



Webinar

Zabbix Webhooks in Practice: Slack, SMS & Voice via SMSEagle, and Jira

All microphones are muted

Please ask questions in Q&A, not in Chat

Use Chat for discussion, networking, or applause

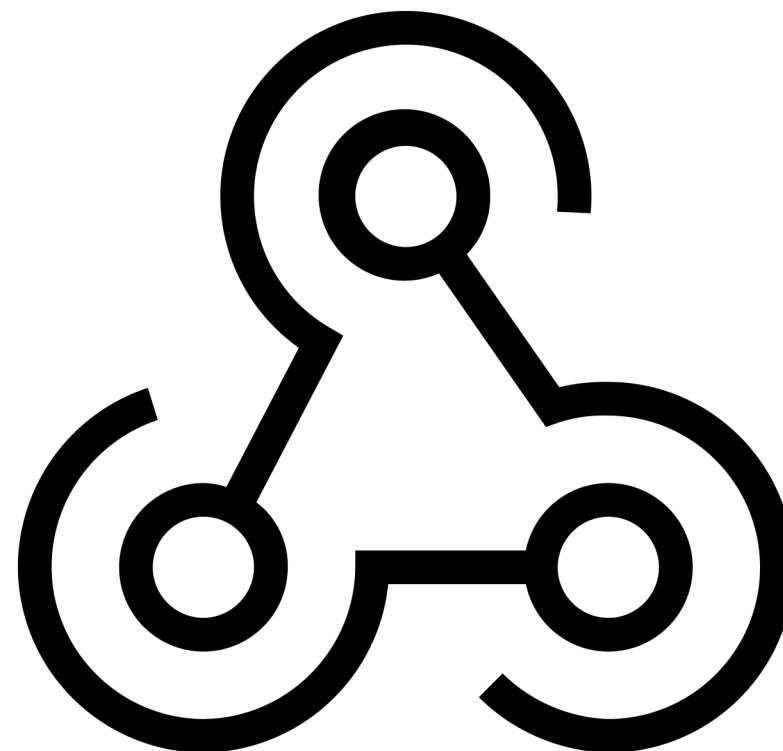
1

What is webhook



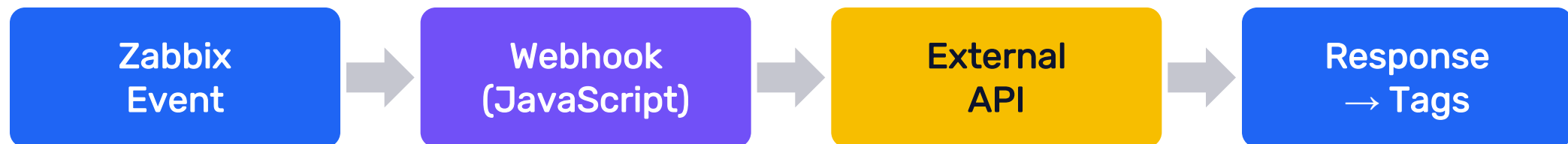
What is a webhook?

- › An HTTP call to a URL – Zabbix sends event data, the target system responds
- › The response is not thrown away – it can be parsed and used back in Zabbix
- › Two-way integration: send the alert, receive a reference, store it on the event
- › Pure JavaScript inside Zabbix – no external scripts, no agents on the server
- › **Runs only on the Zabbix server** – outbound calls leave from the server's IP, not from agents or proxies (use the `http_proxy` parameter – see Tips & Tricks)



How Zabbix calls a webhook

- ▶ Trigger fires → action runs → JavaScript webhook executes with event context
- ▶ HTTP POST to the target API → returns JSON (ticket id, URL, status)
- ▶ Zabbix parses the response and writes values back as event tags



JSON response (e.g. Jira ticket id + URL) is parsed and written back as event tags

Live example: Jira ticket callback

- ▶ Zabbix calls the Jira REST API and creates an issue from the trigger event
- ▶ Jira responds with the new issue: key (OPS-123) and self URL
- ▶ The webhook returns those values to Zabbix as event tags – fully automatic

