CLI commands › browser

CLI commands

browser

Manage OpenClaw’s browser control server and run browser actions (tabs, snapshots, screenshots, navigation, clicks, typing).

Related:

- Browser tool + API: Browser tool
- Chrome extension relay: Chrome extension

Common flags

- `--url <gatewayWsUrl>` : Gateway WebSocket URL (defaults to config).
- `--token <token>` : Gateway token (if required).
- `--timeout <ms>` : request timeout (ms).
- `--browser-profile <name>` : choose a browser profile (default from config).
- `--json` : machine-readable output (where supported).

Quick start (local)

```
openclaw browser --browser-profile chrome tabs
openclaw browser --browser-profile openclaw start
openclaw browser --browser-profile openclaw open https://example.com
openclaw browser --browser-profile openclaw snapshot
```

Profiles

Profiles are named browser routing configs. In practice:

- `openclaw` : launches/attaches to a dedicated OpenClaw-managed Chrome instance (isolated user data dir).
- `chrome` : controls your existing Chrome tab(s) via the Chrome extension relay.

```
openclaw browser profiles
openclaw browser create-profile --name work --color "#FF5A36"
openclaw browser delete-profile --name work
```

Use a specific profile:

```
openclaw browser --browser-profile work tabs
```

Tabs

```
openclaw browser tabs
openclaw browser open https://docs.openclaw.ai
openclaw browser focus <targetId>
openclaw browser close <targetId>
```

Snapshot / screenshot / actions

Snapshot:

```
openclaw browser snapshot
```

Screenshot:

```
openclaw browser screenshot
```

Navigate/click/type (ref-based UI automation):

```
openclaw browser navigate https://example.com
openclaw browser click <ref>
openclaw browser type <ref> "hello"
```

Chrome extension relay (attach via toolbar button)

This mode lets the agent control an existing Chrome tab that you attach manually (it does not auto-attach).

Install the unpacked extension to a stable path:

```
openclaw browser extension install
openclaw browser extension path
```

Then Chrome → `chrome://extensions` → enable “Developer mode” → “Load unpacked” → select the printed folder.

Full guide:

Remote browser control (node host proxy)

If the Gateway runs on a different machine than the browser, run a **node host** on the machine that has Chrome/Brave/Edge/Chromium. The Gateway will proxy browser actions to that node (no separate browser control server required).

Use `gateway.nodes.browser.mode` to control auto-routing and `gateway.nodes.browser.node` to pin a specific node if multiple are connected.

Security + remote setup: **Browser tool**, **Remote access**, **Tailscale**, **Security**

‹ approvals channels ›