



---

≡ Messaging platforms > **LINE**

---

## Messaging platforms

# LINE

LINE connects to OpenClaw via the LINE Messaging API. The plugin runs as a webhook receiver on the gateway and uses your channel access token + channel secret for authentication.

Status: supported via plugin. Direct messages, group chats, media, locations, Flex messages, template messages, and quick replies are supported. Reactions and threads are not supported.

## Plugin required

Install the LINE plugin:

```
openclaw plugins install @openclaw/line
```

Local checkout (when running from a git repo):

```
openclaw plugins install ./extensions/line
```

## Setup

1. Create a LINE Developers account and open the Console:
2. Create (or pick) a Provider and add a **Messaging API** channel.

3. Copy the **Channel access token** and **Channel secret** from the channel settings.
4. Enable **Use webhook** in the Messaging API settings.
5. Set the webhook URL to your gateway endpoint (HTTPS required):

```
https://gateway-host/line/webhook
```

The gateway responds to LINE's webhook verification (GET) and inbound events (POST). If you need a custom path, set `channels.line.webhookPath` or `channels.line.accounts.<id>.webhookPath` and update the URL accordingly.

## Configure

Minimal config:

```
{
  channels: {
    line: {
      enabled: true,
      channelAccessToken: "LINE_CHANNEL_ACCESS_TOKEN",
      channelSecret: "LINE_CHANNEL_SECRET",
      dmPolicy: "pairing",
    },
  },
}
```

Env vars (default account only):

`LINE_CHANNEL_ACCESS_TOKEN`

`LINE_CHANNEL_SECRET`

Token/secret files:



```
channels: {
  line: {
    tokenFile: "/path/to/line-token.txt",
    secretFile: "/path/to/line-secret.txt",
  },
},
}
```

Multiple accounts:

```
{
  channels: {
    line: {
      accounts: {
        marketing: {
          channelAccessToken: "...",
          channelSecret: "...",
          webhookPath: "/line/marketing",
        },
      },
    },
  },
}
```

## Access control

Direct messages default to pairing. Unknown senders get a pairing code and their messages are ignored until approved.

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

Allowlists and policies:



`channels.line.dmPolicy : pairing | allowlist | open | disabled`

`channels.line.allowFrom : allowlisted LINE user IDs for DMs`

---

`channels.line.groupPolicy : allowlist | open | disabled`

`channels.line.groupAllowFrom : allowlisted LINE user IDs for groups`

---

Per-group overrides: `channels.line.groups.<groupId>.allowFrom`

LINE IDs are case-sensitive. Valid IDs look like:

User: `U` + 32 hex chars

Group: `C` + 32 hex chars

Room: `R` + 32 hex chars

## Message behavior

Text is chunked at 5000 characters.

Markdown formatting is stripped; code blocks and tables are converted into Flex cards when possible.

Streaming responses are buffered; LINE receives full chunks with a loading animation while the agent works.

Media downloads are capped by `channels.line.mediaMaxMb` (default 10).

## Channel data (rich messages)

Use `channelData.line` to send quick replies, locations, Flex cards, or template messages.



```

text: "Here you go",
channelData: {
  line: {
    quickReplies: ["Status", "Help"],
    location: {
      title: "Office",
      address: "123 Main St",
      latitude: 35.681236,
      longitude: 139.767125,
    },
    flexMessage: {
      altText: "Status card",
      contents: {
        /* Flex payload */
      },
    },
    templateMessage: {
      type: "confirm",
      text: "Proceed?",
      confirmLabel: "Yes",
      confirmData: "yes",
      cancelLabel: "No",
      cancelData: "no",
    },
  },
},
}

```

The LINE plugin also ships a `/card` command for Flex message presets:

```
/card info "Welcome" "Thanks for joining!"
```

## Troubleshooting



**Webhook verification fails:** ensure the webhook URL is HTTPS and the `channelSecret` matches the LINE console.

---

**No inbound events:** confirm the webhook path matches `channels.line.webhookPath` and that the gateway is reachable from LINE.

---

**Media download errors:** raise `channels.line.mediaMaxMb` if media exceeds the default limit.

---

< Microsoft Teams

Matrix >

Powered by [mintlify](#)