

## ay login (Alias: auth)

Authentifiziert die CLI gegen die aYOUne-Plattform. Standard ist Browser-basiertes Login via WebSocket-Notifier. Alternativ akzeptiert die CLI ein bestehendes JWT.

### Syntax

```
ay login [--token <jwt>]
```

### Verhalten — Browser-Login

CLI generiert ein temporäres Token, öffnet `https://auth.ayoune.app/login?cliToken=...`

Im Browser anmelden (Email/Passwort, WebAuthn, OAuth, ...).

Nach Erfolg pusht der Browser das Session-JWT via `notifier.ayoune.app` zurück.

CLI persistiert es AES-256-CBC-verschlüsselt in `~/aYOUne/storage/`.

### Verhalten — JWT-Login

```
ay login --token eyJhbGciOi...
```

Validiert das JWT-Shape, decodiert es und schreibt es in `secureStorage`. Nutzt nicht den Browser-Flow.

### Beispiele

```
# Standard  
ay login
```

```
# Aliased  
ay auth
```

```
# Per Token (z.B. CI-Pipeline)  
ay login --token "$AYOUNE_TOKEN"
```

### Token-Reihenfolge in der CLI

`--token <jwt>` Flag (CLI-Parameter)

`secureStorage.token` (durch `ay login` befüllt)

`AYOUNE_TOKEN` Environment-Variable

### Siehe auch

- `logout` (<wiki:cli/auth-config/logout>) — Token löschen
- `whoami` (<wiki:cli/auth-config/whoami>) — Aktiven User prüfen
- `storage` (<wiki:cli/crud-core/storage>) — Token-Inspector

---

## Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-10-11 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](mailto:info@tolinax.com)