

DARWIN × MANYCHAT

Let Darwin answer your Instagram DMs — setup guide

Once set up, every DM to your Instagram goes through ManyChat to Darwin. An agent drafts the reply — and nothing goes out until you approve it. The approval question reaches your dashboard and your phone. When the computer is off, no message is lost: Darwin picks them up on startup.

■ What you need

- > A ManyChat account (Pro) connected to your Instagram.
- > Darwin with a public address — the tunnel in Settings → Remote access must be running (e.g. your-name.darwinaassistant.com). ManyChat needs somewhere to deliver.
- > 10–15 minutes of clicking along with this guide.

■ Step 1 — Darwin: Settings → ManyChat

In Darwin open Settings → Communication → ManyChat and fill in:

- > Webhook secret (X-Darwin-Hook) — invent a random string, at least 32 characters (shorter is refused). Until it is set, the webhook does not exist to the outside world. You will paste the same string into ManyChat later.
- > ManyChat API token — in ManyChat: Settings → API → copy the token. Darwin uses it to send replies and to fetch missed messages.
- > Who drafts / who reviews / approval wait — keep the defaults until you have a reason to change them.

Save. No restart needed.

// SETTINGS **many** X CLOSE

All > You and Darwin > **Agents** > Brains and keys > Memory and knowledge > Communication >

Click a heading to read what a section does – or the speaker, and Darwin tells you.

MANYCHAT – INSTAGRAM DM REPLIES

Instagram messages (via ManyChat) are answered by an agent – but **nothing goes out without your approval**: every draft reaches you as a question (dashboard, phone, Telegram, PrivateChat). YES sends, NO discards – and if you type feedback, the agent rewrites the draft and asks again.

Webhook secret (X-Darwin-Hook) SET ✓

ManyChat API token SET ✓

Who drafts replies

Approval wait (hours)

How long a draft waits for your YES/NO. When it expires the message is NOT lost – it lands in the unresolved queue and can be rerun ("handle the unresolved").

Who reviews drafts

▼ How to set it up

1. Invent a **webhook secret** – any random string, **at least 32 characters** (shorter is refused by the server) – and paste it above. Until it is set, the webhook does not exist to the outside world.
2. In ManyChat: **Settings** → **API**, copy the API token and paste it above (this is how Darwin sends the replies).
3. In ManyChat: **Automation** → **Default Reply** → in the flow add **Actions** → **Make External Request**: method POST, URL `https://YOUR-ADDRESS/api/plugin-hook/manychat` (your public tunnel address from above), header `X-Darwin-Hook = your secret`, body = **Add Full Contact Data**.
4. So Darwin gets **every** message (not once a day), set Default Reply to trigger **every time**.

⚠ Requires a public address (tunnel, Remote access section) – ManyChat needs somewhere to deliver the webhook. The reply agents are deliberately tool-less; strangers' text is data only to them.

SAVE

The ManyChat section in Darwin's Settings: secret and token (values are never shown, only "set"), agent choice, approval wait — and the built-in "How to set it up" steps.

Step 2 — ManyChat: two custom fields

In ManyChat create two text fields (Settings → Fields → Custom User Fields → New field, type Text):

- > `darwin_ok` — says whether delivery to Darwin succeeded,
- > `darwin_missed` — marks messages that failed during an outage; Darwin finds them by this mark later.

Step 3 — ManyChat: the Default Reply flow

Automation → New Automation → Start From Scratch (fig. 1), or edit your existing Default Reply flow.

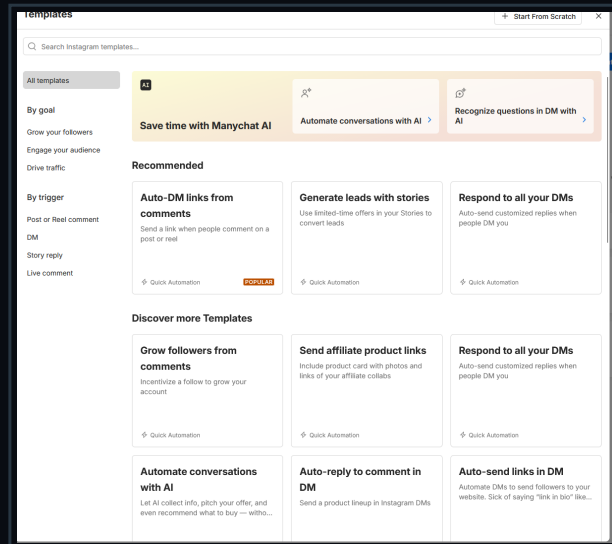


Fig. 1 — the template gallery; Start From Scratch is top right.

Build the flow exactly like this (final shape in fig. 2):

Trigger: User sends a Direct Message (Default Reply)

-> Send Message: "Got it, let me check and get back to you."

