



## ☰ Messaging platforms > **Feishu**

---

**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

The screenshot shows the Feishu Open Platform dashboard. At the top, there's a navigation bar with links like '客户案例', '开发文档', '更新日志', '应用中心', '开发者广场', and '智能助手'. Below the navigation is a search bar with placeholder text '你可以输入文档关键, 开发问题, 日志, 错误码'. On the right side of the header are developer-related links: '开发者后台', '更多', and a user profile icon. The main content area has tabs for '平台公告' and '企业自建应用'. Under '企业自建应用', there's a button labeled '创建企业自建应用' with a red arrow pointing to it. To the right of this button is a search bar for '搜索应用名称或 App ID' and some filter icons. Below the search bar is a card for an app named 'Moltbot', which is listed as '已启用' (Enabled). The card also shows the owner information '所有者' and the last update date '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.

The screenshot shows the Feishu OpenClaw developer console. At the top, there are navigation links: 飞书开放平台 (Feishu Open Platform), 客户案例 (Customer Cases), 开发文档 (Development Documentation), 更新日志 (Update Log), 应用中心 (App Center), 开发者广场 (Developer Square), and 智能助手 (Smart Assistant). A search bar at the top right contains the placeholder text: 你可以输入文档关键词、开发问题、Log ID、错误码 (You can input document keywords, development issues, Log ID, error code). On the far right, there are developer tools icons and a user profile icon.

In the center, there is a sidebar titled '基础信息' (Basic Information) containing the following items:

- 凭证与基础信息 (Selected)
- 成员管理
- 应用能力
- 添加应用能力
- 机器人
- 开发配置
- 权限管理
- 事件与回调
- 安全设置
- 测试企业和人员
- 应用发布

The main content area shows the '凭证与基础信息' (Credentials and Basic Information) section for the app 'Moltbot'. It includes:

- 应用凭证 (Application Credentials):** Shows App ID (cli\_a9f04973e0b81cee) and App Secret (redacted).
- 综合信息 (Comprehensive Information):** Shows the application icon (purple cube icon) and the management homepage status (暂无 (None)).

A red arrow points from the left towards the '凭证与基础信息' tab in the sidebar.

## 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"]
}
}

```

飞书开放平台 客户案例 开发文档 更新日志 应用中心 开发者广场 智能助手  开发者后台 ⋮

Moltbot 已启用 正式应用@飞书个人版 当前修改均已发布

**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](#)

基础信息  
凭证与基础信息  
**成员管理**  
应用能力  
添加应用能力  
机器人  
开发配置  
**权限管理**  
事件与回调  
安全设置  
测试企业和人员  
应用发布

**权限管理**

开通权限 批量导入/导出权限  权限名称 业务模块 权限类型 权限状态

权限名称 权限类型 权限状态 可访问的数据范围 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

开通权限图示：

```

graph LR
    A[开通权限] --> B[先审核]
    B --> C[权限开通成功, 可调用 API/事件]
    C --> D[用户身份权限 user_access_token 调用]
    D --> E[以应用开发者身份调用 API, 免审核即可调用]
    E --> F[提交版本, 管理员审核通过]
    F --> G[权限开通成功, 可调用 API/事件]
    G --> H[在正式企业中调试]
    H --> I[在测试企业中调试]
    I --> J[测试企业中所有权限免审查]
    
```

开通权限流程图：

- 开通权限
- 先审核
- 权限开通成功，可调用 API/事件
- 用户身份权限 user\_access\_token 调用
- 以应用开发者身份调用 API, 免审核即可调用
- 提交版本, 管理员审核通过
- 权限开通成功，可调用 API/事件
- 在正式企业中调试
- 在测试企业中调试
- 测试企业中所有权限免审查

## 5. Enable bot capability



In App Capability > Bot:

1. Enable bot capability

2. Set the bot name

The screenshot shows the Feishu Open Platform developer console interface. On the left, there's a sidebar with various sections like '基础信息' (Basic Information), '凭证与基础信息' (Certificates and Basic Information), '成员管理' (Member Management), '添加应用能力' (Add Application Ability), '机器人' (Robot) [highlighted by a red arrow], '开发配置' (Development Configuration), '权限管理' (Permission Management), '事件与回调' (Events and Callbacks), '安全设置' (Security Settings), '测试企业和人员' (Test Enterprises and Personnel), '应用发布' (Application Release), and '版本管理与发布' (Version Management and Release). The main area is titled '机器人' (Robot) and contains sections for '机器人' (Robot), '机器人配置' (Robot Configuration), '消息卡片回调请求方式' (Message Card Callback Request Method), and '机器人自定义菜单' (Robot Custom Menu). A green button at the top right says '当前修改均已发布' (All changes have been published). A search bar at the top right says '你可以输入文档关键词、开发问题、Log ID、错误码' (You can input document keywords, development issues, Log ID, error codes).

## 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.

The screenshot shows the Feishu OpenClaw configuration interface. On the left sidebar, under '事件与回调' (Events & Callbacks), a red arrow points to the '事件配置' (Event Configuration) tab. In the main content area, the '事件配置' tab is selected, showing the '事件配置' (Event Configuration) section. A red arrow points to the '订阅方式' (Subscription Method) section, which contains two options: '使用长连接接收事件' (Use long connection to receive events) (selected) and '将事件发送至开发者服务器' (Send events to developer server). Below this is a note: '无需注册公钥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' (No need to register a public key domain name or configure encryption policies, just use the official SDK to start the long connection Feishu client, and ensure the connection is successful after starting, to enable this mode). At the bottom are '保存' (Save) and '取消' (Cancel) buttons. To the right, there's a table titled '已添加事件' (Added Events) with columns: '事件名称' (Event Name), '订阅类型' (Subscription Type), '所需权限 (开通以下任一权限即可)' (Required Permissions (Enable any of the following permissions)), and '操作' (Operations). One event is listed: '接收消息 v2.0' (Receive message v2.0) with 'im.message.receive\_v1'. The '所需权限' column shows '读取用户发给机器人的单聊消息' (Read user-to-bot direct message) and '接收群聊中@机器人消息事件' (Receive group message events @bot) both as '已开通' (Enabled). There are '删除事件' (Delete event) and '查看其他权限' (View other permissions) buttons.

## 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

&gt;

**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
```

&gt;

## 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
```

&gt;

## 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
<code>channels.feishu.enabled</code>	Enable/disable channel	<code>true</code>
<code>channels.feishu.domain</code>	API domain ( <code>feishu</code> or <code>lark</code> )	<code>feishu</code>
<code>channels.feishu.accounts.&lt;id&gt;.appId</code>	App ID	-
<code>channels.feishu.accounts.&lt;id&gt;.appSecret</code>	App Secret	-
<code>channels.feishu.accounts.&lt;id&gt;.domain</code>	Per-account API domain override	<code>feishu</code>
<code>channels.feishu.dmPolicy</code>	DM policy	<code>pairing</code>
<code>channels.feishu.allowFrom</code>	DM allowlist ( <code>open_id</code> list)	-
<code>channels.feishu.groupPolicy</code>	Group policy	<code>open</code>

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"	<b>Default.</b> 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

&gt;

 Video Stickers

## Send

 Text Images Files Audio

Rich text (partial support)

[Slack](#)[Google Chat](#)Powered by [mintlify](#)