Configuration and operations › Authentication

Configuration and operations

Authentication

OpenClaw supports OAuth and API keys for model providers. For Anthropic accounts, we recommend using an **API key**. For Claude subscription access, use the long-lived token created by `claude setup-token` .

See [/concepts/oauth]() for the full OAuth flow and storage layout.

Recommended Anthropic setup (API key)

If you’re using Anthropic directly, use an API key.

1. Create an API key in the Anthropic Console.
2. Put it on the **gateway host** (the machine running `openclaw gateway`).

```
export ANTHROPIC_API_KEY="..."
openclaw models status
```

3. If the Gateway runs under systemd/launchd, prefer putting the key in `~/.openclaw/.env` so the daemon can read it:

```
cat >> ~/.openclaw/.env <<'EOF'
ANTHROPIC_API_KEY=...
EOF
```

Then restart the daemon (or restart your Gateway process) and re-check:

```
openclaw models status
openclaw doctor
```

If you'd rather not manage env vars yourself, the onboarding wizard can store API keys for daemon use: `openclaw onboard`.

See for details on env inheritance (`env.shellEnv` , `~/.openclaw/.env` , systemd/launchd).

Anthropic: setup-token (subscription auth)

For Anthropic, the recommended path is an **API key**. If you're using a Claude subscription, the setup-token flow is also supported. Run it on the **gateway host**:

```
claude setup-token
```

Then paste it into OpenClaw:

```
openclaw models auth setup-token --provider anthropic
```

If the token was created on another machine, paste it manually:

```
openclaw models auth paste-token --provider anthropic
```

If you see an Anthropic error like:

```
This credential is only authorized for use with Claude Code and cannot
```

…use an Anthropic API key instead.

Manual token entry (any provider; writes `auth-profiles.json` + updates config):

```
openclaw models auth paste-token --provider anthropic
openclaw models auth paste-token --provider openrouter
```

Automation-friendly check (exit `1` when expired/missing, `2` when expiring):

```
openclaw models status --check
```

Optional ops scripts (systemd/Termux) are documented here:

> `claude setup-token` requires an interactive TTY.

Checking model auth status

```
openclaw models status
openclaw doctor
```

API key rotation behavior (gateway)

Some providers support retrying a request with alternative keys when an API call hits a provider rate limit.

Priority order:

- `OPENCLAW_LIVE_<PROVIDER>_KEY` (single override)
- `<PROVIDER>_API_KEYS`
- `<PROVIDER>_API_KEY`
- `<PROVIDER>_API_KEY_*`

Google providers also include `GOOGLE_API_KEY` as an additional fallback.

The same key list is deduplicated before use.

OpenClaw retries with the next key only for rate-limit errors (for example `429`, `rate_limit`, `quota`, `resource exhausted`).

Non-rate-limit errors are not retried with alternate keys.

If all keys fail, the final error from the last attempt is returned.

Controlling which credential is used

Per-session (chat command)

Use `/model <alias-or-id>@<profileId>` to pin a specific provider credential for the current session (example profile ids: `anthropic:default`, `anthropic:work`).

Use `/model` (or `/model list`) for a compact picker; use `/model status` for the full view (candidates + next auth profile, plus provider endpoint details when configured).

Per-agent (CLI override)

Set an explicit auth profile order override for an agent (stored in that agent's `auth-profiles.json`):

```
openclaw models auth order get --provider anthropic
openclaw models auth order set --provider anthropic anthropic:default
openclaw models auth order clear --provider anthropic
```

Use `--agent <id>` to target a specific agent; omit it to use the configured default agent.

Troubleshooting

“No credentials found”

If the Anthropic token profile is missing, run `claude setup-token` on the **gateway host**, then re-check:

```
openclaw models status
```

Token expiring/expired

Run `openclaw models status` to confirm which profile is expiring. If the profile is missing, rerun `claude setup-token` and paste the token again.

Requirements

- Claude Max or Pro subscription (for `claude setup-token`)
- Claude Code CLI installed (`claude` command available)

‹ Configuration Examples

Trusted proxy auth ›

Powered by mintlify