Jira API response → parsed by the webhook script

```
{  
  "id": "10042",  
  "key": "OPS-123",  
  "self": "https://jira.initmax.cz/rest/api/2/issue/10042"  
}
```

- ▶ Zabbix event tags: `jira_key=OPS-123`

Where webhooks live in Zabbix

- ▶ Two places to define a webhook: **Alerts** → **Media types** (backend) and **Alerts** → **Scripts** (frontend).
- ▶ **Only Media types** can write event tags and add a URL link to the event.
- ▶ Scripts cover manual menu entries and Action-operation scripts (often with `{MANUALINPUT}`) for calls like AWX role triggers.

MEDIA TYPE (BACKEND)

Notification webhook

- ✔ Adds tags to the event
- ✔ Adds URL link to the event
- ✗ Runs only inside an Action

▶ e.g. Jira ticket, SMS via SMSEagle

SCRIPT – MANUAL HOST ACTION

Host / event menu entry

- ✗ No tags, no event URL
- ✔ Menu entry name + URL
- ✔ Runs on demand from UI

▶ e.g. "Open Jira ticket" menu link

SCRIPT – MANUAL EVENT ACTION

Webhook from an Action

- ✗ No tags, no event URL
- ✔ Works with `{MANUALINPUT}`
- ✔ Runs via Action operation

💡 Tip: Update OP as workaround

▶ e.g. Trigger an AWX role

2

Native webhooks templates



Native webhook templates in Zabbix

Zabbix ships with 25+ webhook media types out of the box – no scripting required, just plug in the API token.

Ticketing & ITSM

- ▶ Jira / Jira Service Management
- ▶ ServiceNow
- ▶ Redmine
- ▶ GitHub
- ▶ GitLab
- ▶ Zendesk
- ▶ OTRS
- ▶ iLert

Chat & Messaging

- ▶ Slack
- ▶ Microsoft Teams
- ▶ Discord
- ▶ Mattermost
- ▶ Rocket.Chat
- ▶ Telegram
- ▶ Pushover
- ▶ Express.ms

From initMAX

- ▶ WhatsApp (Cloud API)
- ▶ SMSEagle media types

github.com/initMAX/Zabbix-Templates

Inside a template: Jira webhook media type

- ▶ Parameters block: Jira URL, project key, issue type, API token, summary template
- ▶ JavaScript handles auth, payload assembly, HTTP call and error handling
- ▶ Update operation reuses jira_key tag to comment on the existing issue – no duplicates
- ▶ Recovery operation transitions the issue to Done when the trigger resolves

PROBLEM Create issue

POST → Jira creates new ticket; key + URL stored as event tags

UPDATE Comment

Re-uses jira_key tag; appends update; no duplicate ticket

RECOVERY Transition to Done

Auto-resolves Jira issue when Zabbix problem clears

3

What is SMSEagle



What is SMSEagle

- ▶ Hardware SMS/MMS gateway made in the EU – runs on-premise, sends over the cellular network
- ▶ **Works even when the Internet is down** – recent Cloudflare, AWS and Azure outages prove why cloud SMS providers can fail
- ▶ **Data SIM as last-resort connectivity** – can feed your VPN gateway during a total outage so the NOC stays reachable
- ▶ **Built-in network monitoring** – SMSEagle can test Zabbix and other devices on its own and send notification if they go down
- ▶ **Hardware add-ons** – temperature sensors, GPIO, relay outputs; used by Alza, Allwyn... aligns with NIS2 / GDPR

RECENT OUTAGE

Cloudflare

Nov 2025

Workers + DNS resolver

RECENT OUTAGE

AWS us-east-1

Oct 2025

DynamoDB + IAM cascading

RECENT OUTAGE

Azure

Multiple

Front Door + identity

Product lineup

Three product families - from a single SIM box for branch offices to enterprise gateways for high-volume sending.



