

# Users & Roles

Das User-Management von aYOUne kombiniert klassische Rollen mit individuellen User-Rights und einer Paket-Schicht. Wer das Modell versteht, kann jeden noch so feinen Berechtigungsfall abbilden.

## Das Rechte-Modell in einem Bild

```

Customer.packages[]           !• was ist gebucht (additiv)
    "* Customer.customRights[]
        %
        !" Schnittmenge
Role.rights "* User.customRights.rights
        %
        !"
        Effektive Rechte

```

Wichtig: Pakete sind additiv, nicht restriktiv. Wenn dein Reseller ein Premium-Paket bucht, kannst du nur mehr Rechte vergeben, nicht weniger als die Rolle hergibt. Ein Tenant, der "nur CRM" verkaufen soll, braucht entweder ein getrimmtes Paket oder eine zugeschnittene Rolle.

## User anlegen

In der UI: admin-v2 !' Users !' "+ Neuer User". Per CLI:

```

ay create users "max@firma.tld" \
  --set first_name=Max \
  --set last_name=Hanrieder \
  --set role=<roleId>

```

Pflichtfelder:

| Feld | Beschreibung | |---|---| | email | Login-Identität | | firstname / lastname | Anzeige-Name | | role | ObjectId aus aYOUneRoles | | language | de, en, ... |

Nach dem Anlegen erhält der User eine Begrüßungsmail mit Passwort-Reset-Link (sofern in Notifications (wiki:admin-handbook/notifications) konfiguriert).

## Role-Editor

Roles bündeln Rechte zu wiederverwendbaren Sets. Der Modern Role Editor (admin-v2 v2026.x) bietet:

- Tabs pro Modul (CRM, Marketing, CMS, ...)
- Live-Suche über alle Rechte
- Default-Templates (Sales-Team, Support, Read-Only) als Startpunkt
- Diff-View beim Speichern: was ändert sich für betroffene User?

Ein Right hat das Format <modul>.<entity>.<verb> — z.B. crm.consumers.view, marketing.coupons.create. Drei-Teil-Rechte werden in ayounestates als stateType:"right" gespiegelt.

## Custom-Rights pro User

Manche User brauchen eine Sonderlocke ohne eigene Rolle. Dafür gibt es User.customRights.rights[]:

```

ay update users <id> \

```

```
--set customRights.rights='["marketing.coupons.export"]'
```

Diese werden zur Rolle addiert (nicht ersetzt). Sichtbar im User-Profil unter "Zusätzliche Rechte".

## Best Practices

- Wenig Rollen, viele User: 5-10 wohldurchdachte Rollen schlagen 100 individuelle Rechte-Sets.
- Templates nutzen: Statt jede Rolle from scratch zu bauen, dupliziere ein globales Default und passe an.
- Audit aktivieren: Jede Rechteänderung landet im Audit-Trail ([wiki:admin-handbook/audit-trail](https://wiki.tolinax.com/admin-handbook/audit-trail)) — nutze das.
- Service-Tokens für Integrationen: Niemals einen normalen User für Webhook-Auth missbrauchen. Siehe CORS & Auth ([wiki:developer-portal/cors-and-auth](https://wiki.tolinax.com/developer-portal/cors-and-auth)).

## Deaktivieren statt löschen

Inaktive Mitarbeiter setzt du auf `_status: "inactive"` — die Login-Versuche werden geblockt, Audit-Spuren bleiben aber lesbar. Hartes Löschen sollte nur für Test-Accounts genutzt werden.

```
ay update users <id> --set _status=inactive
```

---

## 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)