
Faster browsing. Real logins. Zero hassle.

How to give your OpenClaw agent a real browser with authenticated access to X, GitHub, LinkedIn, and more.

— THE PROBLEM

Your agent browses the web like it's 2004

Right now, when your OpenClaw agent visits a website, it's using a full Chrome browser under the hood. That works, but it's **slow**, uses a ton of memory, and can't log into anything without you manually setting up sessions.

Want your agent to check your X mentions? Read your GitHub notifications? Monitor a LinkedIn thread? It can't, because it's not logged in, and logging in with a headless browser triggers bot detection on every major platform.

What if your agent had its own fast browser that could log in as itself?

That's Plasmate.



10x faster page loads

Plasmate compiles HTML into structured data without rendering pixels. No GPU, no layout engine, no wasted work.



Real authenticated sessions

Log into X, GitHub, or any site yourself. Hand the cookies to your agent. It browses as that account.



Agent-native output

Pages become structured "Semantic Object Models" instead of raw HTML. Your agent understands pages, not pixels.



Tiny footprint

~15MB binary. No Chrome download. Runs on a Raspberry Pi. Uses a fraction of the memory.

`plasmate.app`

What you need

Three pieces, all free and open source. Setup takes about five minutes.

1. Plasmate (the browser engine)

Install with one command:

```
curl -sSf https://plasmate.app/install.sh | sh
```

This puts the `plasmate` binary on your machine. It's a single file, no dependencies.

2. The OpenClaw Skill

Install the Plasmate skill so your agent knows how to use it:

```
clawhub install plasmate
```

Once installed, your agent automatically uses Plasmate for web browsing when it makes sense. No configuration needed.

3. The Chrome Extension (for authentication)

This is what lets you hand login cookies to your agent. Install it once:

- › Go to github.com/plasmate-labs/plasmate-extension
- › Download or clone the repo
- › Open `chrome://extensions` in Chrome
- › Turn on **Developer mode** (toggle in the top right)
- › Click **Load unpacked** and select the folder you downloaded

You'll see a small Plasmate icon appear in your Chrome toolbar.

Chrome Web Store listing is coming soon. For now, the manual install takes about 30 seconds.

`plasmate.app`

Giving your agent access to X

Let's walk through the most common use case: letting your OpenClaw agent browse X/Twitter as a logged-in account.

The idea

You log into X in Chrome like normal. The extension grabs the auth cookies. Plasmate stores them encrypted on your machine. Now your agent can browse X as that account.

1

Start the bridge server

Open a terminal and run: `plasmate auth serve`

This starts a local server that the extension talks to. Leave it running.

2

Log into X in Chrome

Go to x.com and log into the account you want your agent to use. This could be your personal account, a brand account, or an account you created specifically for your agent.

3

Click the Plasmate extension icon

A popup shows all cookies for x.com. The important ones (`ct0` and `auth_token`) are already highlighted and selected.

4

Click "Push to Plasmate"

One click. The cookies are encrypted and stored locally. The extension confirms with a green message. You're done.

That's it. Your agent can now browse X as that account. It can read timelines, check mentions, view DMs (if the cookies include that access), and interact with posts.

Cookies expire eventually. Plasmate tracks expiry dates and warns you when cookies are getting stale. When that happens, just repeat steps 2-4. Takes about 10 seconds.

`plasmate.app`

Works with everything

The same process works for any website. The extension auto-selects the right cookies for popular platforms:



X / Twitter

Read timelines, check mentions, monitor keywords, view analytics.



GitHub

Review PRs, read private repos, check notifications, browse issues.



LinkedIn

Read messages, check connection requests, monitor company pages.



Instagram

View feeds, check DMs, monitor engagement on posts.

For any other site, just navigate there in Chrome, click the extension, select the cookies that look like session or auth tokens, and push. If you're not sure which cookies matter, select all of them.

Managing your profiles

Check which sites your agent can access:

```
plasmate auth list
# ✓ x.com (2 cookies) - valid
# ✓ github.com (2 cookies) - valid
# △ linkedin.com (2 cookies) - expires soon
```

Remove access for a site:

```
plasmate auth revoke linkedin.com
```


Security

- › All cookies are encrypted with AES-256-GCM on disk
- › The encryption key is auto-generated and stored with strict file permissions
- › The bridge server only listens on localhost (127.0.0.1), never the network
- › Expired cookies are automatically skipped
- › You can revoke any profile instantly

`plasmate.app`

What your agent can do now

Once Plasmate is installed and you've pushed cookies for the sites you care about, your OpenClaw agent can do things like:

- › "Check my X mentions from the last 24 hours"
- › "Read this GitHub PR and summarize the review comments"
- › "Go to this LinkedIn post and tell me what people are saying"
- › "Check if anyone replied to my tweet about the product launch"
- › "Browse this private repo and find the config file"
- › "Scrape this page that requires a login"

The agent uses Plasmate automatically through the skill. You don't need to tell it "use Plasmate" or configure anything per-request. It just works.

What makes this different from regular browsing?

Regular browser tool

Launches Chrome, renders pixels, takes screenshots, uses OCR to read them. Slow, heavy, can't log in.

Plasmate

Compiles HTML directly into structured data. Understands forms, buttons, links, and content natively. Fast, light, authenticated.

Quick reference

```
# Install
curl -sSf https://plasmate.app/install.sh | sh
clawhub install plasmate
```

```
# Start the bridge (leave running)
plasmate auth serve
```

```
# Push cookies (from Chrome extension)
```

```
# Navigate to site > Click extension > Push to Plasmate
```

```
# Check status  
plasmate auth list
```

```
# Your agent now browses authenticated
```

Questions? Check **docs.plasmate.app** for the full documentation, or ask in the **OpenClaw Discord** community.

plasmate.app