-> Actions: 1) Set User Field darwin_missed = 1
2) Set User Field darwin_ok = 0
3) External Request

-> Condition: darwin_ok is v2

yes -> Actions #1: Set User Field darwin_missed = 0

no -> (leave the branch empty)

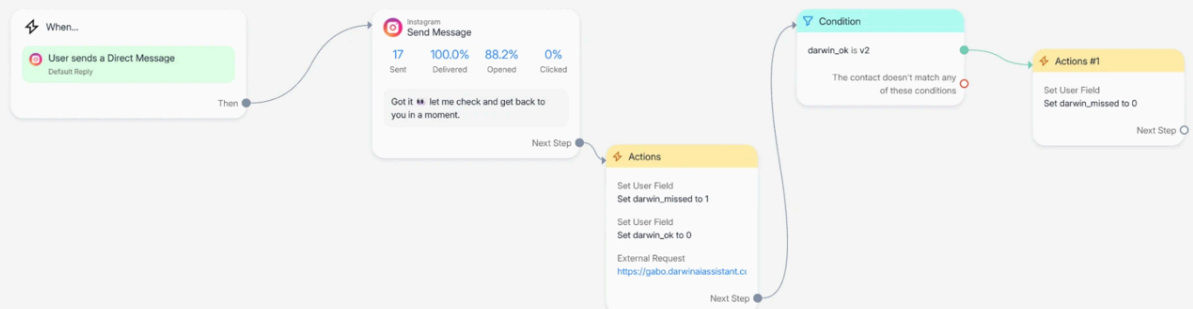


Fig. 2 — the finished Default Reply flow: mark before the request, unmark only after confirmed success.

Why this order: every message is first marked “undelivered” (darwin_missed = 1) and unmarked only once delivery demonstrably succeeded — the darwin_ok field receives the value v2 from the server's response. If delivery fails (computer off, outage), the mark stays and Darwin fetches the message later by itself.

Careful

Use the exact comparison “is v2” in the condition (fig. 3) — not “has any value”; zero is a value too.

Write the field values as plain text 1 / 0 — ManyChat has no “clear field” option.

Switch the Default Reply trigger to “every time” — otherwise ManyChat delivers only one message per 24 hours per person.

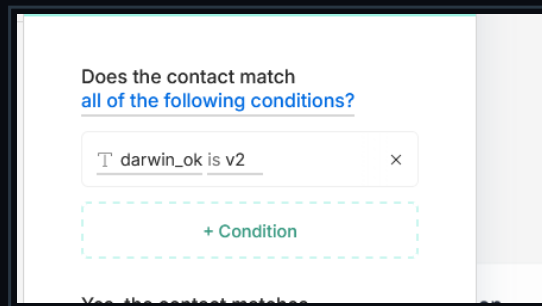


Fig. 3 — the success condition: darwin_ok is v2.

Step 4 — the External Request (the heart of it)

In the Actions block click + Action → External Request and open Edit Request (fig. 4). Fill in:

Request Type	POST
Request URL	https://YOUR-ADDRESS/api/plugin-hook/manychat
Headers	X-Darwin-Hook = your secret from step 1
Body	Full Contact Data (ManyChat's option — name, id, message text)
Response mapping	JSONPath \$.version → field darwin_ok (fig. 5)

Fig. 4 — Edit Request: the URL and the Headers / Body / Response mapping tabs.

Fig. 5 — Response mapping: \$.version → darwin_ok.

Then click Test Request — it must return 200 OK with the body {"version":"v2",...} (fig. 6). On errors, check your tunnel address and the secret (at least 32 characters, identical to Settings). Finally Save, and Set Live on the flow.

Edit Request

Request Type: POST | Request URL: <https://gabio.darwinassistant.com/api/plugin-hook/manychat> | Contact for testing: Gabriele Istruttore di Windsurf

Headers | Body | **Response** | Response mapping

► Response Body

► Request Headers

▼ Request Body

```
{
  "key": "user:1995653579",
  "id": "1995653579",
  "page_id": "",
  "user_refs": [],
  "status": "active",
  "first_name": "Gabriele",
  "last_name": "Istruttore di Windsurf",
  "name": "Gabriele Istruttore di Windsurf",
  "gender": null,
  "profile_pic": "https://app.manychat.com/ava/5325826/1995653579/b1c19ee97b968b38da09a788ae535a7",
  "locale": null,
  "language": null,
  "reference": "HUFFMAN"
}
```

Cancel | Test Request | Save

Fig. 6 — a successful test: 200 OK, version v2, Full Contact Data in the request body.

■ Step 5 — try it

- > From another Instagram account, send your page a message.
- > Within seconds the automatic acknowledgement arrives ("Got it...").
- > An approval bubble appears in Darwin: the person's message + the draft reply. YES sends it, NO discards it — and if you type feedback instead, the agent rewrites the draft and asks again.
- > Once approved, the reply goes out to Instagram.

■ How it runs from here (nothing left to configure)

- > Approvals: you have 4 hours to decide (configurable in Darwin Settings); after that the message lands in the unresolved queue — ask Darwin "what's unresolved in ManyChat" and say "handle it" to rerun the round. Nothing is lost.
- > Computer off: messages get marked in ManyChat (`darwin_missed = 1`) and Darwin fetches and processes them on startup and every 30 minutes after. "Catch up on ManyChat" does it immediately.
- > The only hard limit is Instagram itself: a business can reply only within 24 hours of the person's last message (Meta's rule). Older messages are dropped — they can no longer be sent.
- > Safety: the drafting agents deliberately have no tools — strangers' text is data only to them. And not a single reply leaves without your explicit YES.

Darwin — the local AI assistant for Windows · darwinaiassistant.com