ENTRY

NXS-9700

Single 4G/5G modem. Perfect for critical alerts in small and mid-size environments.



RECOMMENDED

NXS-9750-4G

Dual modem - redundancy across two SIMs and carriers. Automatic failover. Our pick for most deployments.



HIGH VOLUME

MHD-8100-4G

Eight modems in a 1U rack. Enterprise gateway for telco, mass campaigns and multi-carrier failover.

initMAX × SMSEagle – Silver Partner

SILVER PARTNER

initMAX is your SMSEagle expert

Official Silver Partner of SMSEagle - worldwide partnership. Direct access to engineering, firmware roadmap and unreleased features.

Partner discount up to **20%** - for existing and new customers

TRY BEFORE YOU BUY

Free live demo instance

Real hardware. Real SMS / voice credit.
No sales call.

Register, login arrives in seconds, send your first SMS in minutes.

Register and more info: initmax.com/smseagle-silver-partner • initmax.cz/smseagle-silver-partner

Key features & add-ons

One device covers the full notification spectrum - and you can extend it with hardware add-ons.

SMS & MMS

Direct sending from your own SIM card. Phonebook, groups, delivery reports, two-way messaging.

Voice calls (TTS)

Critical alerts that actually ring the phone and read the message aloud - in any supported language including Czech.

Email-to-SMS

Any system that can send email can send SMS - SMSEagle converts it automatically.

REST API & webhooks

Easy integration with Zabbix, Jira, custom scripts - same webhook pattern as anywhere else.

Network monitoring

Built-in IP / TCP port checks - SMSEagle can alert you when your Zabbix server itself goes down.

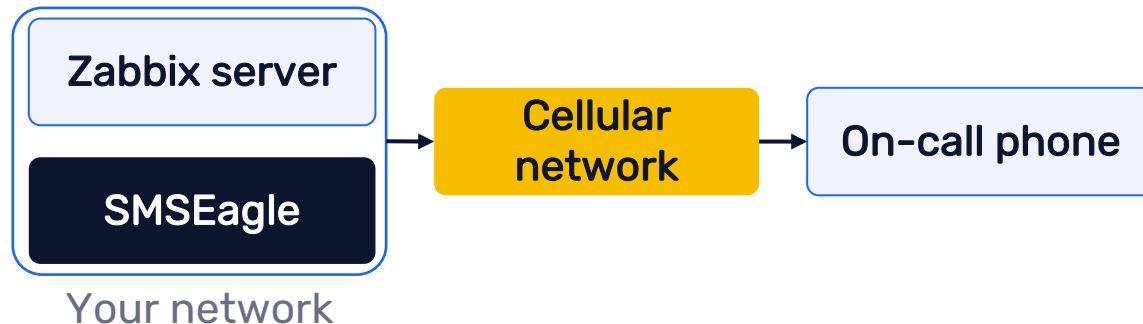
Hardware add-ons

Temperature sensors, contact inputs, GPS antenna - monitor the server room from the same box.

On-premise vs cloud SMS

When the Internet is down, cloud SMS providers are useless. SMSEagle keeps notifying because it does not need the Internet at all.

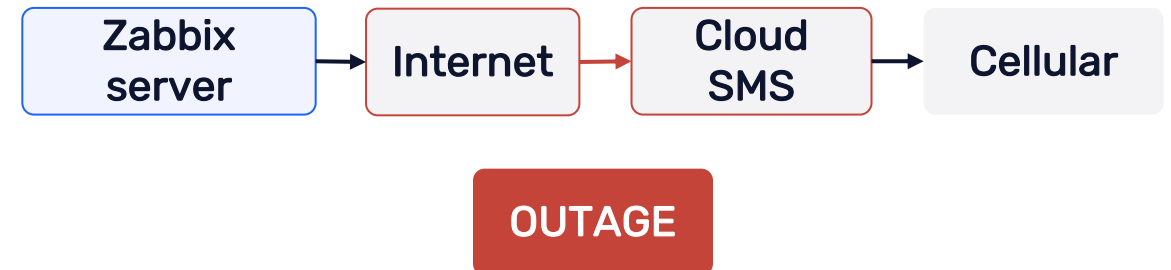
SMSEagle on-premise



Notification delivered

- ⊙ Works without Internet
- ⊙ Data stays in your network (NIS2 / GDPR)
- ⊙ No per-message billing

