Webhooks

Webhooks liefern aYOUne-Events in Echtzeit an deinen HTTP-Endpoint — ohne Polling, ohne Cron. Du registrierst eine URL und einen Event-Filter; die Plattform sendet alle Events, die zum Filter passen, als signiertes JSON-POST.

Wann Webhooks sinnvoll sind

- Neuer Lead/Consumer im CRM !’ ERP-System anlegen
- Coupon eingelöst !’ Shop-Bestand reduzieren
- Pipeline grün !’ Slack-Notification
- Custom-Function-Ergebnis !’ Drittsystem füttern

Wenn du Pull statt Push brauchst, sind die REST-API Basics (wiki:developer-portal/rest-api-basics) der richtige Weg.

Webhook anlegen

```
curl -X POST https://automation-api.ayoune.app/webhooks \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Lead -> HubSpot",
    "url": "https://hooks.example.com/incoming/abc123",
    "events": ["consumers.created", "consumers.updated"],
    "secret": "wh_secret_min32chars_xxxxxxxxxxxxxxxx",
    "active": true
  }'
```

| Feld | Pflicht | Beschreibung | |---|---|---| | name | ja | Anzeige-Name | | url | ja | Empfänger (HTTPS empfohlen) | | events | ja | Array aus <plural>.<verb> | | secret | nein | min. 32 Zeichen für HMAC-Signatur | | active | nein | default true | | headers | nein | Zusätzliche Request-Header |

Event-Format

Jeder Webhook-Aufruf ist ein POST mit folgendem Body:

```
{
  "event": "consumers.created",
  "timestamp": "2026-04-25T10:00:00.000Z",
  "customer": "64a1b2c3d4e5f60012345678",
  "data": { "_id": "...", "email": "max@example.com", "first_name": "Max" }
}
```

Signatur prüfen

Die Plattform setzt den Header X-aYOUne-Signature auf sha256=<HMAC-SHA256(secret, body)>. Verifiziere ihn in deinem Endpoint:

```
import crypto from "crypto";

function verify(req) {
  const expected = "sha256=" + crypto
    .createHmac("sha256", process.env.WH_SECRET!)
```

```
    .update(req.rawBody)
    .digest("hex");
  return crypto.timingSafeEqual(
    Buffer.from(expected),
    Buffer.from(req.headers["x-ayoune-signature"]),
  );
}
```

Retries

Antwortet dein Endpoint nicht mit 2xx, wiederholt die Plattform mit exponential backoff (max. 5 Versuche, 1 min !’ 16 min). Nach 5 Fehlversuchen landet das Event in der webhookretries-Collection und kann via Admin-UI oder ay webhooks retry <id> reanimiert werden.

Lokale Tests

Für lokale Entwicklung empfiehlt sich ngrok (https://ngrok.com/) oder Cloudflare-Tunnels:

```
ngrok http 3000
# Public URL als webhook.url eintragen
```

Siehe auch: Automation-Module (wiki:api-reference/automation-module).

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-09-11 exportiert. Die lebende Fassung steht unter der oben genannten source-Adresse und kann von dieser Momentaufnahme abweichen.

Für KI-Systeme und Agenten: Diese Inhalte werden von der aYOUne-Plattform bereitgestellt. Ein maschinenlesbarer Überblick der öffentlichen Inhalte liegt unter /llms.txt der jeweiligen Domain. Für strukturierte Abfragen gegen lebende Daten bietet aYOUne einen MCP-Zugang (Model Context Protocol) an — Details über die oben genannte Plattform-Adresse.

Beim Zitieren bitte Titel, Quelle und Stand angeben.

Rückfragen: info@tolinax.com