Automation

Webhooks

Gateway can expose a small HTTP webhook endpoint for external triggers.

Enable

```
{
  hooks: {
    enabled: true,
    token: "shared-secret",
    path: "/hooks",
    // Optional: restrict explicit `agentId` routing to this allowlist.
    // Omit or include "*" to allow any agent.
    // Set [] to deny all explicit `agentId` routing.
    allowedAgentIds: ["hooks", "main"],
  },
}
```

Notes:

- `hooks.token` is required when `hooks.enabled=true`.
- `hooks.path` defaults to `/hooks`.

Auth

Every request must include the hook token. Prefer headers:

- `Authorization: Bearer <token>` (recommended)
- `x-openclaw-token: <token>`

Query-string tokens are rejected (`?token=...` returns `400`).

Endpoints

POST /hooks/wake

Payload:

```
{ "text": "System line", "mode": "now" }
```

- `text` **required** (string): The description of the event (e.g., “New email received”).
- `mode` optional (`now` | `next-heartbeat`): Whether to trigger an immediate heartbeat (default `now`) or wait for the next periodic check.

Effect:

- Enqueues a system event for the **main** session
- If `mode=now` , triggers an immediate heartbeat

POST /hooks/agent

Payload:

```
{
  "message": "Run this",
  "name": "Email",
  "agentId": "hooks",
  "sessionKey": "hook:email:msg-123",
  "wakeMode": "now",
  "deliver": true,
  "channel": "last",
  "to": "+15551234567",
  "model": "openai/gpt-5.2-mini",
  "thinking": "low",
  "timeoutSeconds": 120
}
```

`message` **required** (string): The prompt or message for the agent to process.

`name` optional (string): Human-readable name for the hook (e.g., “GitHub”), used as a prefix in session summaries.

`agentId` optional (string): Route this hook to a specific agent. Unknown IDs fall back to the default agent. When set, the hook runs using the resolved agent’s workspace and configuration.

`sessionKey` optional (string): The key used to identify the agent’s session. By default this field is rejected unless `hooks.allowRequestSessionKey=true` .

`wakeMode` optional (`now` | `next-heartbeat`): Whether to trigger an immediate heartbeat (default `now`) or wait for the next periodic check.

`deliver` optional (boolean): If `true` , the agent’s response will be sent to the messaging channel. Defaults to `true` . Responses that are only heartbeat acknowledgments are automatically skipped.

`channel` optional (string): The messaging channel for delivery. One of: `last` , `whatsapp` , `telegram` , `discord` , `slack` , `mattermost` (plugin), `signal` , `imessage` , `msteams` . Defaults to `last` .

`to` optional (string): The recipient identifier for the channel (e.g., phone number for WhatsApp/Signal, chat ID for Telegram, channel ID for Discord/Slack/Mattermost (plugin), conversation ID for MS Teams). Defaults to the last recipient in the main session.

`model` optional (string): Model override (e.g., `anthropic/claude-3-5-sonnet` or an alias). Must be in the allowed model list if restricted.

`thinking` optional (string): Thinking level override (e.g., `low` , `medium` , `high`).

`timeoutSeconds` optional (number): Maximum duration for the agent run in seconds.

Effect:

Runs an **isolated** agent turn (own session key)

Always posts a summary into the **main** session

If `wakeMode=now` , triggers an immediate heartbeat

Session key policy (breaking change)

`/hooks/agent` payload `sessionKey` overrides are disabled by default.

Recommended: set a fixed `hooks.defaultSessionKey` and keep request overrides off.

Optional: allow request overrides only when needed, and restrict prefixes.

Recommended config:

```
{
  hooks: {
    enabled: true,
    token: "${OPENCLAW_HOOKS_TOKEN}",
    defaultSessionKey: "hook:ingress",
    allowRequestSessionKey: false,
    allowedSessionKeyPrefixes: ["hook:"],
  },
}
```

Compatibility config (legacy behavior):

```
{
  hooks: {
    enabled: true,
    token: "${OPENCLAW_HOOKS_TOKEN}",
    allowRequestSessionKey: true,
    allowedSessionKeyPrefixes: ["hook:"], // strongly recommended
  },
}
```

POST /hooks/<name> (mapped)

Custom hook names are resolved via `hooks.mappings` (see configuration). A mapping can turn arbitrary payloads into `wake` or `agent` actions, with optional templates or code transforms.

