Mozout

Generated on 15/11/2026 00:38

KNOWLEDGE BASE · PUBLIC DOCUMENT

n8n and WhatsApp: a message flow with Evolution API

n8n Workflows

CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-2844	v2026.10.10	International · English

MozOut International
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/2844

n8n and WhatsApp: a message flow with Evolution API

n8n Workflows Created on **October 10, 2026** 3 min read



n8n does not talk to WhatsApp on its own. The usual flow puts **Evolution API** in the middle: it is the bridge to WhatsApp, and n8n is the brain. When a message arrives, Evolution API tells n8n by webhook; to reply, n8n makes an HTTP request to Evolution API. Both are third-party programs you run on a VPS of your own.

> **There is a risk worth knowing about.**
> With the QR code connection (Baileys), the WhatsApp link is unofficial: the number can be blocked, especially if you send a lot, or to people who did not ask. Read **Baileys or the Cloud API: the risk and the rules** before putting real customers on this flow.

The two directions

Direction	How it is done	In n8n
Receiving a message	In Evolution API you point the instance webhook at the n8n production address and choose the incoming-messages event.	A Webhook node as the trigger.
Sending a message	n8n calls the Evolution API send endpoint, with the API key in a header and the number and text in the body.	An HTTP request node.

The exact endpoint names and fields change with the Evolution API version. Use its documentation for your version, not an example copied from an old tutorial. Evolution API itself is covered in **installing Evolution API with Docker Compose**.

Building the flow

1. **Put both on the same VPS**, on the same Docker network. Inside a container, `localhost` is the container itself, so n8n calls Evolution API by the **service name** in the compose file, not by `localhost`.

2. **Create the instance and connect it** with the QR code. See **sending your first message with Evolution API**.

3. **Create the workflow with a Webhook node** and copy the **production** address into Evolution API. The test one only works while you design. See **n8n webhooks**.

4. **Ignore the messages the instance itself sent.** The incoming data says whether the message is yours. Without that filter the workflow answers its own replies, in an endless loop.

5. **Reply with an HTTP request node** to Evolution API. Keep the API key in the n8n credential, never typed into the workflow.

6. **Add pauses and limits.** Avoid sending many messages in a row to different numbers.

When it does not work

Symptom	Likely cause
The workflow never starts	Evolution API is posting to the test address, or the workflow is inactive.
The send request gives a connection error	You used `localhost` instead of the service name, or the containers do not share a network.
401 or 403 from Evolution API	The API key is not going in the header, or it is the wrong one.
Works, then stops	The instance disconnected from WhatsApp. See **the instance stuck on "connecting"**.

The same flow, step by step,
is in **Evolution API and n8n: a working flow**. To protect Evolution API, see **securing Evolution API**.

RECOMMENDED
Need a VPS to run n8n and Evolution API?
See the VPS servers →

SEE ALSO

- What Evolution API is
- Evolution API and n8n: a working flow
- Receiving messages: Evolution API webhooks
- WhatsApp for business: the risks, and the practices that avoid them

Mozout

MozOut International

RLX Holdings (Pty) Ltd. · Group RLX Holdings

TAX NO. (SARS)	943 907 6234
REGISTRATION NO. (CIPC)	2025/667953/07
REGISTERED OFFICE	1111 Olive Street Saint Louis 63101, Missouri, United States
CONTACT	info@mozout.com · +27 78 823 3325 · www.mozout.com
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-INT-KB-2844 · v2026.10.10 · Generated on 15/11/2026 00:38