End to End Testing for Browser Extensions







About Me

- Senior Software Engineer at OpenSauced
- From Montreal, Quebec, Canada
- I'm <u>@nickytonline</u> everywhere
- Also find me at <u>nickyt.co</u>
- Streaming at <u>nickyt.live</u>
- Not a big fan of spiders







What We'll Cover



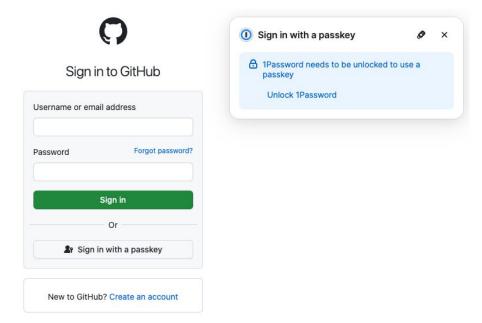
What We'll Cover

- What is a browser extension?
- What is end to end (E2E) testing?
- Browser extension E2E testing in action
- Questions



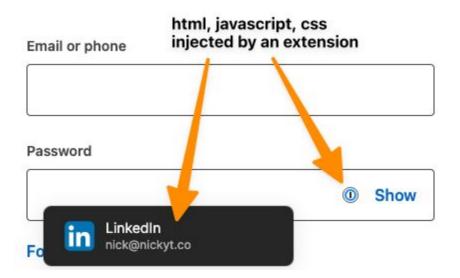


As the name suggests, it's an extension of the browser. Most of you probably use at least one browser extension, like a password manager.





A browser extension can alter the user interface of a website and add additional interactivity





It can inject:

- HTML
- JavaScript
- CSS



It can:

- Use <u>browser storage</u>
- Sync extension settings across browser instances (on multiple devices)
- from the application sidebar
- Call application programming interfaces (APIs)



The manifest is the main piece of the puzzle

```
"manifest version": 3,
"name": "OpenSauced.ai",
"version": "1.14.2",
"action": { "default popup": "index.html" },
"content_scripts": [
"js": ["src/content-scripts/github.ts"],
"matches": ["https://github.com/*"],
 --- "run": - "document end"
"background": {
 "service_worker": "src/worker/background.ts",
  "tupe": "module"
"icons": {
 "16": "src/assets/os-icons/os-icon-16.png".
  -"32": "src/assets/os-icons/os-icon-32.png",
  "48": "src/assets/os-icons/os-icon-48.png".
  "128": "src/assets/os-icons/os-icon-128.png"
"host_permissions": ["https://github.com/*", "https://*.app.opensauced.pizza/*", "https://www.linkedin.com/*"],
"permissions": ["scripting", "storage", "tabs", "cookies"]
```



Some general metadata:

```
"manifest_version": 3,
"name": "OpenSauced.ai",
"version": "1.14.2",
```



```
"content scripts": [
"js": ["src/content-scripts/github.ts"],
"matches": ["https://github.com/*"],
"run": "document_end"
```



```
"background": {
    "service_worker": "src/worker/background.ts",
    "type": "module"
},
```



```
"icons": {
    "16": "src/assets/os-icons/os-icon-16.png",
    "32": "src/assets/os-icons/os-icon-32.png",
    "48": "src/assets/os-icons/os-icon-48.png",
    "128": "src/assets/os-icons/os-icon-128.png"
},
```



```
"host_permissions": [
    "https://github.com/*",
    "https://*.app.opensauced.pizza/*",
    "https://www.linkedin.com/*"
],
```



```
"permissions": [
   "scripting", "storage", "tabs", "cookies"
]
```



```
<button</pre>
className="flex items-center ■bg-slate-700 ■h
    gap-2 p-1.5 px-3 w-full rounded-sm font-medium
onClick=\{() \Rightarrow \{
const message: Message = {
type: MessageType.LinkedInProject,
data: { url: pageUrl },
void chrome.runtime.sendMessage(message);
. . . . . }}
<IoLogoLinkedin />
Add to LinkedIn Projects.
</button>
```





Building an extension: CRXJS



We have getting started guides for the following frameworks:

- **React**
- **Solid**
- **Vanilla JavaScript**
- **V** Vue

The first two sections for any framework only take 90 seconds to complete.



