CLI commands › hooks

CLI commands

hooks

Manage agent hooks (event-driven automations for commands like `/new` , `/reset` , and gateway startup).

Related:

- Hooks: **Hooks**
- Plugin hooks: **Plugins**

List All Hooks

```
openclaw hooks list
```

List all discovered hooks from workspace, managed, and bundled directories.

Options:

- `--eligible` : Show only eligible hooks (requirements met)
- `--json` : Output as JSON
- `-v, --verbose` : Show detailed information including missing requirements

Example output:

```
Hooks (4/4 ready)

Ready:
  🚀 boot-md ✓ - Run BOOT.md on gateway startup
  📎 bootstrap-extra-files ✓ - Inject extra workspace bootstrap files during agent
  📝 command-logger ✓ - Log all command events to a centralized audit file
  💾 session-memory ✓ - Save session context to memory when /new command is issue
```

Example (verbose):

```
openclaw hooks list --verbose
```

Shows missing requirements for ineligible hooks.

Example (JSON):

```
openclaw hooks list --json
```

Returns structured JSON for programmatic use.

Get Hook Information

```
openclaw hooks info <name>
```

Show detailed information about a specific hook.

Arguments:

- `<name>` : Hook name (e.g., `session-memory`)

Options:

- `--json` : Output as JSON

Example:

```
openclaw hooks info session-memory
```

Output:

```
💾 session-memory ✓ Ready

Save session context to memory when /new command is issued

Details:
  Source: openclaw-bundled
  Path: /path/to/openclaw/hooks/bundled/session-memory/HOOK.md
  Handler: /path/to/openclaw/hooks/bundled/session-memory/handler.ts
  Homepage: https://docs.openclaw.ai/automation/hooks#session-memory
  Events: command:new

Requirements:
  Config: ✓ workspace.dir
```

Check Hooks Eligibility

```
openclaw hooks check
```

Show summary of hook eligibility status (how many are ready vs. not ready).

Options:

- `--json` : Output as JSON

Example output:

```
Hooks Status

Total hooks: 4
Ready: 4
Not ready: 0
```

Enable a Hook

```
openclaw hooks enable <name>
```

Enable a specific hook by adding it to your config (`~/.openclaw/config.json`).

Note: Hooks managed by plugins show `plugin:<id>` in `openclaw hooks list` and can’t be enabled/disabled here. Enable/disable the plugin instead.

Arguments:

- `<name>` : Hook name (e.g., `session-memory`)

Example:

```
openclaw hooks enable session-memory
```

Output:

```
✓ Enabled hook: 💾 session-memory
```

What it does:

- Checks if hook exists and is eligible
- Updates `hooks.internal.entries.<name>.enabled = true` in your config

- Saves config to disk

After enabling:

- Restart the gateway so hooks reload (menu bar app restart on macOS, or restart your gateway process in dev).

Disable a Hook

```
openclaw hooks disable <name>
```

Disable a specific hook by updating your config.

Arguments:

- `<name>` : Hook name (e.g., `command-logger`)

Example:

```
openclaw hooks disable command-logger
```

Output:

```
⏸ Disabled hook: 📝 command-logger
```

After disabling:

- Restart the gateway so hooks reload

Install Hooks

```
openclaw hooks install <path-or-spec>
```

Install a hook pack from a local folder/archive or npm.

Npm specs are **registry-only** (package name + optional version/tag). Git/URL/file specs are rejected. Dependency installs run with `--ignore-scripts` for safety.

What it does:

- Copies the hook pack into `~/.openclaw/hooks/<id>`
- Enables the installed hooks in `hooks.internal.entries.*`
- Records the install under `hooks.internal.installs`

Options:

- `-l, --link` : Link a local directory instead of copying (adds it to `hooks.internal.load.extraDirs`)

Supported archives: `.zip` , `.tgz` , `.tar.gz` , `.tar`

Examples:

```
# Local directory
openclaw hooks install ./my-hook-pack

# Local archive
openclaw hooks install ./my-hook-pack.zip

# NPM package
openclaw hooks install @openclaw/my-hook-pack

# Link a local directory without copying
openclaw hooks install -l ./my-hook-pack
```

Update Hooks

```
openclaw hooks update <id>
openclaw hooks update --all
```

Update installed hook packs (npm installs only).

Options:

- `--all` : Update all tracked hook packs
- `--dry-run` : Show what would change without writing

Bundled Hooks

session-memory

Saves session context to memory when you issue `/new` .

Enable:

```
openclaw hooks enable session-memory
```

Output: `~/.openclaw/workspace/memory/YYYY-MM-DD-slug.md`

See:

bootstrap-extra-files

Injects additional bootstrap files (for example monorepo-local `AGENTS.md` / `TOOLS.md`) during `agent:bootstrap` .

Enable:

```
openclaw hooks enable bootstrap-extra-files
```

See: bootstrap-extra-files documentation

command-logger

Logs all command events to a centralized audit file.

Enable:

```
openclaw hooks enable command-logger
```

Output: `~/.openclaw/logs/commands.log`

View logs:

```
# Recent commands
tail -n 20 ~/.openclaw/logs/commands.log

# Pretty-print
cat ~/.openclaw/logs/commands.log | jq .

# Filter by action
grep '"action":"new"' ~/.openclaw/logs/commands.log | jq .
```

See:

boot-md

Runs `BOOT.md` when the gateway starts (after channels start).

Events: `gateway:startup`

Enable:

```
openclaw hooks enable boot-md
```

See: boot-md documentation

‹ health ›

logs ›