



≡ Configuration > **Pairing**

Configuration

Pairing

“Pairing” is OpenClaw’s explicit **owner approval** step. It is used in two places:

1. **DM pairing** (who is allowed to talk to the bot)
2. **Node pairing** (which devices/nodes are allowed to join the gateway network)

Security context: [Security](#)

1) DM pairing (inbound chat access)

When a channel is configured with DM policy `pairing`, unknown senders get a short code and their message is **not processed** until you approve.

Default DM policies are documented in: [Security](#)

Pairing codes:

8 characters, uppercase, no ambiguous chars (`001I`).

Expire after 1 hour. The bot only sends the pairing message when a new request is created (roughly once per hour per sender).

Pending DM pairing requests are capped at **3 per channel** by default; additional requests are ignored until one expires or is approved.

Approve a sender

```

openclaw pairing list telegram
openclaw pairing approve telegram <CODE>

```

>

Supported channels: telegram , whatsapp , signal , imessage , discord , slack , feishu .

Where the state lives

Stored under ~/.openclaw/credentials/ :

Pending requests: <channel>-pairing.json

Approved allowlist store: <channel>-allowFrom.json

Treat these as sensitive (they gate access to your assistant).

2) Node device pairing (iOS/Android/macOS/headless nodes)

Nodes connect to the Gateway as **devices** with `role: node` . The Gateway creates a device pairing request that must be approved.

Pair via Telegram (recommended for iOS)

If you use the `device-pair` plugin, you can do first-time device pairing entirely from Telegram:

1. In Telegram, message your bot: `/pair`
2. The bot replies with two messages: an instruction message and a separate **setup code** message (easy to copy/paste in Telegram).
3. On your phone, open the OpenClaw iOS app → Settings → Gateway.
4. Paste the setup code and connect.
5. Back in Telegram: `/pair approve`

The setup code is a base64-encoded JSON payload that contains:



`url` : the Gateway WebSocket URL (`ws://...` or `wss://...`)

`token` : a short-lived pairing token

Treat the setup code like a password while it is valid.

Approve a node device

```
openclaw devices list
openclaw devices approve <requestId>
openclaw devices reject <requestId>
```

Node pairing state storage

Stored under `~/.openclaw/devices/` :

`pending.json` (short-lived; pending requests expire)

`paired.json` (paired devices + tokens)

Notes

The legacy `node.pair.*` API (CLI: `openclaw nodes pending/approve`) is a separate gateway-owned pairing store. WS nodes still require device pairing.

Related docs

Security model + prompt injection:

Updating safely (run doctor):

Channel configs:

Telegram:

WhatsApp:



Signal: Signal

BlueBubbles (iMessage): BlueBubbles

iMessage (legacy): iMessage

Discord: Discord

Slack: Slack

< Zalo Personal

Group Messages >

Powered by [mintlify](#)