ay exec (Alias: x)

Führt eine registrierte API-Action aus, identifiziert über operationId. Ruft die apiactions-Registry des Servers ab, um die Action-Definition (Host, Endpoint, Method, Params) aufzulösen, dann der eigentliche HTTP-Call mit dem User-Token.

Syntax

```
ay exec <operationId> [args] [flags]
```

Beispiele

```
# Custom-Action für einen Consumer
ay exec sendWelcomeMail --param consumerId=64a1b2c3

# AI-Generate-Aufruf
ay exec aiGenerate --param prompt="Schreibe eine Willkommens-Email"

# Action ohne Args
ay exec recomputeKpis

# Body via JSON
ay x publishCampaign --body
'{"campaignId":"64a1...","scheduledAt":"2026-05-01T10:00:00Z"}'
```

Auflösungs-Reihenfolge

CLI fragt config-api/apiactions?q=<operationId> (für non-Admin User).

Bei 401/403 !’ Fallback auf su/apiactions (Superuser-Path).

Exact-Match auf operationId, sonst Partial-Match (erster Treffer).

Bei keinem Treffer !’ Exit 1; bei nur 403 in beiden Modulen !’ Exit 5.

Hinweis

Custom-Actions liegen unter /<collection>/:id/actions/<action-name>. Sub-Resource-GETs (z.B. /:id/logs) sind keine Actions — diese laufen via describe (wiki:cli/crud-core/describe) <collection> <id> <subResource>.

Siehe auch

- actions (wiki:cli/utilities/actions) — Action-Discovery
- curl (wiki:cli/misc/curl) — Direkter HTTP-Call

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG (haftungsbeschränkt) und wurde am 2026-11-26 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