



≡ Messaging platforms > **Mattermost**

## Messaging platforms

# Mattermost

Status: supported via plugin (bot token + WebSocket events). Channels, groups, and DMs are supported. Mattermost is a self-hostable team messaging platform; see the official site at [mattermost.com](https://mattermost.com) for product details and downloads.

## Plugin required

Mattermost ships as a plugin and is not bundled with the core install.

Install via CLI (npm registry):

```
openclaw plugins install @openclaw/mattermost
```

Local checkout (when running from a git repo):

```
openclaw plugins install ./extensions/mattermost
```

If you choose Mattermost during configure/onboarding and a git checkout is detected, OpenClaw will offer the local install path automatically.

Details:

## Quick setup



1. Install the Mattermost plugin.
2. Create a Mattermost bot account and copy the **bot token**.
3. Copy the Mattermost **base URL** (e.g., `https://chat.example.com`).
4. Configure OpenClaw and start the gateway.

Minimal config:

```
{
  channels: {
    mattermost: {
      enabled: true,
      botToken: "mm-token",
      baseUrl: "https://chat.example.com",
      dmPolicy: "pairing",
    },
  },
}
```

## Environment variables (default account)

Set these on the gateway host if you prefer env vars:

```
MATTERMOST_BOT_TOKEN=...
```

```
MATTERMOST_URL=https://chat.example.com
```

Env vars apply only to the **default** account ( `default` ). Other accounts must use config values.

## Chat modes

Mattermost responds to DMs automatically. Channel behavior is controlled by `chatmode` :



`oncall` (default): respond only when @mentioned in channels.

`onmessage` : respond to every channel message.

`onchar` : respond when a message starts with a trigger prefix.

Config example:

```
{
  channels: {
    mattermost: {
      chatmode: "onchar",
      oncharPrefixes: [ ">", "!" ],
    },
  },
}
```

Notes:

`onchar` still responds to explicit @mentions.

`channels.mattermost.requireMention` is honored for legacy configs but `chatmode` is preferred.

## Access control (DMs)

Default: `channels.mattermost.dmPolicy = "pairing"` (unknown senders get a pairing code).

Approve via:

```
openclaw pairing list mattermost
```

```
openclaw pairing approve mattermost <CODE>
```

Public DMs: `channels.mattermost.dmPolicy="open"` plus  
`channels.mattermost.allowFrom=["*"]` .

## Channels (groups)



Default: `channels.mattermost.groupPolicy = "allowlist"` (mention-gated).

Allowlist senders with `channels.mattermost.groupAllowFrom` (user IDs or `@username`).

Open channels: `channels.mattermost.groupPolicy="open"` (mention-gated).

## Targets for outbound delivery

Use these target formats with `openclaw message send` or `cron/webhooks`:

`channel:<id>` for a channel

`user:<id>` for a DM

`@username` for a DM (resolved via the Mattermost API)

Bare IDs are treated as channels.

## Reactions (message tool)

Use `message action=react` with `channel=mattermost`.

`messageId` is the Mattermost post id.

`emoji` accepts names like `thumbsup` or `:+1:` (colons are optional).

Set `remove=true` (boolean) to remove a reaction.

Reaction add/remove events are forwarded as system events to the routed agent session.

Examples:

```
message action=react channel=mattermost target=channel:<channelId> me:
message action=react channel=mattermost target=channel:<channelId> messageId=<postId>
```

Config:



`channels.mattermost.actions.reactions` : enable/disable reaction actions  
(default true).

Per-account override: `channels.mattermost.accounts.  
>  
<id>.actions.reactions` .

## Multi-account

Mattermost supports multiple accounts under `channels.mattermost.accounts` :

```
{
  channels: {
    mattermost: {
      accounts: {
        default: { name: "Primary", botToken: "mm-token", baseUrl: "https://chat.o
        alerts: { name: "Alerts", botToken: "mm-token-2", baseUrl: "https://alert
      },
    },
  },
}
```

## Troubleshooting

No replies in channels: ensure the bot is in the channel and mention it (oncall), use a trigger prefix (onchar), or set `chatmode: "onmessage"` .

Auth errors: check the bot token, base URL, and whether the account is enabled.

Multi-account issues: env vars only apply to the `default` account.

< Google Chat

Signal >



Powered by **mintlify**

>