



☰ CLI > **models**

CLI

models

Model discovery, scanning, and configuration (default model, fallbacks, auth profiles).

Related:

Providers + models: [Models](#)

Provider auth setup: [Getting started](#)

Common commands

```
openclaw models status
openclaw models list
openclaw models set <model-or-alias>
openclaw models scan
```



`openclaw models status` shows the resolved default/fallbacks plus an auth overview. When provider usage snapshots are available, the OAuth/token status section includes provider usage headers. Add `--probe` to run live auth probes against each configured provider profile. Probes are real requests (may consume tokens and trigger rate limits). Use `--agent <id>` to inspect a configured agent's model/auth state. When omitted, the command uses `OPENCLAW_AGENT_DIR / PI_CODING_AGENT_DIR` if set, otherwise the configured default agent.

Notes:

`models set <model-or-alias>` accepts provider/model or an alias.

 Model refs are parsed by splitting on the **first /**. If the model ID includes **/** (OpenRouter-style), include the provider prefix (example: `openrouter/moonshotai/kimi-k2`).

› If you omit the provider, OpenClaw treats the input as an alias or a model for the **default provider** (only works when there is no **/** in the model ID).

models status

Options:

```
--json
--plain
--check (exit 1=expired/missing, 2=expiring)
--probe (live probe of configured auth profiles)
--probe-provider <name> (probe one provider)
--probe-profile <id> (repeat or comma-separated profile ids)
--probe-timeout <ms>
--probe-concurrency <n>
--probe-max-tokens <n>
--agent <id> (configured agent id; overrides
OPENCLAW_AGENT_DIR / PI_CODING_AGENT_DIR )
```

Aliases + fallbacks

```
openclaw models aliases list
openclaw models fallbacks list
```



Auth profiles

```
openclaw models auth add
openclaw models auth login --provider <id>
openclaw models auth setup-token
openclaw models auth paste-token
```



`models auth login` runs a provider plugin's auth flow (OAuth/API key). Use `openclaw plugins list` to see which providers are installed.

Notes:

`setup-token` prompts for a setup-token value (generate it with `claude setup-token` on any machine).

`paste-token` accepts a token string generated elsewhere or from automation.

< memory

logs >

Powered by mintlify