Messaging platforms

Feishu

Feishu (Lark) is a team chat platform used by companies for messaging and collaboration. This plugin connects OpenClaw to a Feishu/Lark bot using the platform’s WebSocket event subscription so messages can be received without exposing a public webhook URL.

Plugin required

Install the Feishu plugin:

```
openclaw plugins install @openclaw/feishu
```

Local checkout (when running from a git repo):

```
openclaw plugins install ./extensions/feishu
```

Quickstart

There are two ways to add the Feishu channel:

Method 1: onboarding wizard (recommended)

If you just installed OpenClaw, run the wizard:

```
openclaw onboard
```

The wizard guides you through:

1. Creating a Feishu app and collecting credentials
2. Configuring app credentials in OpenClaw
3. Starting the gateway

✅ **After configuration**, check gateway status:

```
openclaw gateway status
openclaw logs --follow
```

Method 2: CLI setup

If you already completed initial install, add the channel via CLI:

```
openclaw channels add
```

Choose **Feishu**, then enter the App ID and App Secret.

✅ **After configuration**, manage the gateway:

```
openclaw gateway status
openclaw gateway restart
openclaw logs --follow
```

Step 1: Create a Feishu app

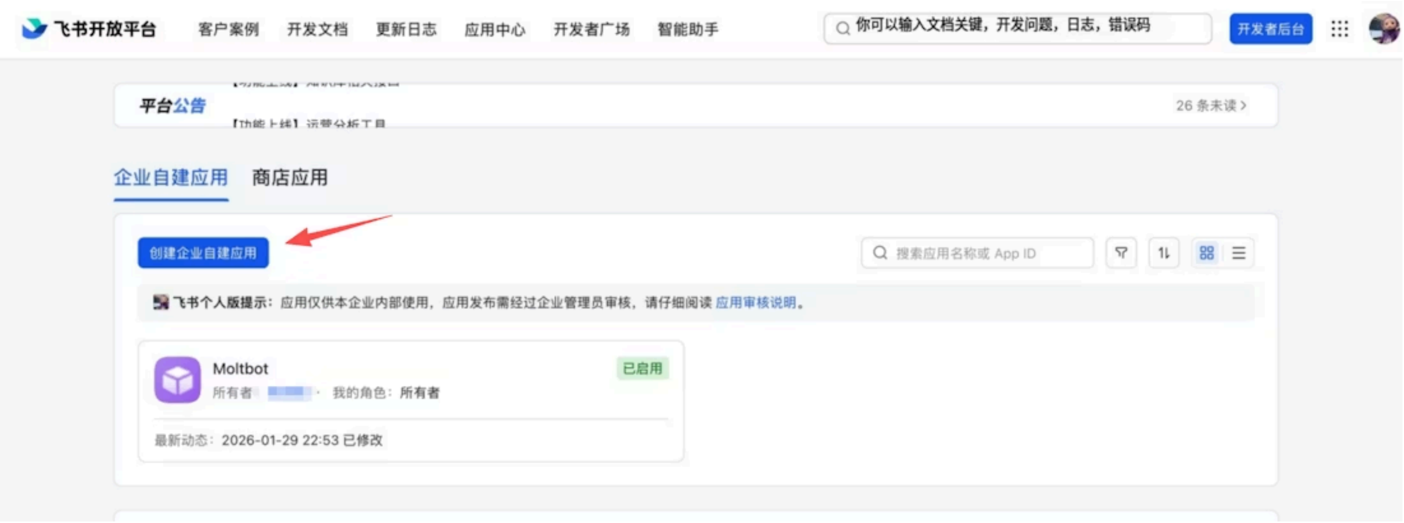
1. Open Feishu Open Platform

Visit **Feishu Open Platform** and sign in.

Lark (global) tenants should use **https://open.larksuite.com/app** and set `domain: "lark"` in the Feishu config.