Building an extension: CRXJS

- You get hot module reloading (HMR)
- Static asset imports
- Access to the Vite ecosystem
- Lots of frameworks to build an extension with

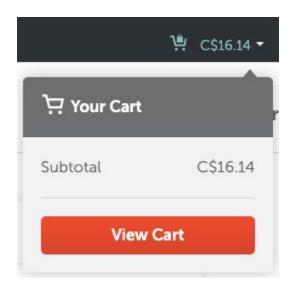


What is end to end testing?



What is end to end testing?

End to End (E2E) testing allows you to test your application through a whole flow in a browser, e.g. testing login, adding to a shopping cart.





What is end to end testing?

Unlike unit tests, component tests, and integration tests which test certain aspects of your application like business logic, a component or an API, E2E tests test your application like a real user.



What makes a good E2E test?

Should test a full flow or journey in an application:

- Login
- Onboarding
- Adding an item to a cart
- Completing a purchase
- etc.



What makes a good E2E test?

- Can it be rerun and have the same outcome?
- Does it run quickly?
- Not flaky



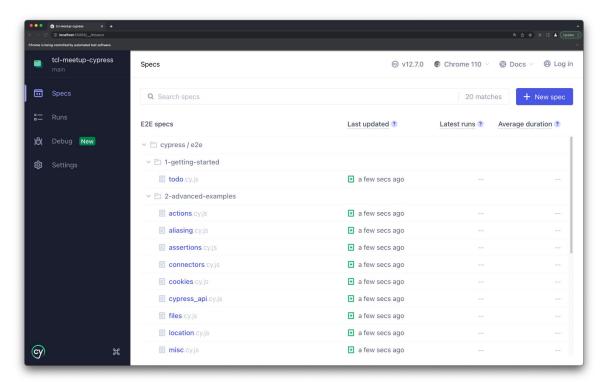
- We won't cover all of them
- Cypress
- Playwright



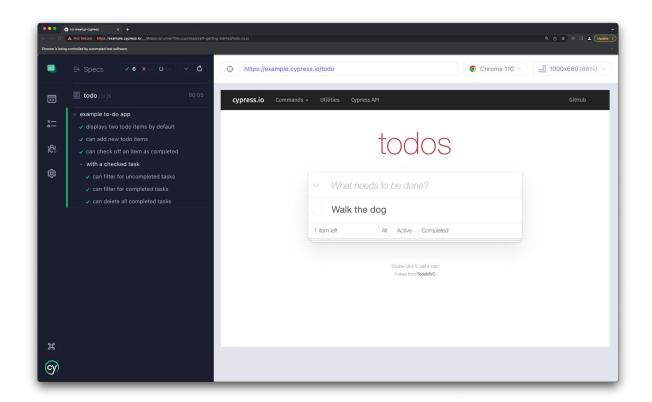
- User interface for local development
- Run headless for continuous integration/continuous deployment (CI/CD)
- Supports multiple browsers



