ay functions (Alias: fns)

Lokale Entwicklung, Deployment und Verwaltung von Custom Functions (FaaS). Sandbox via isolated-vm, 5 Trigger-Types.

Subcommands

| Subcommand | Zweck | |---|---| | list | Deployed Functions mit Version, Status, letzter Aufruf | | get <name> | Function-Details + Code + Config | | create <name> | Scaffold lokal + Remote-Stub anlegen | | deploy <name> | Code + Config hochladen, neue Version aktivieren | | invoke <name> | Ad-hoc-Aufruf mit --data '<json>' | | delete <name> | Function entfernen | | rollback <name> <version> | Auf frühere Version zurück | | versions <name> | Version-History | | logs <name> | Letzte N Execution-Logs streamen |

Beispiel: Neue Function

```
ay functions create welcome-mail
# 'œ tooling/functions/welcome-mail/{index.ts, config.yaml}

# Code editieren, dann:
ay functions deploy welcome-mail
ay functions invoke welcome-mail --data '{"consumerId":"..."}'
ay functions logs welcome-mail --follow
```

Function-Config

```
# config.yaml
name: welcome-mail
description: Sendet Willkommens-Email an neuen Consumer
trigger:
  type: event           # oder: cron, webhook, manual, scheduled
  topic: consumers.created
runtime: node20
timeout: 10s
memory: 128MB
secrets:
  - SMTP_PASSWORD       # Credentials-Vault-Key
allowedEntities:
  - Consumers
  - Emails
```

Trigger-Types

| Type | Auslöser | |---|---| | event | Platform-Event (z.B. orders.paid) | | cron | Repeatable-Schedule (Cron-Expression) | | webhook | Externer HTTP-Aufruf | | manual | Per ay functions invoke | | scheduled | Einmaliger Zeitpunkt |

Platform-SDK

Inside einer Function steht ay.* zur Verfügung:

```
export default async function({ event, ay }) {
  const consumer = await ay.db('Consumers').findById(event.consumerId);
  await ay.email.send({ to: consumer.email, template: 'welcome' });
```

```
}
```

Der SDK ist opt-in-scoped: nur Entities in allowedEntities + mit availableInSDK=true in modelsAndRights.ts sind erreichbar (Stand 2026-04: 26 kuratierte Entries).

Siehe auch

- credentials (wiki:cli/misc/credentials) — Secrets verwalten
- jobs (wiki:cli/misc/jobs) — Function-Queues beobachten

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