2. Create an app

1. Click **Create enterprise app**
2. Fill in the app name + description
3. Choose an app icon



飞书个人版提示：应用仅供本企业内部使用，应用发布需经过企业管理员审核，请仔细阅读 应用审核说明。

所有者 [redacted] · 我的角色：所有者

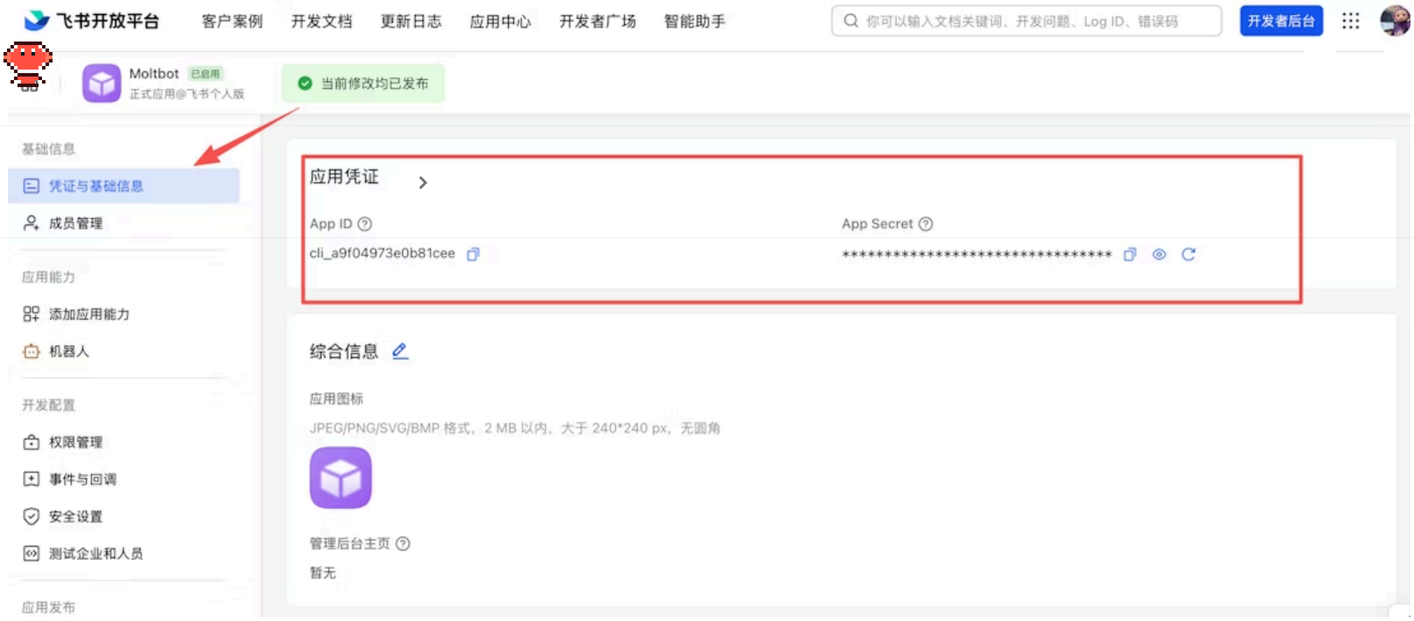
最新动态：2026-01-29 22:53 已修改

3. Copy credentials

From **Credentials & Basic Info**, copy:

- **App ID** (format: `cli_xxx`)
- **App Secret**

❗ **Important:** keep the App Secret private.



