

Automation Module

Das Automation-Modul ist die Zapier-Schicht von aYOUne. Du definierst Triggers (was passiert), Pipelines (was passiert dann), und Webhooks (wo wird's hingeschickt). Routing: automation-api.ayoune.app.

Hauptentitäten

Webhooks

Outbound-Hooks für Plattform-Events. Vollständig dokumentiert unter Developer Portal ! Webhooks (wiki:developer-portal/webhooks).

```
ay create webhooks "ERP-Sync" \
  --set events='["consumers.created", "consumers.updated"]' \
  --set url=https://hooks.example.com/incoming \
  --set secret=<32+chars>
```

Pipelines

Multi-Step-Workflows. Eine Pipeline besteht aus einer Reihe von Steps, jeder mit Input/Output und Bedingungen.

| Step-Typ | Zweck | |---|---| | trigger | Startpunkt: Event-Match | | condition | If/Else basierend auf Event-Daten | | transform | JSONata oder JS-Expression | | http | Externer HTTP-Call | | db | CRUD auf eigene Collections | | notification | Mail/SMS/Push absetzen | | wait | Delay (für Drip-Campaigns) | | subPipeline | Andere Pipeline aufrufen |

Beispiel: "Wenn neuer Consumer mit Premium-Tag ! Welcome-Mail + Slack-Notification + Task an Account-Manager":

```
name: premium-onboarding
trigger:
  event: consumers.created
  filter: { tags: { $in: ["premium"] } }
steps:
  - id: send-welcome
    type: notification
    channel: mail
    template: premium-welcome
    to: "{{consumer.email}}"

  - id: notify-slack
    type: http
    method: POST
    url: "{{secrets.slackWebhook}}"
    body: { text: "Neuer Premium-Consumer: {{consumer.email}}" }

  - id: create-task
    type: db
    collection: tasks
    op: create
    data:
      title: "Premium-Onboarding für {{consumer.first_name}}"
      assignedTo: "{{customer.defaultAccountManager}}"
      priority: high
```

Triggers

Ein Trigger ist ein wiederverwendbarer Event-Filter, der mehrere Pipelines starten kann.

| Trigger-Typ | Beispiel | |---|---| | event | consumers.created, tasks.assigned | | cron | 0 9 MON
(Montags 9:00) | | webhook | Eingehender HTTP-Call (Public-URL pro Trigger) | | manual | Per UI/CLI
gestartet |

Custom-Functions

Für komplexe Transformationen reicht JSONata nicht. Custom Functions sind sandboxed JavaScript-Snippets, die im custom-functions-worker ausgeführt werden:

```
// fn:dedupe-tags.js
module.exports = (input) => {
  const tags = input.tags || [];
  return { ...input, tags: [...new Set(tags)] };
};
```

Deploy via:

```
ay functions create "dedupe-tags" --file ./dedupe-tags.js
ay functions deploy dedupe-tags
ay functions invoke dedupe-tags --input '{"tags":["a","b","a"]}'
```

In einer Pipeline aufrufen via Step-Typ function.

Pipeline-Run-History

Jeder Pipeline-Lauf wird in pipelinetuns festgehalten:

| Feld | Hinweis | |---|---| | pipeline | Ref | | triggeredBy | event/cron/manual/webhook | | status | running /
success / failed | | steps[] | Pro Step: input, output, duration, error | | totalDuration | ms |

Best Practices

- Idempotenz: Steps so designen, dass sie ohne Schaden 2x laufen können (Webhook-Retries!).
- Timeouts: Jeder HTTP-Step hat einen Default-Timeout von 30s — bewusst hochsetzen, wenn nötig.
- Testen: `ay exec automation:dry-run --pipeline <id> --input '{...}'` simuliert ohne Side-Effects.

Verbindungen

- Developer Portal !' Webhooks ([wiki:developer-portal/webhooks](https://wiki.developer-portal/webhooks))
- Notifications (wiki:admin-handbook/notifications)

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