Cloud SMS provider



Notification fails

- × Depends on the Internet and the provider
- × Cloudflare / AWS / Azure outages affect you
- × Data leaves your network

4

Where can webhooks be used



Where webhooks fit in your workflow

- ▶ A webhook is just an HTTP call - so anywhere a system speaks HTTP, Zabbix can talk to it. Four common categories:

Notifications

Slack, MS Teams, Discord, WhatsApp, SMS, Voice calls, Telegram, push notifications. Reach the right people on the right channel.

Ticketing & ITSM

Jira, ServiceNow, Zendesk, GitHub Issues, Redmine. Open / update / close tickets automatically and link them back to the event.

Automation & remediation

AWX / Ansible Tower, Rundeck, custom REST APIs, cloud functions. Webhook can trigger a remediation playbook, not just an alert.

Data sync & enrichment

Netbox / CMDB, asset databases, dashboards, SIEM. Push events to other systems or pull metadata into Zabbix tags.

Three patterns of webhook usage

- ▶ The same webhook mechanism solves very different problems depending on what you do with the response.

PATTERN 1

Fire-and-forget

Send a notification

Slack message, SMS, email, Discord ping. We do not care about the response - the notification either arrived or it did not.

Example: Slack alert, push notification

PATTERN 2

Two-way

Send and store the response

Open a Jira ticket, create a ServiceNow incident, register a record in Netbox. Zabbix stores the returned ID and links the event to it.

Example: Jira ticket URL on the problem

PATTERN 3

Chained

Trigger another action

Webhook calls an automation API (AWX, Rundeck, custom REST) which then triggers a remediation playbook.

Example: auto-remediation, escalation

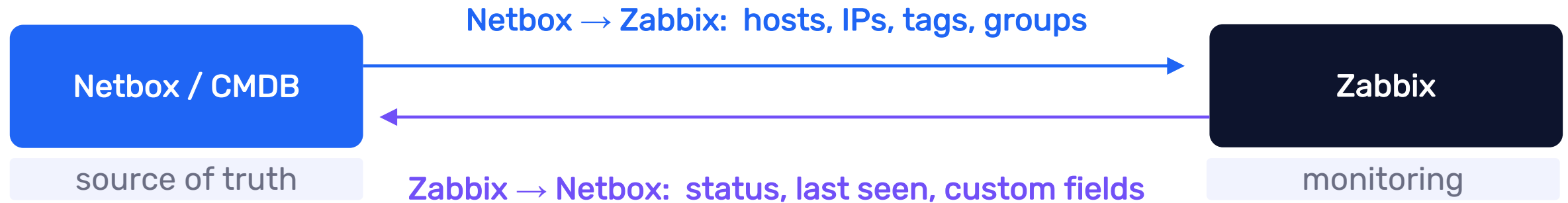
5

Tips & Tricks



Two-way sync with CMDB (Netbox)

- › Keep Zabbix and your CMDB consistent - in both directions - using webhooks.



Netbox → Zabbix (via Netbox webhook)

- › New device in Netbox → host auto-created in Zabbix
- › Rack / site / role → host group + template
- › Tags, owner, custom fields → Zabbix host tags

Zabbix → Netbox (via Zabbix webhook)

- › Host down → flag the device in Netbox
- › Discovered services / interfaces back into CMDB
- › One source of truth, no manual data entry twice

Conditional logic & chained webhooks

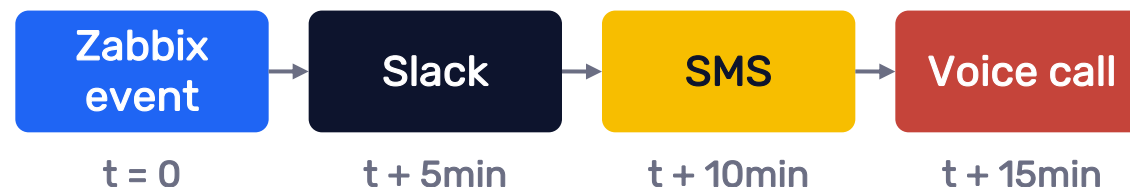
- ▶ The webhook script is plain JavaScript - put real logic in there, do not just forward every event the same way.