App ID: cli_a9f04973e0b81cee

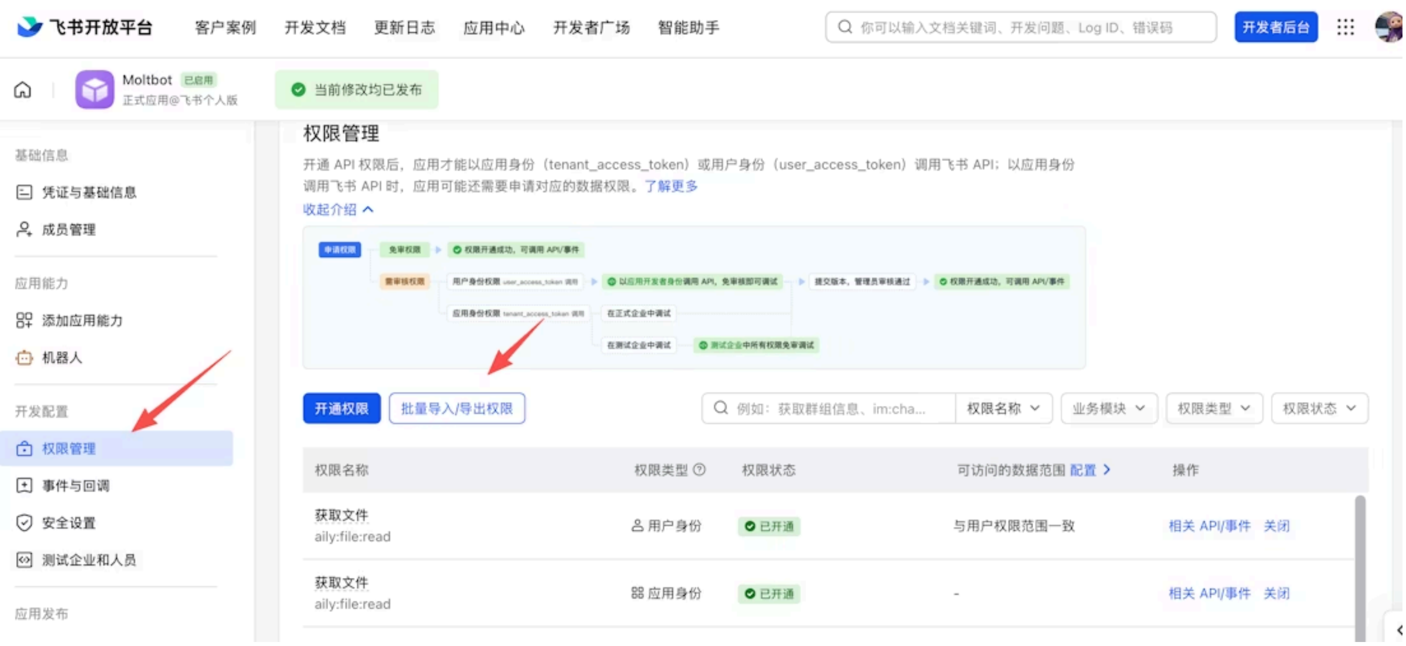
App Secret: ********************************

管理后台主页: 暂无

4. Configure permissions

On **Permissions**, click **Batch import** and paste:

```
{
  "scopes": {
    "tenant": [
      "aily:file:read",
      "aily:file:write",
      "application:application.app_message_stats.overview:readonly",
      "application:application:self_manage",
      "application:bot.menu:write",
      "contact:user.employee_id:readonly",
      "corehr:file:download",
      "event:ip_list",
      "im:chat.access_event.bot_p2p_chat:read",
      "im:chat.members:bot_access",
      "im:message",
      "im:message.group_at_msg:readonly",
      "im:message.p2p_msg:readonly",
      "im:message:readonly",
      "im:message:send_as_bot",
      "im:resource"
    ],
    "user": ["aily:file:read", "aily:file:write", "im:chat.access_event.bot_p2p_cl
  }
}
```



权限管理

开通 API 权限后，应用才能以应用身份（tenant_access_token）或用户身份（user_access_token）调用飞书 API；以应用身份调用飞书 API 时，应用可能还需要申请对应的数据权限。了解更多