Cypress UI

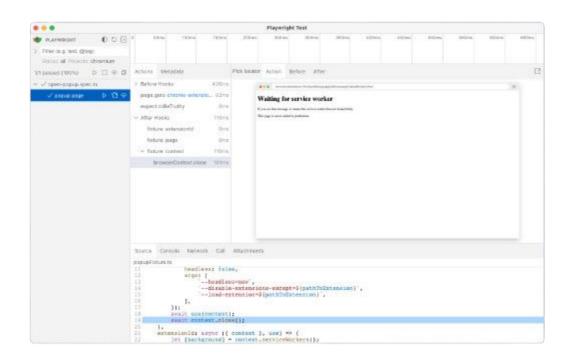




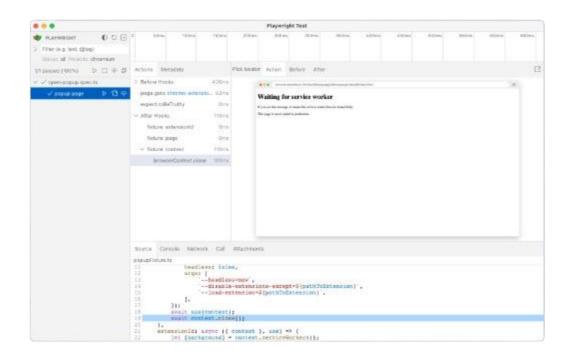




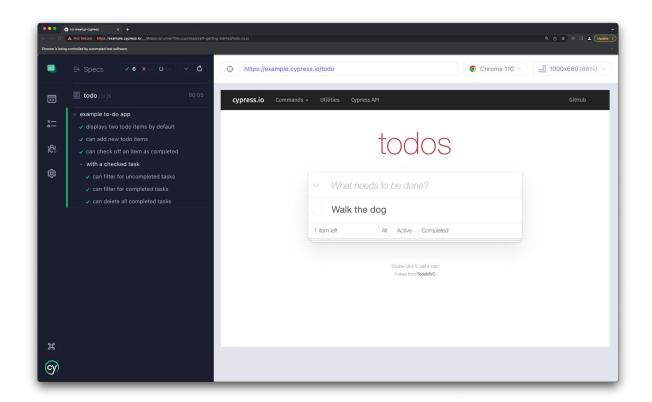
Playwright UI













Writing your first Cypress Test

```
it('Smoke Test', async () => {
    cy.visit('/');
    cy.findByText('hello').should('exist');
});
```



Writing your first Cypress Test

Using Cypress, definitely install <u>Cypress Testing Library</u>





Writing your first Playwright Test

```
test('Smoke Test', async ({ page }) => {
   await page.goto('/');
   expect(page.locator('body')).toContainText('hello');
});
```



Loading the extension in E2E tests

```
import { test as base, chromium, type BrowserContext } from "@playwright/test";
import path from "path";
export const test = base.extend<{
context: BrowserContext;
extensionId: string;
}>({
context: async (\{\}, use) \Rightarrow \{
const pathToExtension = path.join(__dirname, "../dist");
const context = await chromium.launchPersistentContext("", {
••••••••••headless: false, Marcus Chan, 6 months ago via PR #262 • test: s
args:
await use(context);
await context.close();
```



Loading the extension in E2E tests

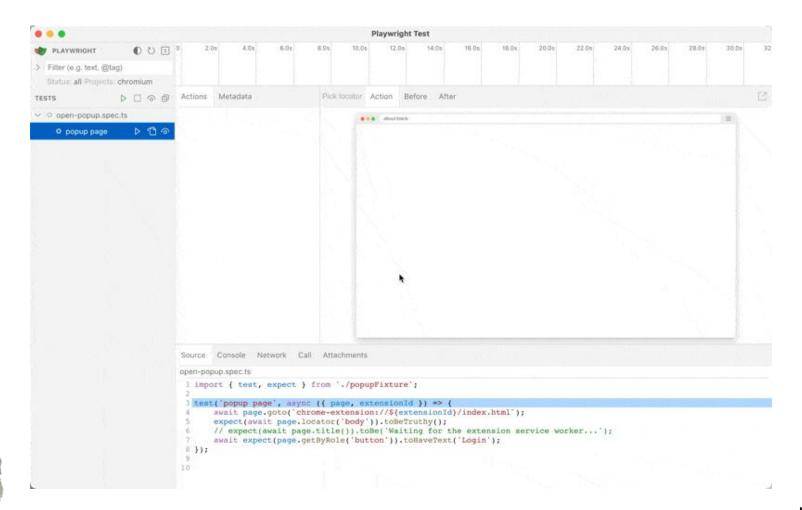
```
extensionId: async ({ context }, use) => {
... let [background] = context.serviceWorkers();
... if (!background) background = await context.waitForEvent("serviceworker");
... const extensionId = background.url().split("/")[2];
... await use(extensionId);
...},
});
export const expect = test.expect;
```



```
ai on beta [!] via v20.11.0

o ) nr e2e_
```









- Chrome for Developers: Extensions > Get started
- Mozilla > Add-ons > Browser Extensions > Your first extension
- Manifest v3
- CRXJS
- Playwright
- Cypress



- How to debug a Firefox add-on (extension)
- End to End Testing with Cypress
- Expert Panel: Trending Tools and Frameworks What's Hype and What's Not
- Principle of least privilege



Some open source browser extension examples I've contributed to:

- OpenSauced AI
- Refined GitHub
- a11y Twitter



Slide deck available at nickyt.co/slides/ext-e2e



That's all folks!

Thank you!

Stay in touch!

- @nickytonline everywhere
- nickyt.co
- <u>nickyt.live</u>
- <u>nickyt.tube</u>