1. Conditional routing in the script

```
// inside the webhook JS
if (severity >= 4) sendSMS();
else if (tags.team) sendSlack();
else sendEmail();
// suppress out-of-hours
if (isWeekend()) skipCall();
```

- ▶ Route by severity, tag, host group or time of day
- ▶ SMS only for production-critical; Slack for the rest
- ▶ One media type, many destinations - cheaper to maintain

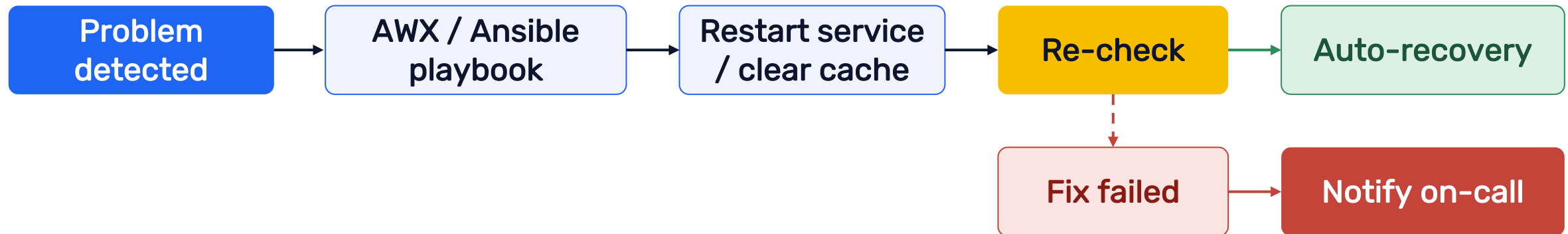
2. Chained webhooks (escalation)



- ▶ First webhook posts to Slack and stores `message_ts`
- ▶ If not ACKed in 5 min, Zabbix escalation step fires the SMS webhook
- ▶ Still no ACK → voice call wakes the on-call engineer
- ▶ Each webhook can call the next system - chain them however you need

Self-healing with webhooks

- › Stop waking people up for problems that fix themselves. Let a webhook try the obvious remediation first - alert the human only if it fails.



Real-world examples

- › Filesystem full → run cleanup playbook, rotate logs
- › Service down → systemctl restart, then re-check status
- › SSL cert expiring → trigger acme.sh / cert-manager renewal
- › Node unhealthy → cordon / drain in Kubernetes
- › Queue backlog → scale workers via cloud API
- › Webhook ACK on success → Zabbix event auto-closes

Routing webhooks through an HTTP proxy

- ▶ Every official Zabbix webhook media type already has `req.setProxy()` wired up. It reads the proxy parameter from the Parameters list - `http_proxy` on 7.0.11+ / 7.4 / 8.0, `HTTPProxy` on older versions. The catch: the row is not in the default Parameters list - just add it manually.

Media type: Parameters tab (default)

<code>alert_message</code>	<code>{ALERT.MESSAGE}</code>
<code>alert_subject</code>	<code>{ALERT.SUBJECT}</code>
<code>endpoint_url</code>	<code>https://api.example.com/v2</code>
<code>token</code>	<code>{\$WEBHOOK.TOKEN}</code>
<code>severity</code>	<code>{EVENT.SEVERITY}</code>

http_proxy parameter is not pre-filled

Add the row manually. Use `HTTPProxy` on 6.0 / pre-7.0.11.

```
// Webhook script (JavaScript)
var params = JSON.parse(value);

// 7.0.11+ / 7.4 / 8.0 use lowercase:
if (params.http_proxy) {
    req.setProxy(params.http_proxy);
}

req.addHeader('Content-Type:
  application/json');
var resp = req.post(params.endpoint_url,
  payload);
return resp;
```

Renamed in 7.0.11 LTS / 7.4 / 8.0. On older versions the parameter is `HTTPProxy`.

