

Config Module

Das Config-Modul ist die Schaltzentrale der Plattform: 165+ CRUD-Endpoints für Credentials, Feature-Flags, Notification-Channels, Computing-Entities, Module-States. Routing: config-api.ayoune.app.

Warum so viele Endpoints?

Config ist das Catch-All für Cross-Cutting-Settings. Statt jede Funktion in eigenes Modul zu zerlegen, leben hier zentrale Sub-Domänen:

| Sub-Domäne | Endpoints | |--|--| | Credentials & Secrets | credentials, apikeys, oauthtokens | | Feature-Flags | flags, flagrules, flaghistory | | Notifications | notificationchannels, notificationtemplates, notificationroutes | | Computing-Entities | computingentities, programs, installations | | State-Registry | ayounestates, modulestate-cache | | Translation | translations, locales | | Marketplace | marketplaceapps, bundles, installations | | Branding | themes, assets |

Credentials-Vault

Sensible Daten (API-Keys, OAuth-Tokens, SMTP-Passwörter) liegen verschlüsselt (AES-256-CTR) in credentials. Pro Customer separat.

```
ay credentials list
ay credentials get <id>
ay credentials create "Stripe Prod" --type stripe \
  --set credentials.secretKey=sk_live_XXX
ay credentials test <id>           # Provider-Roundtrip
ay credentials decrypt <id>        # Klartext (audit-protokolliert!)
ay credentials refresh <id>        # OAuth-Refresh
ay credentials revoke <id>
```

Felder:

| Feld | Hinweis | |--|--| | name | Display | | type | Provider-Slug (stripe, paypal, twilio, ...) | | credentials | Verschlüsselter Sub-Doc | | _status | active / expired / revoked | | lastTestedAt | Date | | lastTestStatus | ok / fail |

Rights: config.credentials.{view|create|edit|delete|decrypt|test}. decrypt und delete sollten nur Admins haben.

Feature-Flags

Bereits ausführlich beschrieben unter Admin-Handbook !' Feature Flags (wiki:admin-handbook/feature-flags). Endpoints:

```
GET    /flags
GET    /flags/:slug
POST   /flags
PATCH /flags/:slug
POST   /flags/:slug/actions/toggle
```

Notifications

11 Channels, n Templates, m Routes. Endpoints:

```
GET /notificationchannels
GET /notificationtemplates
GET /notificationroutes
POST /notificationroutes/:id/actions/test
```

Mehr unter Admin-Handbook !' Notifications ([wiki:admin-handbook/notifications](https://wiki.admin-handbook.com/notifications)).

Computing-Entities

Virtuelle Maschinen, Container, Cron-Boxen pro Customer. Wird vom desktop-client und programs-Service genutzt.

Feld	Hinweis	typ	name	Display	type	vm-aws	vm-hetzner	container	cron	awsState	running	stopped	terminated	ipAddress	string	tags	Routing/Cost-Center

State-Registry

ayounestates ist die zentrale Registry für UI-States — von `registerModuleStates(moduleName, states[])` aus dem Core gefüllt. Jeder State enthält:

- `_module` (Modul-Name)
- `appName` (Scope, default `AYOUNERELEASE_NAME`)
- `name` (z.B. `crm.consumers`)
- `stateType` (module-overview, table, right, ...)
- `cols`, `filters`, `actions`, `links`

Mehr unter Self-Hosting Setup ([wiki:self-hosting/getting-started](https://wiki.self-hosting.com/getting-started)) — Pods schreiben das beim Boot.

Translation

translations enthält alle UI-Strings + Block-Texte:

Feld	Hinweis	key	dot.notation	locale	de, en, ...	value	string	module	Scope

CLI-Helfer:

```
ay exec translations:export --locale de --out ./de.json
ay exec translations:import --file ./de.json
```

Verbindungen

- Admin-Handbook !' Feature Flags ([wiki:admin-handbook/feature-flags](https://wiki.admin-handbook.com/feature-flags))
- Admin-Handbook !' Notifications ([wiki:admin-handbook/notifications](https://wiki.admin-handbook.com/notifications))
- Self-Hosting !' Environment Variables ([wiki:self-hosting/environment-variables](https://wiki.self-hosting.com/environment-variables))

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