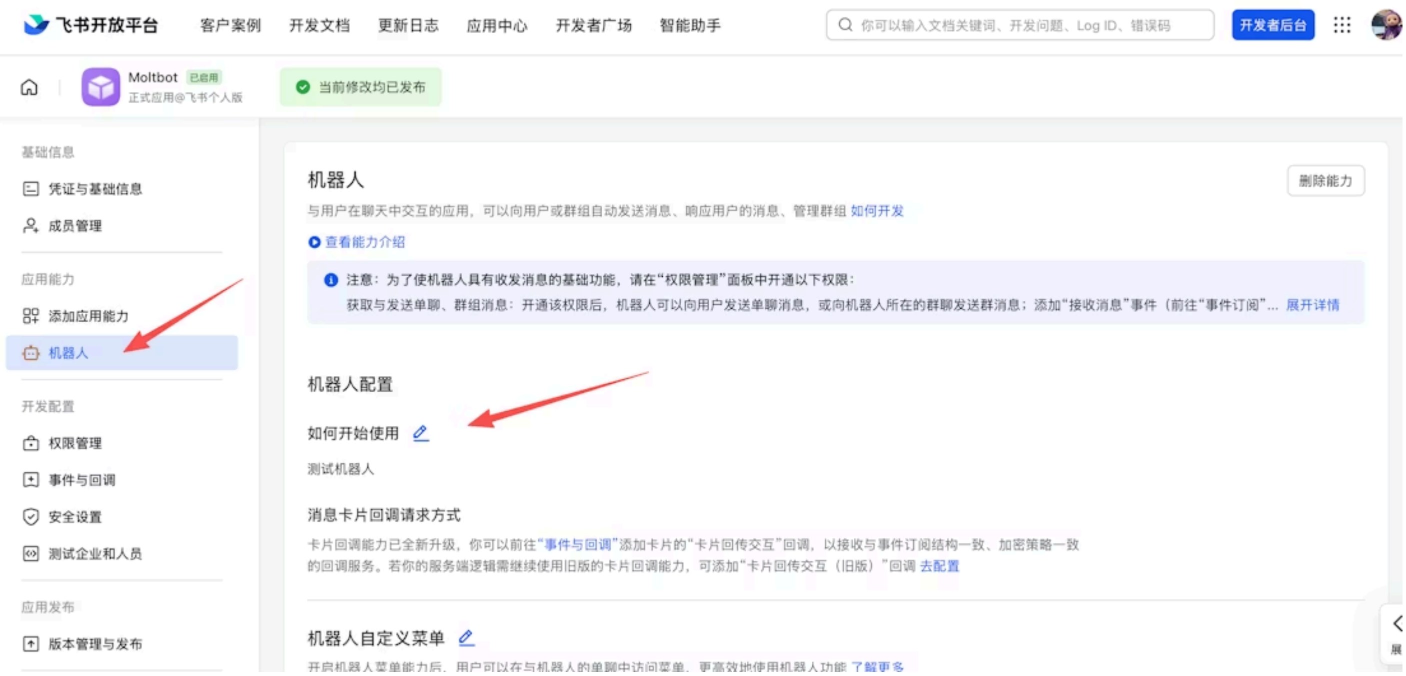
收起介绍 ^

权限名称	权限类型	权限状态	可访问的数据范围 配置 >	操作
获取文件 aily:file:read	用户身份	已开通	与用户权限范围一致	相关 API/事件 关闭
获取文件 aily:file:read	应用身份	已开通	-	相关 API/事件 关闭

5. Enable bot capability

In **App Capability** > **Bot**:

1. Enable bot capability
2. Set the bot name



机器人

与用户在聊天中交互的应用，可以向用户或群组自动发送消息、响应用户的消息、管理群组 如何开发

查看能力介绍

注意：为了使机器人具有收发消息的基础功能，请在“权限管理”面板中开通以下权限：
获取与发送单聊、群组消息：开通该权限后，机器人可以向用户发送单聊消息，或向机器人所在的群聊发送群消息；添加“接收消息”事件（前往“事件订阅”... 展开详情

机器人配置

如何开始使用

测试机器人

消息卡片回调请求方式

卡片回调能力已全新升级，你可以前往“事件与回调”添加卡片的“卡片回传交互”回调，以接收与事件订阅结构一致、加密策略一致的回调服务。若你的服务端逻辑需继续使用旧版的卡片回调能力，可添加“卡片回传交互（旧版）”回调 去配置

