# Playwright for Scraping

(Research)

**Why use Playwright?**

Open source browser automation library→ developed by Microsoft

Supports→ Chrome, Firefox, Webkit

Good at scraping JS heavy sites where scrapers like (requests) fail

Python —→ Playwright( acts as a middle man) —→ Chrome

Supports many programming languages
Allows scraping dynamic content like replies, infinite scroll ,popups

**When to prefer Playwright over Snscrape?**

**Snscrape** → fast, works without login, but limited to accessible endpoints (can miss replies/hidden threads).

**Playwright** → slower (real browser), but captures everything exactly as a user sees it.

One of its advantages is Playwright doesn't require any API keys

Launches on the machine→ interacts with the site → no authentication keys dev access required

Playwright works well with many dynamic websites (e.g., Reddit, news sites, forums), but scraping platforms like Instagram or Facebook requires caution due to strict anti bot policies.

**Workflow**

Install and launch

Navigate to the page

Content will be loaded

Use CSS selectors to extract data

Handles  scroll, dynamic rendering and pagination

Load any public Twitter profile or tweet ,captures rendered HTML, text, images, timestamps, links, replies, etc...

Handles infinite scroll to fetch older tweets or more replies.

Capture content generated by JavaScript that static scrapers miss.

**Installation**

pip install playwright

playwright install

**Issues I faced:**

shows login browser issues

should try with more selectors

headless mode→ less CPU power

headed mode→ high CPU power

Some of the common issues faced by other developers,

→Playwright (async) still heavy (had the same issue)

→Look for more lightweight browsers, if JS heavy websites need to be scraped,

→Issue: Key data like follower count or tweet content is loaded dynamically.

→Solution: Use Playwright's full browser rendering (non-headless mode) and wait for the page to fully load before extracting data. Use `page.waitForSelector()` and `page.content()` to ensure elements are present.

→Issue: Pages may fail to load or return errors.

→Solution: Implement retry logic, exponential backoff, and monitor request success rates. Use Playwright's built in timeout handling.