

# Module 06



## Connecting Messaging Channels

Taking Your Agent Mobile

# Navigation Chart

---

By the end of this module, you will be able to:

1. **Create** a Telegram bot and connect it to 🦏 OpenClaw
2. **Understand** DM pairing mode and why it is critical for security
3. **Explain** the security implications of each messaging channel
4. **Chat** with your agent from your phone via Telegram
5. **Optionally** set up Discord and WhatsApp 🔗 connections
6. **Apply** the security rules from Module 01 to your channel configuration

## Ship's Logbook

Term	Definition
Channel	A messaging platform connected to 🦏 OpenClaw (Telegram, Discord, WhatsApp, etc.)
Bot token	A secret 🔑 credential from the messaging platform that lets OpenClaw control a bot account
DM pairing	Security feature: unknown senders must enter a code before the bot responds
BotFather	Telegram's official bot for creating and managing other bots
Guild ID	Discord's term for a server ID
Allowlist	A list of specific users permitted to message your bot — everyone else is blocked

# Which Channel to Set Up First

Channel	Ease of Setup	Security	Recommendation
Telegram	Easy	Good (pairing mode)	Start here
Discord	Medium	Moderate (server isolation)	Second option
WhatsApp	Medium	Moderate (needs 2nd phone #)	Third option
iMessage	macOS only	N/A for us	Not available on Windows
Slack	Medium	Good (workspace isolation)	For work contexts

*Telegram is recommended as your first and possibly only  channel.*

# Why Telegram?

---

- **Free** and available on every platform (phone, desktop, web)
- **Native bot support** — no hacks or workarounds needed
- **Threading and message chunking** — long AI responses split cleanly
- **DM pairing mode** works well for security
- **Access from any device** without touching your 🦏 OpenClaw laptop

# Telegram Step 1: Install Telegram

---

1. **Phone:** Download "Telegram" from App Store (iPhone) or Google Play (Android)
2. **Computer:** Download from <https://telegram.org/> or use <https://web.telegram.org/>
3. Create an account (requires a phone number)
4. Sign in on at least your phone — that is where you will chat with your agent most often

# Telegram Step 2: Create a Bot with BotFather

---

1. Open Telegram and search for @BotFather
2. Click on **BotFather** (look for the blue checkmark)
3. Click **Start** or type /start
4. Type /newbot and send it
5. When asked for a name, type: your agent
6. When asked for a username, choose something ending in bot :
  - Example: MyAssistantBot
  - Must be globally unique on Telegram
  - If taken, try: MyAI\_bot or My\_2026\_bot

# Telegram Step 3: Copy Your Bot Token


BotFather responds with:

```
Done! Congratulations on your new bot.
```

```
Use this token to access the HTTP API:
```

```
7123456789:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw
```

**Copy the bot token** — the long string starting with numbers.


**Security warning:** This  token controls your bot. Anyone with it can send messages as your bot. Treat it like a password. Never share it, screenshot it, or commit it to public repos.

**Use the Notepad trick:** Paste into Notepad first, verify it is one clean line, then copy from there.

## Telegram Step 4: Connect to OpenClaw

In your Ubuntu terminal:

```
openclaw channels add telegram
```

The wizard asks for your bot  token:

```
? Enter your Telegram bot token:  
7123456789:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw
```

**Expected output:**

```
- Telegram bot connected: @MyAssistantBot  
- DM pairing mode: enabled (default)
```

# Telegram Step 5: Restart and Pair

Restart the  gateway:

```
openclaw service restart
```

Start a chat with your bot:

1. Open Telegram on your phone
2. Search for your bot's username (e.g., @MyAssistantBot )
3. Click on it and press **Start**

Complete DM pairing:

1. The bot sends a pairing code: *"Pairing required. Your code: ABC123"*
2. Approve the pairing in the  OpenClaw TUI or dashboard
3. Once paired, the bot responds to your messages

# Telegram Step 6: Test It

---

Send a message from your phone:

```
Hi there! This is a test from Telegram.  
Do you remember my name?
```

If your agent responds with your name (from `USER.md`), everything is connected and working.

## Optional — Set a profile picture:

1. Go back to BotFather in Telegram
2. Type `/setuserpic`
3. Select your bot
4. Send a profile picture
5. Also try `/setdescription` and `/setabouttext`

# Security: DM Pairing Mode

## Verify it is active:

```
openclaw config channels telegram dm-mode
```

If it shows `open`, change it immediately:

```
openclaw config channels telegram dm-mode pairing
```

Mode	What Happens	Security Level
<b>pairing</b>	Unknown senders get a code; you must approve	High
<b>allowlist</b>	Only pre-approved user IDs can message	Very High
<b>open</b>	Anyone can message the bot	<b>DANGEROUS</b>
<b>disabled</b>	Bot does not respond to any DMs	Maximum (but useless)

# Security: Setting an Allowlist

For extra security, specify exactly who can message the bot:

```
openclaw config channels telegram allowlist \  
  add YOUR_TELEGRAM_USER_ID
```

To find your Telegram user ID:

1. Search for `@userinfobot` on Telegram
2. Start a chat with it
3. It tells you your numeric user ID

***Never set DM mode to "open"** unless you have a very specific reason and understand prompt injection risks.*

# ▮ Rough Waters: Group Chat Danger

---

**DO NOT put your 🦀 OpenClaw bot in group chats.**


Not "be careful." Not "use with caution." **Do not do it.**

- Every person in the group can send messages the bot reads
- Every message is a potential **prompt injection**
- You cannot control what other people type
- The bot might respond with **private information**
- Social engineering is easy in groups

**The rule: DMs only. Always.**

The only exception requires sandboxing (Module 10), require-mention, and tool policies.

