



≡ Configuration > Group Messages

Configuration

Group Messages

Goal: let Clawd sit in WhatsApp groups, wake up only when pinged, and keep that thread separate from the personal DM session.

Note: `agents.list[].groupChat.mentionPatterns` is now used by Telegram/Discord/Slack/iMessage as well; this doc focuses on WhatsApp-specific behavior. For multi-agent setups, set `agents.list[].groupChat.mentionPatterns` per agent (or use `messages.groupChat.mentionPatterns` as a global fallback).

What's implemented (2025-12-03)

Activation modes: `mention` (default) or `always`. `mention` requires a ping (real WhatsApp @-mentions via `mentionedJids`, regex patterns, or the bot's E.164 anywhere in the text). `always` wakes the agent on every message but it should reply only when it can add meaningful value; otherwise it returns the silent token `NO_REPLY`. Defaults can be set in config (`channels.whatsapp.groups`) and overridden per group via `/activation`. When `channels.whatsapp.groups` is set, it also acts as a group allowlist (include `"*"` to allow all).

Group policy: `channels.whatsapp.groupPolicy` controls whether group messages are accepted (`open|disabled|allowlist`). `allowlist` uses `channels.whatsapp.groupAllowFrom` (fallback: explicit `channels.whatsapp.allowFrom`). Default is `allowlist` (blocked until you add senders).



Per-group sessions: session keys look like `agent:`

`<agentId>:whatsapp:group:<jid>` so commands such as `/verbose on` or `/think high` (sent as standalone messages) are scoped to that group; personal DM state is untouched. Heartbeats are skipped for group threads.

Context injection: **pending-only** group messages (default 50) that *did not* trigger a run are prefixed under `[Chat messages since your last reply - for context]` , with the triggering line under `[Current message - respond to this]` . Messages already in the session are not re-injected.

Sender surfacing: every group batch now ends with `[from: Sender Name (+E164)]` so Pi knows who is speaking.

Ephemeral/view-once: we unwrap those before extracting text/mentions, so pings inside them still trigger.

Group system prompt: on the first turn of a group session (and whenever `/activation` changes the mode) we inject a short blurb into the system prompt like `You are replying inside the WhatsApp group "<subject>". Group members: Alice (+44...), Bob (+43...), ... Activation: trigger-only ... Address the specific sender noted in the message context. If metadata isn't available we still tell the agent it's a group chat.`

Config example (WhatsApp)

Add a `groupChat` block to `~/.openclaw/openclaw.json` so display-name pings work even when WhatsApp strips the visual `@` in the text body:



```
channels: {
  whatsapp: {
    groups: { >
      "*": { requireMention: true },
    },
  },
},
agents: {
  list: [
    {
      id: "main",
      groupChat: {
        historyLimit: 50,
        mentionPatterns: ["@?openclaw", "\\+?15555550123"],
      },
    },
  ],
},
}
```

Notes:

The regexes are case-insensitive; they cover a display-name ping like @openclaw and the raw number with or without + /spaces.


WhatsApp still sends canonical mentions via mentionedJids when someone taps the contact, so the number fallback is rarely needed but is a useful safety net.

Activation command (owner-only)

Use the group chat command:

```
/activation mention
```

```
/activation always
```

Only the owner number (from `channels.whatsapp.allowFrom` , or the bot's own 164 when unset) can change this. Send `/status` as a standalone message in the group to see the current activation mode.

>

How to use

1. Add your WhatsApp account (the one running OpenClaw) to the group.
2. Say `@openclaw ...` (or include the number). Only allowlisted senders can trigger it unless you set `groupPolicy: "open"` .
3. The agent prompt will include recent group context plus the trailing `[from: ...]` marker so it can address the right person.
4. Session-level directives (`/verbose on` , `/think high` , `/new` or `/reset` , `/compact`) apply only to that group's session; send them as standalone messages so they register. Your personal DM session remains independent.

Testing / verification

Manual smoke:

Send an `@openclaw` ping in the group and confirm a reply that references the sender name.

Send a second ping and verify the history block is included then cleared on the next turn.

Check gateway logs (run with `--verbose`) to see `inbound web message` entries showing `from: <groupId>` and the `[from: ...]` suffix.

Known considerations

Heartbeats are intentionally skipped for groups to avoid noisy broadcasts.



Echo suppression uses the combined batch string; if you send identical text twice without mentions, only the first will get a response.

Session store entries will appear as `agent:<agentId>:whatsapp:group:<jid>` in the session store

(`~/.openclaw/agents/<agentId>/sessions/sessions.json` by default); a missing entry just means the group hasn't triggered a run yet.

Typing indicators in groups follow `agents.defaults.typingMode` (default: `message` when unmentioned).

[← Pairing](#)

[Groups →](#)

Powered by [mintlify](#)