Skills › ClawHub

Skills

ClawHub

ClawHub is the **public skill registry for OpenClaw**. It is a free service: all skills are public, open, and visible to everyone for sharing and reuse. A skill is just a folder with a `SKILL.md` file (plus supporting text files). You can browse skills in the web app or use the CLI to search, install, update, and publish skills.

Site: clawhub.ai

What ClawHub is

- A public registry for OpenClaw skills.
- A versioned store of skill bundles and metadata.
- A discovery surface for search, tags, and usage signals.

How it works

1. A user publishes a skill bundle (files + metadata).
2. ClawHub stores the bundle, parses metadata, and assigns a version.
3. The registry indexes the skill for search and discovery.
4. Users browse, download, and install skills in OpenClaw.

What you can do

- Publish new skills and new versions of existing skills.
- Discover skills by name, tags, or search.
- Download skill bundles and inspect their files.
- Report skills that are abusive or unsafe.
- If you are a moderator, hide, unhide, delete, or ban.

Who this is for (beginner-friendly)

If you want to add new capabilities to your OpenClaw agent, ClawHub is the easiest way to find and install skills. You do not need to know how the backend works. You can:

- Search for skills by plain language.
- Install a skill into your workspace.
- Update skills later with one command.
- Back up your own skills by publishing them.

Quick start (non-technical)

1. Install the CLI (see next section).
2. Search for something you need:

   ```
   clawhub search "calendar"
   ```

3. Install a skill:

   ```
   clawhub install <skill-slug>
   ```

4. Start a new OpenClaw session so it picks up the new skill.

Install the CLI

Pick one:

```
npm i -g clawhub
```

```
pnpm add -g clawhub
```

How it fits into OpenClaw

By default, the CLI installs skills into `./skills` under your current working directory. If a OpenClaw workspace is configured, `clawhub` falls back to that workspace unless you override `--workdir` (or `CLAWHUB_WORKDIR`). OpenClaw loads workspace skills from `<workspace>/skills` and will pick them up in the **next** session. If you already use `~/.openclaw/skills` or bundled skills, workspace skills take precedence.

For more detail on how skills are loaded, shared, and gated, see .

Skill system overview

A skill is a versioned bundle of files that teaches OpenClaw how to perform a specific task. Each publish creates a new version, and the registry keeps a history of versions so users can audit changes.

A typical skill includes:

- A `SKILL.md` file with the primary description and usage.
- Optional configs, scripts, or supporting files used by the skill.
- Metadata such as tags, summary, and install requirements.

ClawHub uses metadata to power discovery and safely expose skill capabilities. The registry also tracks usage signals (such as stars and downloads) to improve ranking and visibility.

What the service provides (features)

- **Public browsing** of skills and their `SKILL.md` content.
- **Search** powered by embeddings (vector search), not just keywords.
- **Versioning** with semver, changelogs, and tags (including `latest`).
- **Downloads** as a zip per version.
- **Stars and comments** for community feedback.
- **Moderation** hooks for approvals and audits.
- **CLI-friendly API** for automation and scripting.

Security and moderation

ClawHub is open by default. Anyone can upload skills, but a GitHub account must be at least one week old to publish. This helps slow down abuse without blocking legitimate contributors.

Reporting and moderation:

- Any signed in user can report a skill.
- Report reasons are required and recorded.
- Each user can have up to 20 active reports at a time.
- Skills with more than 3 unique reports are auto hidden by default.
- Moderators can view hidden skills, unhide them, delete them, or ban users.
- Abusing the report feature can result in account bans.

Interested in becoming a moderator? Ask in the OpenClaw Discord and contact a moderator or maintainer.

CLI commands and parameters

Global options (apply to all commands):

- `--workdir <dir>` : Working directory (default: current dir; falls back to OpenClaw workspace).
- `--dir <dir>` : Skills directory, relative to workdir (default: `skills`).
- `--site <url>` : Site base URL (browser login).
- `--registry <url>` : Registry API base URL.
- `--no-input` : Disable prompts (non-interactive).
- `-V, --cli-version` : Print CLI version.

Auth:

- `clawhub login` (browser flow) or `clawhub login --token <token>`
- `clawhub logout`
- `clawhub whoami`

Options:

- `--token <token>` : Paste an API token.
- `--label <label>` : Label stored for browser login tokens (default: `CLI token`).
- `--no-browser` : Do not open a browser (requires `--token`).

Search:

- `clawhub search "query"`
- `--limit <n>` : Max results.

Install:

- `clawhub install <slug>`
- `--version <version>` : Install a specific version.
- `--force` : Overwrite if the folder already exists.

Update:

- `clawhub update <slug>`
- `clawhub update --all`
- `--version <version>` : Update to a specific version (single slug only).
- `--force` : Overwrite when local files do not match any published version.

List:

- `clawhub list` (reads `.clawhub/lock.json`)

Publish:

- `clawhub publish <path>`
- `--slug <slug>` : Skill slug.
- `--name <name>` : Display name.
- `--version <version>` : Semver version.
- `--changelog <text>` : Changelog text (can be empty).
- `--tags <tags>` : Comma-separated tags (default: `latest`).

Delete/undelete (owner/admin only):

- `clawhub delete <slug> --yes`
- `clawhub undelete <slug> --yes`

Sync (scan local skills + publish new/updated):

- `clawhub sync`
- `--root <dir...>` : Extra scan roots.
- `--all` : Upload everything without prompts.
- `--dry-run` : Show what would be uploaded.
- `--bump <type>` : `patch|minor|major` for updates (default: `patch`).
- `--changelog <text>` : Changelog for non-interactive updates.

- `--tags <tags>` : Comma-separated tags (default: `latest`).
- `--concurrency <n>` : Registry checks (default: 4).

›

Common workflows for agents

Search for skills

```
clawhub search "postgres backups"
```

Download new skills

```
clawhub install my-skill-pack
```

Update installed skills

```
clawhub update --all
```

Back up your skills (publish or sync)

For a single skill folder:

```
clawhub publish ./my-skill --slug my-skill --name "My Skill" --version
```

To scan and back up many skills at once:

```
clawhub sync --all
```

Advanced details (technical)

Versioning and tags

- Each publish creates a new **semver** `SkillVersion` .
- Tags (like `latest`) point to a version; moving tags lets you roll back.
- Changelogs are attached per version and can be empty when syncing or publishing updates.

Local changes vs registry versions

Updates compare the local skill contents to registry versions using a content hash. If local files do not match any published version, the CLI asks before overwriting (or requires `--force` in non-interactive runs).

Sync scanning and fallback roots

`clawhub sync` scans your current workdir first. If no skills are found, it falls back to known legacy locations (for example `~/openclaw/skills` and `~/.openclaw/skills`). This is designed to find older skill installs without extra flags.

Storage and lockfile

- Installed skills are recorded in `.clawhub/lock.json` under your workdir.
- Auth tokens are stored in the ClawHub CLI config file (override via `CLAWHUB_CONFIG_PATH`).

Telemetry (install counts)

When you run `clawhub sync` while logged in, the CLI sends a minimal snapshot to compute install counts. You can disable this entirely:

```
export CLAWHUB_DISABLE_TELEMETRY=1
```

Environment variables

- `CLAWHUB_SITE` : Override the site URL.
- `CLAWHUB_REGISTRY` : Override the registry API URL.
- `CLAWHUB_CONFIG_PATH` : Override where the CLI stores the token/config.
- `CLAWHUB_WORKDIR` : Override the default workdir.
- `CLAWHUB_DISABLE_TELEMETRY=1` : Disable telemetry on `sync` .

‹ Skills Config Plugins ›