机器人自定义菜单

开启机器人菜单能力后，用户可以在与机器人的单聊中访问菜单，更高效地使用机器人功能 了解更多

6. Configure event subscription

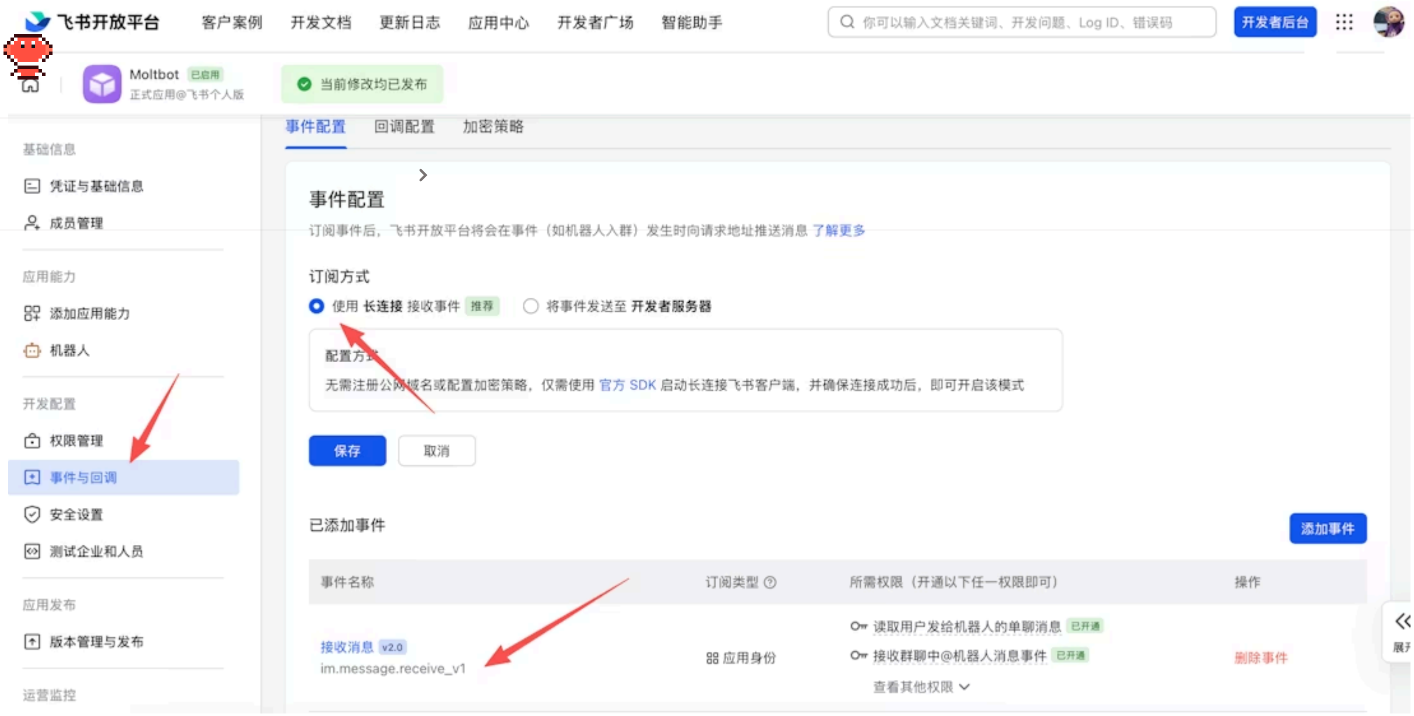
⚠ **Important:** before setting event subscription, make sure:

1. You already ran `openclaw channels add` for Feishu
2. The gateway is running (`openclaw gateway status`)

In **Event Subscription**:

1. Choose **Use long connection to receive events** (WebSocket)
2. Add the event: `im.message.receive_v1`

⚠ If the gateway is not running, the long-connection setup may fail to save.



事件配置

订阅事件后，飞书开放平台将会在事件（如机器人入群）发生时向请求地址推送消息 了解更多

订阅方式

● 使用 **长连接** 接收事件 推荐 ○ 将事件发送至 **开发者服务器**

配置方式

无需注册公网域名或配置加密策略，仅需使用 官方 SDK 启动长连接飞书客户端，并确保连接成功后，即可开启该模式

已添加事件

事件名称	订阅类型	所需权限（开通以下任一权限即可）	操作
接收消息 v2.0 im.message.receive_v1	应用身份	读取用户发给机器人的单聊消息 已开通 接收群聊中@机器人消息事件 已开通 查看其他权限	删除事件

7. Publish the app

1. Create a version in **Version Management & Release**
2. Submit for review and publish
3. Wait for admin approval (enterprise apps usually auto-approve)

Step 2: Configure OpenClaw

Configure with the wizard (recommended)

```
openclaw channels add
```

Choose **Feishu** and paste your App ID + App Secret.

