



≡ Messaging platforms > **WhatsApp**

Messaging platforms

WhatsApp

Status: production-ready via WhatsApp Web (Baileys). Gateway owns linked session(s).

Pairing

Default DM policy is pairing for unknown senders.

Channel

troubleshooting

Cross-channel diagnostics and repair playbooks.

Gateway

configuration

Full channel config patterns and examples.

Quick setup

- 1 **Configure WhatsApp access policy**



```
{  
  channels: {  
    whatsapp: {  
      dmPolicy: "pairing",  
      allowFrom: ["+15551234567"],  
      groupPolicy: "allowlist",  
      groupAllowFrom: ["+15551234567"],  
    },  
  },  
}
```

2 Link WhatsApp (QR)

```
openclaw channels login --channel whatsapp
```

For a specific account:

```
openclaw channels login --channel whatsapp --account work
```

3 Start the gateway

```
openclaw gateway
```

4 Approve first pairing request (if using pairing mode)

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



Pairing requests expire after 1 hour. Pending requests are capped at 3 per channel.

>

- ❗ OpenClaw recommends running WhatsApp on a separate number when possible. (The channel metadata and onboarding flow are optimized for that setup, but personal-number setups are also supported.)

Deployment patterns

Dedicated number (recommended)

Personal-number fallback

WhatsApp Web-only channel scope

Runtime model

Gateway owns the WhatsApp socket and reconnect loop.

Outbound sends require an active WhatsApp listener for the target account.

Status and broadcast chats are ignored (`@status` , `@broadcast`).

Direct chats use DM session rules (`session.dmScope` ; default `main` collapses DMs to the agent main session).

Group sessions are isolated (`agent:<agentId>:whatsapp:group:<jid>`).

Access control and activation

DM policy Group policy + allowlists Mentions + /activation

`channels.whatsapp.dmPolicy` controls direct chat access:



pairing (default)

allowlist

open (requires allowFrom to include "*")

disabled

allowFrom accepts E.164-style numbers (normalized internally).

Multi-account override: `channels.whatsapp.accounts.<id>.dmPolicy` (and `allowFrom`) take precedence over channel-level defaults for that account.

Runtime behavior details:

pairings are persisted in channel allow-store and merged with configured `allowFrom`

if no allowlist is configured, the linked self number is allowed by default

outbound `fromMe` DMs are never auto-paired

Personal-number and self-chat behavior

When the linked self number is also present in `allowFrom` , WhatsApp self-chat safeguards activate:

skip read receipts for self-chat turns

ignore mention-JID auto-trigger behavior that would otherwise ping yourself

if `messages.responsePrefix` is unset, self-chat replies default to `[{identity.name}]` or `[openclaw]`

Message normalization and context



Inbound envelope + reply context

Media placeholders and location/contact extraction

Pending group history injection

Read receipts

Delivery, chunking, and media

Text chunking

Outbound media behavior

Media size limits and fallback behavior

Acknowledgment reactions

WhatsApp supports immediate ack reactions on inbound receipt via `channels.whatsapp.ackReaction` .

```
{
  channels: {
    whatsapp: {
      ackReaction: {
        emoji: "👁👁",
        direct: true,
        group: "mentions", // always | mentions | never
      },
    },
  },
}
```

Behavior notes:



sent immediately after inbound is accepted (pre-reply)

failures are logged but do not block normal reply delivery

group mode mentions reacts on mention-triggered turns; group activation always acts as bypass for this check

WhatsApp uses `channels.whatsapp.ackReaction` (legacy `messages.ackReaction` is not used here)

Multi-account and credentials

Account selection and defaults

Credential paths and legacy compatibility

Logout behavior

Tools, actions, and config writes

Agent tool support includes WhatsApp reaction action (`react`).

Action gates:

`channels.whatsapp.actions.reactions`

`channels.whatsapp.actions.polls`

Channel-initiated config writes are enabled by default (disable via `channels.whatsapp.configWrites=false`).

Troubleshooting

Not linked (QR required)



Linked but disconnected / reconnect loop

No active listener when sending

Group messages unexpectedly ignored

Bun runtime warning

Configuration reference pointers

Primary reference:

Configuration reference - WhatsApp

High-signal WhatsApp fields:

```
access:  dmPolicy , allowFrom , groupPolicy , groupAllowFrom , groups
delivery: textChunkLimit , chunkMode , mediaMaxMb , sendReadReceipts ,
          ackReaction
multi-account: accounts.<id>.enabled , accounts.<id>.authDir , account-
              level overrides
operations:  configWrites , debounceMs , web.enabled ,
            web.heartbeatSeconds , web.reconnect.*
session behavior: session.dmScope , historyLimit , dmHistoryLimit ,
                dms.<id>.historyLimit
```

Related

Pairing

Channel routing

Multi-agent routing

Troubleshooting

< Chat Channels

Telegram >



Powered by mintlify

>