

# Automations Grundlagen

# Automations Grundlagen

**Automations** sind Chatlix' visueller Builder für „Wenn X, dann Y“. Im Gegensatz zu Event Reactions oder Reward-Bindings können sie verzweigen, Bedingungen prüfen, Counter führen, Variablen lesen/schreiben und externe Webhooks ansprechen.

Kurz: Automations sind das, was du nimmst, wenn eine einfache Reaction nicht mehr reicht.

## Wann eine Automation?

- „Auf jeden 100. Cheer ein Konfetti-Overlay.“ → Counter.
- „Wenn Stream länger als 1 h läuft, andere Begrüßung als am Anfang.“ → Bedingung.
- „Bei einem Sub: Sound + Overlay + Chat + Discord-Webhook + 5 Loyalty-Punkte für Inviter.“ → mehrere Actions inkl. externer.
- „Pro Stream zähle Subs, zeig die Zahl live im Overlay.“ → Counter + Overlay-Variable.
- „Pro Zuschauer max 1× pro Stunde diese Action ausführen.“ → User-Cooldown (siehe eigene Seite).

## Wo im Dashboard

- `/dashboard/automations` — Liste aller Automations.

- `/dashboard/automations/templates` — vorgefertigte Automations (siehe *Automation-Templates*).
- `/dashboard/automations/variables` — Runtime-Variablen.
- `/dashboard/automations/counters` — Counter-Verwaltung.
- `/dashboard/automations/queues` — Action-Queues und Throughput.
- `/dashboard/automations/runs` — Lauf-Historie pro Automation.
- `/dashboard/automations/debug` — Live-Debugging einzelner Automations.
- `/dashboard/automations/webhooks` — eingehende und ausgehende Webhooks.

# Anatomie einer Automation

Jede Automation besteht aus drei Teilen:

1. **Trigger** — Was startet sie? Twitch-Event, Reward-Einlösung, Chat-Befehl, Webhook, Timer, Loyalty-Schwelle.
2. **Conditions** — Optionale Bedingungen. *Tier  $\geq$  2, Counter % 100 = 0, Zeit zwischen 18:00-22:00, Variable `mode` = `relaxed`, ...*
3. **Actions** — Was wird ausgeführt? Sound, Overlay, TTS, OBS-Befehl, Chat-Nachricht, Webhook, Counter setzen, Variable schreiben, Loyalty-Punkte vergeben.

Im Builder ziehst du das visuell zusammen, ohne Code.

## Routen / API

- `GET /v1/automations/` — alle Automations.
- `POST /v1/automations/` — neue Automation.
- `PUT /v1/automations/{id}` — bearbeiten.
- `DELETE /v1/automations/{id}` — löschen.

- `GET /v1/automations/templates/` — Templates-Library.
- `GET /v1/automations/variables/` — alle Variablen.
- `GET /v1/automations/counters/` — alle Counter.
- `GET /v1/automations/queues/` — Queue-Status.
- `POST /v1/automations/test/` — Automation isoliert testen (Dry-Run mit Mock-Event).
- `POST /v1/automations/webhooks/{id}` — Webhook-Receiver (eingehend von externen Tools).
- `POST /webhook/automations/{id}` — Public Webhook-Receiver-Endpoint.

## Service-Struktur (zum Verstehen)

Im Backend liegt die Logik in `Chatlix\Service\Automation\`:

- `Conditions/` — alle Bedingungs-Operatoren.
- `Actions/` — alle Action-Typen.
- `EventBinding/` — Verknüpfung Trigger → Automation.
- `Testing/` — Dry-Run-Engine.
- `Variables/` — Variable-Schreib-/Lese-Runtime.
- `Counters/` — Counter-Logik (mit Persistenz).
- `Queues/` — Throttling und Reihenfolge.
- `WebhookDispatcher/` — ausgehende HTTP-Calls.

Für dich relevant: Wenn was kaputt ist, im *AutomationDebug*-Panel siehst du, welche Stage geknallt hat.

## Internal Pipeline 200/ok:false

Wichtig zu wissen, falls du selber externe Webhooks anbindest: Chatlix' interne Worker→PHP-Endpoints geben **fachliche** Fehler mit `HTTP 200` + JSON `{"ok": false, "error": "..."}`  zurück,

nicht mit 4xx/5xx. Das verhindert dumme Retry-Loops in Workern. Wenn du selber eine externe API mit einem Webhook ansprichst, plan ähnlich — Chatlix retried bei 5xx auch von außen.

# Trigger-Typen im Überblick

Trigger	Beispiel
Twitch Event	Sub, Bits, Raid, Follow, Hype Train, Stream Online/Offline
Channel-Points-Reward	spezifischer Reward
Chat-Befehl	<code>!song</code>
Timer / Cron	alle 15 Minuten
Loyalty-Schwelle	User erreicht Level 10
Manueller Button	im Dashboard oder Stream Deck
Eingehender Webhook	externes Tool (Discord-Bot, eigene Tools)

## Was du auf den nächsten Seiten findest

- *Erste Automation bauen* — Schritt-für-Schritt eine simple Automation.
- *Cooldowns* — wie du Spam verhinderst.
- *Automation-Templates* — fertige Vorlagen ausrollen.

---

Version #2

Erstellt: 2026-05-11 21:27:49 CEST von Chatlix Admin

Zuletzt aktualisiert: 2026-05-15 00:50:47 CEST von Chatlix Admin