Configure via config file

Edit `~/.openclaw/openclaw.json`:

```
{
  channels: {
    feishu: {
      enabled: true,
      dmPolicy: "pairing",
      accounts: {
        main: {
          appId: "cli_xxx",
          appSecret: "xxx",
          botName: "My AI assistant",
        },
      },
    },
  },
}
```

Configure via environment variables

```
export FEISHU_APP_ID="cli_xxx"
export FEISHU_APP_SECRET="xxx"
```

Lark (global) domain

If your tenant is on Lark (international), set the domain to `lark` (or a full domain string). You can set it at `channels.feishu.domain` or per account (`channels.feishu.accounts.<id>.domain`).

```
{
  channels: {
    feishu: {
      domain: "lark",
      accounts: {
        main: {
          appId: "cli_xxx",
          appSecret: "xxx",
        },
      },
    },
  },
}
```

Step 3: Start + test

1. Start the gateway

```
openclaw gateway
```

2. Send a test message

In Feishu, find your bot and send a message.

3. Approve pairing

By default, the bot replies with a pairing code. Approve it:

```
openclaw pairing approve feishu <CODE>
```

After approval, you can chat normally.

Overview

- **Feishu bot channel**: Feishu bot managed by the gateway
- **Deterministic routing**: replies always return to Feishu
- **Session isolation**: DMs share a main session; groups are isolated
- **WebSocket connection**: long connection via Feishu SDK, no public URL needed

Access control

Direct messages

- **Default**: `dmPolicy: "pairing"` (unknown users get a pairing code)
- **Approve pairing**:

```
openclaw pairing list feishu
openclaw pairing approve feishu <CODE>
```

- **Allowlist mode**: set `channels.feishu.allowFrom` with allowed Open IDs

Group chats

1. **Group policy** (`channels.feishu.groupPolicy`):
   - `"open"` = allow everyone in groups (default)
   - `"allowlist"` = only allow `groupAllowFrom`
   - `"disabled"` = disable group messages
2. **Mention requirement** (`channels.feishu.groups.<chat_id>.requireMention`):

```
true  = require @mention (default)

false  = respond without mentions
```

›

Group configuration examples

Allow all groups, require @mention (default)

```
{
  channels: {
    feishu: {
      groupPolicy: "open",
      // Default requireMention: true
    },
  },
}
```

Allow all groups, no @mention required

```
{
  channels: {
    feishu: {
      groups: {
        oc_xxx: { requireMention: false },
      },
    },
  },
}
```

Allow specific users in groups only

```
{
  channels: {
    feishu: {
      groupPolicy: "allowlist",
      groupAllowFrom: ["ou_xxx", "ou_yyy"],
    },
  },
}
```

Get group/user IDs

Group IDs (chat_id)

Group IDs look like `oc_xxx` .

Method 1 (recommended)

1. Start the gateway and @mention the bot in the group
2. Run `openclaw logs --follow` and look for `chat_id`

Method 2

Use the Feishu API debugger to list group chats.

User IDs (open_id)

User IDs look like `ou_xxx` .

Method 1 (recommended)

1. Start the gateway and DM the bot
2. Run `openclaw logs --follow` and look for `open_id`

Method 2

Check pairing requests for user Open IDs:

```
openclaw pairing list feishu
```

Common commands

Command	Description
`/status`	Show bot status
`/reset`	Reset the session
`/model`	Show/switch model

> Note: Feishu does not support native command menus yet, so commands must be sent as text.

Gateway management commands

Command	Description
`openclaw gateway status`	Show gateway status
`openclaw gateway install`	Install/start gateway service
`openclaw gateway stop`	Stop gateway service
`openclaw gateway restart`	Restart gateway service
`openclaw logs --follow`	Tail gateway logs

Troubleshooting

Bot does not respond in group chats

1. Ensure the bot is added to the group
2. Ensure you @mention the bot (default behavior)
3. Check `groupPolicy` is not set to `"disabled"`
4. Check logs: `openclaw logs --follow`

Bot does not receive messages

1. Ensure the app is published and approved
2. Ensure event subscription includes `im.message.receive_v1`
3. Ensure **long connection** is enabled
4. Ensure app permissions are complete
5. Ensure the gateway is running: `openclaw gateway status`
6. Check logs: `openclaw logs --follow`

