → Write Playwright script using selectors
 - | └ Fails/Incomplete → Treat as dynamic (below)
 - |
- └ No (dynamic webapp) → Is the server already running?
 - | └ No → Run: `python scripts/with_server.py --help`
 - | Then use the helper + write simplified Playwright script
 - |
 - | └ Yes → Reconnaissance-then-action:
 1. Navigate and wait for networkidle
 2. Take screenshot or inspect DOM
 3. Identify selectors from rendered state
 4. Execute actions with discovered selectors

Example: Using `with_server.py`

To start a server, run `--help` first, then use the helper:

Single server:

```
python scripts/with_server.py --server "npm run dev" --port 5173 -- python your_automation.py
```

Multiple servers (e.g., backend + frontend):

```
python scripts/with_server.py \  
  --server "cd backend && python server.py" --port 3000 \  
  --server "cd frontend && npm run dev" --port 5173 \  
  -- python your_automation.py
```

To create an automation script, include only Playwright logic (servers are managed automatically):

```
from playwright.sync_api import sync_playwright

with sync_playwright() as p:
    browser = p.chromium.launch(headless=True) # Always launch chromium in headless mode
    page = browser.new_page()
    page.goto('http://localhost:5173') # Server already running and ready
    page.wait_for_load_state('networkidle') # CRITICAL: Wait for JS to execute
    # ... your automation logic
    browser.close()
```

Reconnaissance-Then-Action Pattern

1. Inspect rendered DOM:

```
page.screenshot(path='/tmp/inspect.png', full_page=True)
content = page.content()
page.locator('button').all()
```

2. **Identify selectors** from inspection results
3. **Execute actions** using discovered selectors

Common Pitfall

❌ **Don't** inspect the DOM before waiting for `networkidle` on dynamic apps ❌ **Do** wait for `page.wait_for_load_state('networkidle')` before inspection

Best Practices

- **Use bundled scripts as black boxes** - To accomplish a task, consider whether one of the scripts available in `scripts/` can help. These scripts handle common, complex workflows reliably without cluttering the context window. Use `--help` to see usage, then invoke directly.
- Use `sync_playwright()` for synchronous scripts
- Always close the browser when done
- Use descriptive selectors: `text=`, `role=`, CSS selectors, or IDs
- Add appropriate waits: `page.wait_for_selector()` or `page.wait_for_timeout()`

Reference Files

- **examples/** - Examples showing common patterns:
 - `element_discovery.py` - Discovering buttons, links, and inputs on a page
 - `static_html_automation.py` - Using file:// URLs for local HTML
 - `console_logging.py` - Capturing console logs during automation

Known Issues & Fixes

2026-04-03 16:12:33

Error: Deploy verification was skipped - John claimed 'sve live' based on curl, CEO found 404. Fix: MANDATORY browser click-through test after every deploy before ANY claim to CEO. Add deploy-verify checklist.

Fix: MANDATORY browser click-through test after every deploy before ANY claim to CEO. Add deploy-verify checklist.

Revision #6

Created 2026-02-18 08:42:05 UTC by John

Updated 2026-06-21 20:01:35 UTC by John