Mapping options (summary):

- `hooks.presets: ["gmail"]` enables the built-in Gmail mapping.
- `hooks.mappings` lets you define `match`, `action`, and templates in config.
- `hooks.transformsDir` + `transform.module` loads a JS/TS module for custom logic.

`hooks.transformsDir` (if set) must stay within the transforms root under your OpenClaw config directory (typically `~/.openclaw/hooks/transforms`).

`transform.module` must resolve within the effective transforms directory (traversal/escape paths are rejected).

Use `match.source` to keep a generic ingest endpoint (payload-driven routing).

TS transforms require a TS loader (e.g. `bun` or `tsx`) or precompiled `.js` at runtime.

Set `deliver: true` + `channel` / `to` on mappings to route replies to a chat surface (`channel` defaults to `last` and falls back to WhatsApp).

`agentId` routes the hook to a specific agent; unknown IDs fall back to the default agent.

`hooks.allowedAgentIds` restricts explicit `agentId` routing. Omit it (or include `*`) to allow any agent. Set `[]` to deny explicit `agentId` routing.

`hooks.defaultSessionKey` sets the default session for hook agent runs when no explicit key is provided.

`hooks.allowRequestSessionKey` controls whether `/hooks/agent` payloads may set `sessionKey` (default: `false`).

`hooks.allowedSessionKeyPrefixes` optionally restricts explicit `sessionKey` values from request payloads and mappings.

`allowUnsafeExternalContent: true` disables the external content safety wrapper for that hook (dangerous; only for trusted internal sources).

`openclaw webhooks gmail setup` writes `hooks.gmail` config for `openclaw webhooks gmail run`. See **Gmail Pub/Sub** for the full Gmail watch flow.

Responses

`200` for `/hooks/wake`

`202` for `/hooks/agent` (async run started)

`401` on auth failure

`429` after repeated auth failures from the same client (check `Retry-After`)

`400` on invalid payload

`413` on oversized payloads

Examples

```
curl -X POST http://127.0.0.1:18789/hooks/wake \
  -H 'Authorization: Bearer SECRET' \
  -H 'Content-Type: application/json' \
  -d '{"text":"New email received","mode":"now"}'
```

```
curl -X POST http://127.0.0.1:18789/hooks/agent \
  -H 'x-openclaw-token: SECRET' \
  -H 'Content-Type: application/json' \
  -d '{"message":"Summarize inbox","name":"Email","wakeMode":"next-heartbeat"}'
```

Use a different model

Add `model` to the agent payload (or mapping) to override the model for that run:

```
curl -X POST http://127.0.0.1:18789/hooks/agent \
  -H 'x-openclaw-token: SECRET' \
  -H 'Content-Type: application/json' \
  -d '{"message":"Summarize inbox","name":"Email","model":"openai/gpt-5.2-mini"}'
```

If you enforce `agents.defaults.models`, make sure the override model is included there.

```
curl -X POST http://127.0.0.1:18789/hooks/gmail \
  -H 'Authorization: Bearer SECRET' \
  -H 'Content-Type: application/json' \
  -d '{"source":"gmail","messages":[{"from":"Ada","subject":"Hello","snippet":"Hi
```

Security

- Keep hook endpoints behind loopback, tailnet, or trusted reverse proxy.
- Use a dedicated hook token; do not reuse gateway auth tokens.
- Repeated auth failures are rate-limited per client address to slow brute-force attempts.
- If you use multi-agent routing, set `hooks.allowedAgentIds` to limit explicit `agentId` selection.
- Keep `hooks.allowRequestSessionKey=false` unless you require caller-selected sessions.
- If you enable request `sessionKey`, restrict `hooks.allowedSessionKeyPrefixes` (for example, `["hook:"]`).
- Avoid including sensitive raw payloads in webhook logs.
- Hook payloads are treated as untrusted and wrapped with safety boundaries by default. If you must disable this for a specific hook, set `allowUnsafeExternalContent: true` in that hook's mapping (dangerous).

‹ Automation Troubleshooting Gmail PubSub ›

›