App Secret leak

1. Reset the App Secret in Feishu Open Platform
2. Update the App Secret in your config
3. Restart the gateway

Message send failures

1. Ensure the app has `im:message:send_as_bot` permission
2. Ensure the app is published
3. Check logs for detailed errors

Advanced configuration

Multiple accounts

```
{
  channels: {
    feishu: {
      accounts: {
        main: {
          appId: "cli_xxx",
          appSecret: "xxx",
          botName: "Primary bot",
        },
        backup: {
          appId: "cli_yyy",
          appSecret: "yyy",
          botName: "Backup bot",
          enabled: false,
        },
      },
    },
  },
}
```

Message limits

- `textChunkLimit` : outbound text chunk size (default: 2000 chars)
- `mediaMaxMb` : media upload/download limit (default: 30MB)

Streaming

Feishu supports streaming replies via interactive cards. When enabled, the bot updates a card as it generates text.

```
{
  channels: {
    feishu: {
      streaming: true, // enable streaming card output (default true)
      blockStreaming: true, // enable block-level streaming (default true)
    },
  },
}
```

Set **`streaming: false`** to wait for the full reply before sending.

Multi-agent routing

Use **`bindings`** to route Feishu DMs or groups to different agents.

```
  agents: {
    list: [
      { id: "main" },
      {
        id: "clawd-fan",
        workspace: "/home/user/clawd-fan",
        agentDir: "/home/user/.openclaw/agents/clawd-fan/agent",
      },
      {
        id: "clawd-xi",
        workspace: "/home/user/clawd-xi",
        agentDir: "/home/user/.openclaw/agents/clawd-xi/agent",
      },
    ],
  },
  bindings: [
    {
      agentId: "main",
      match: {
        channel: "feishu",
        peer: { kind: "direct", id: "ou_xxx" },
      },
    },
    {
      agentId: "clawd-fan",
      match: {
        channel: "feishu",
        peer: { kind: "direct", id: "ou_yyy" },
      },
    },
    {
      agentId: "clawd-xi",
      match: {
        channel: "feishu",
        peer: { kind: "group", id: "oc_zzz" },
      },
    },
  ],
```

Routing fields:

- `match.channel` : `"feishu"`
- `match.peer.kind` : `"direct"` or `"group"`
- `match.peer.id` : user Open ID (`ou_xxx`) or group ID (`oc_xxx`)

See **Get group/user IDs** for lookup tips.

Configuration reference

Full configuration: **Gateway configuration**

Key options:

Setting	Description	Default
`channels.feishu.enabled`	Enable/disable channel	`true`
`channels.feishu.domain`	API domain (`feishu` or `lark`)	`feishu`
`channels.feishu.accounts.<id>.appId`	App ID	-
`channels.feishu.accounts.<id>.appSecret`	App Secret	-
`channels.feishu.accounts.<id>.domain`	Per-account API domain override	`feishu`
`channels.feishu.dmPolicy`	DM policy	`pairing`
`channels.feishu.allowFrom`	DM allowlist (open_id list)	-
`channels.feishu.groupPolicy`	Group policy	`open`

Setting	Description	Default
`channels.feishu.groupAllowFrom`	Group allowlist	-
`channels.feishu.groups.<chat_id>.requireMention`	Require @mention	`true`
`channels.feishu.groups.<chat_id>.enabled`	Enable group	`true`
`channels.feishu.textChunkLimit`	Message chunk size	`2000`
`channels.feishu.mediaMaxMb`	Media size limit	`30`
`channels.feishu.streaming`	Enable streaming card output	`true`
`channels.feishu.blockStreaming`	Enable block streaming	`true`

dmPolicy reference

Value	Behavior
`"pairing"`	**Default.** Unknown users get a pairing code; must be approved
`"allowlist"`	Only users in `allowFrom` can chat
`"open"`	Allow all users (requires `"*"` in allowFrom)
`"disabled"`	Disable DMs

Supported message types

Receive

- ✅ Text
- ✅ Rich text (post)

- ✅ Images
- ✅ Files
- ✅ Audio
- ✅ Video
- ✅ Stickers

Send

- ✅ Text
- ✅ Images
- ✅ Files
- ✅ Audio
- ⚠️ Rich text (partial support)

‹ Slack

Google Chat ›