HTTPProxy & Timeout in the docs

agent	string	Client agent. Zabbix will pretend to be the selected browser. This is useful when a website returns different content for different browsers. Default: Zabbix.
http_proxy	string	Proxy that will be used by the web scenario given as: <code>http://[username[:password]@]proxy.example.com[:port]</code>
variables		Root element for web scenario variables that may be used in scenario steps.

HTTPProxy → http_proxy

Renamed to lowercase in 7.0.11 (LTS), 7.4, 8.0. Older versions (6.0 LTS, 7.0.0-7.0.10) still use HTTPProxy.

`http://[user[:pass]@]host[:port]`

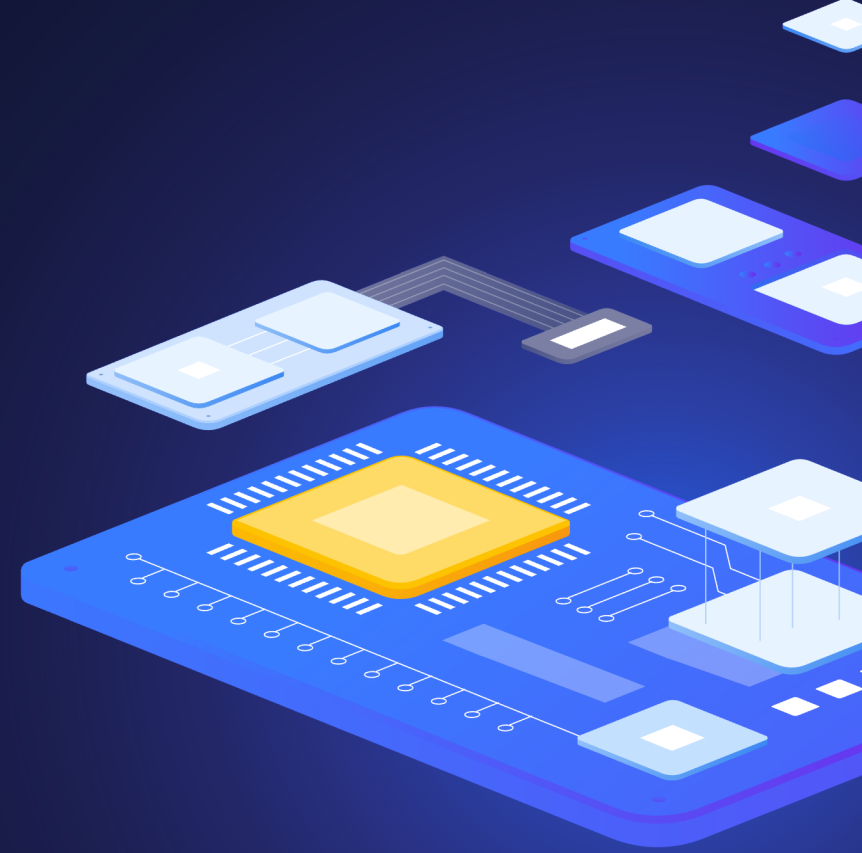
Timeout (webhook)

Media type field, defines how long the script may run.

Value must be one of 1-60 seconds.

If the remote API is slow, your webhook fails - retries via Action escalation steps.

DEMO



Questions?



Contact us:

Phone:



+420 800 244 442

Web:



<https://www.initmax.cz>

Email:



tomas.hermanek@initmax.cz

LinkedIn:



<https://www.linkedin.com/company/initmax>

Twitter:



<https://twitter.com/initmax>

Tomáš Heřmánek:



+420 732 447 184