# Setting Up Discord (Optional)

1. Go to <https://discord.com/developers/applications>
2. Click **New Application**, name it "your agent"
3. Click **Bot** in sidebar, click **Reset Token**, copy  token
4. Enable **Message Content Intent** under Privileged Gateway Intents
5. Click **OAuth2**, check **bot** scope, set permissions:
  - Read Messages / View Channels
  - Send Messages
  - Read Message History
6. Copy invite URL, invite bot to your **private server**
7. Enable Developer Mode, copy **Server ID** and **Channel ID**

# Setting Up WhatsApp (Optional)

```
openclaw channels add whatsapp
```

1. A QR code appears in the terminal
2. On your phone: **Settings > Linked Devices > Link a Device**
3. Scan the QR code


## Important considerations:

- **Strongly recommended:** Use a second phone number, not your personal one
- The bot can see your entire WhatsApp message history
- The bot can send messages as you
- If compromised, damage is limited to the secondary account

```
openclaw config channels whatsapp allowlist \
```


# Security Checklist for All Channels

---

Check	Status
DM pairing mode is enabled (not "open")	[ ]
Only your user ID / phone number is in the allowlist	[ ]
Bot is NOT in any group chats	[ ]
Bot  token is stored securely (not in public repos or screenshots)	[ ]

## Security Checklist for All Channels (continued)

---

Check	Status
If using Discord: bot is in a private server with only you	<input type="checkbox"/>
If using WhatsApp: ideally using a second phone number	<input type="checkbox"/>
 Gateway authentication token is set	<input type="checkbox"/>
You have tested that the bot responds ONLY to you	<input type="checkbox"/>

# Channel Security: Best Practices from the Community


## Dedicated Accounts

- Create a **fresh Gmail/email** specifically for the bot
- Share calendar **read-only** from your personal account
- If compromised, you only lose the bot account — not your personal one

## Email Safety Protocol

If connecting email: check inbox every 2 minutes. **Trusted senders** (whitelist) can trigger actions. All other senders: **read-only summary only**. NEVER auto-reply to unknown senders.

## Password Manager Integration

Store bot  credentials in a **dedicated password manager vault** (e.g., "Shared with OpenClaw" in 1Password). API keys belong in the password manager, not in TOOLS.md or on disk.

## ► Shoals and Sandbars

Mistake	Fix
Bot does not respond on Telegram	Check dashboard for pending pairing requests
"Unauthorized" error	Regenerate 🔑 token via BotFather, use Notepad trick
Bot responds to strangers	Change DM mode to "pairing" immediately
WhatsApp connection drops	Re-scan QR code: <code>openclaw channels login</code>
Discord bot can't see messages	Enable Message Content Intent in Developer Portal
Bot is slow to respond	Normal: AI responses take 2-10 seconds
Bot 🔑 token accidentally shared	Regenerate immediately via BotFather <code>/revoke</code>

# Hands on Deck

---

## **Part 1: Set Up Telegram** *(15 min)*

1. Create a bot with BotFather
2. Connect it to 🦏 OpenClaw
3. Complete DM pairing
4. Send a test message from your phone

## **Part 2: Verify Security** *(5 min)*

1. Confirm DM pairing mode is active
2. Check that only your user ID is allowed
3. If possible, test that the bot blocks other accounts

# Hands on Deck *(continued)*

## Part 3: Have a Mobile Conversation *(10 min)*

From your phone via Telegram:

1. Ask your agent: *"What's your name and what do you know about me?"*
2. Ask your agent to create a to-do list for tomorrow
3. Ask your agent a question about a topic you are interested in

*Notice how the experience feels different from the TUI. Telegram feels like texting a knowledgeable friend. The TUI feels like working at a terminal.*

## Part 4: Security Test *(5 min)*

- Have a friend (or second account) try to message the bot
- They should be blocked by DM pairing mode


# Treasure Chest

---

1. **Telegram is the recommended first channel** — easy setup, good security, available everywhere
2. **DM pairing mode is your first line of defense** — never set it to "open"
3. **Never put your bot in group chats** — DMs only, always
4. **Bot tokens are secrets** — treat them like passwords
5. **WhatsApp ideally needs a second phone number** — otherwise the bot accesses your personal messages
6. **Discord needs a private server** — do not add the bot to public servers
7. **Test security after setup** — verify the bot only responds to you
8. **Start with one channel** — expand when comfortable

# Next Port of Call

## Module 07: Skills and ClawHub

*Your agent is now mobile. In Module 07, we teach your agent new  abilities. Skills are like apps for your agent — email management, image generation, note-taking, and thousands more from the community marketplace.*