Mozout

Generated on 15/11/2026 00:39

KNOWLEDGE BASE · PUBLIC DOCUMENT

Webhook or polling: how one system learns that something changed in another

Servers & Technology

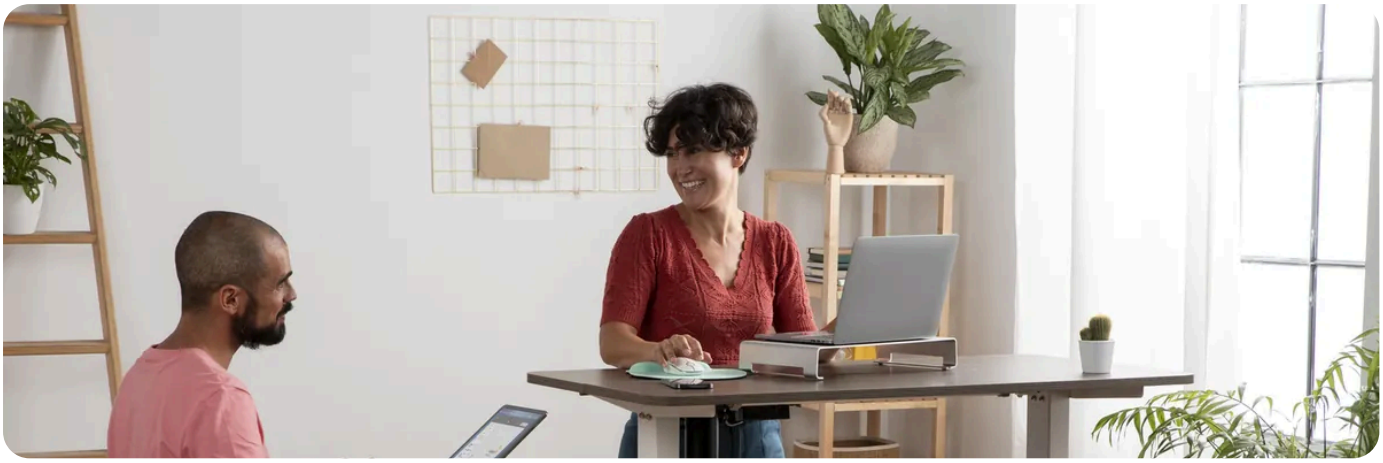
CODE	VERSION	COUNTRY · LANGUAGE
RLX-INT-KB-3540	v2026.10.10	International · English

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/3540

Webhook or polling: how one system learns that something changed in another

Servers & Technology Created on **October 10, 2026** 3 min read



If the other system can notify you, use a webhook: it arrives at once and wastes no requests. If it cannot, or if you cannot risk missing a notice, poll. The most robust setup combines the two. It is the question behind “how do I know the customer paid”, “how do I receive the messages” or “how do I know the order changed”.

The two ideas

With **polling**, your system asks every so often, “anything new?”. You are the one asking, at the pace you choose. With a **webhook**, it is the other system that calls you: when something happens, it makes an HTTPS request to an address of yours and hands over the data. It does the notifying, at the moment of the event.

Side by side

Criterion	Polling	Webhook
Speed	Delayed until the next interval.	Almost immediate.
Waste	Many requests that answer “nothing new”.	A request only when something happens.
What you need on your side	A scheduled task. No public address needed.	A public HTTPS address that is always answering.
If your server is down	Catches what it missed on the next poll.	May lose the notice. Services usually retry, but only for a limited time.
Security	You control the connection you make.	You must confirm the notice really came from who it says.
Simplicity	Very simple.	A little more care.

When to use each

1. **Use a webhook** for payments, incoming messages and orders, where waiting minutes costs you. See **webhooks from Meta: verifying and receiving**.

2. **Use polling** when the other side has no webhooks, when you only want an hourly report, or when you have no public address to receive on. It is what a scheduled task does. See **cron jobs**.

3. **Combine them** when missing a notice is costly: the webhook handles the immediate side and a slow poll, say hourly, picks up what escaped.

4. **In n8n**, the Webhook trigger matches the first case and the Schedule Trigger the second. See **n8n Schedule Trigger or a cron job**.

What a well-made webhook does

1. Answers quickly with a success code, and only then does the heavy work.

2. Confirms where the request came from, normally through a signature. See **receiving a webhook in PHP and checking its signature**.

3. Copes with the same notice twice: services may resend, and you must not record the same payment twice.

4. Logs what it receives, so you know what happened when something fails.

In n8n, the test webhook and the production webhook are different addresses.
The test one only answers while it is listening in the editor. The production one only answers with the workflow active. Putting the test address into an outside service is the most common cause of “the webhook does not fire”.

So that notices are not lost, store them before processing.
A notice that arrives and fails halfway can be picked up again if it was stored; if not, it is gone.

NEED A HAND?

Connecting two systems and not sure which route to use? Tell us what they need to say to each other.

Open a support ticket →

SEE ALSO

- Receiving a webhook in PHP and checking its signature →
- n8n Schedule Trigger or a cron job: which to use →
- Cron jobs: what they are for and how to create one →
- VPS server →

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-3540 · v2026.10.10 · Generated on 15/11/2026 00:39