

# Benutzer mit externen Systemen synchronisieren

Grundsätzlich gibt es mehrere Möglichkeiten, um Benutzer in FabAccess anzulegen:

- über eine `users.toml` Datei
- direkt über die Client App (als Administrator)
- über die API. Hierzu gibt es eine Referenzimplementation [pyfabi](#) mit einem Beispielscript `add_del_user.py`

## Hinweise zum Arbeiten mit users.toml Dateien

Wer mit einer automatisierten Erstellung einer `users.toml` aus externen Nutzerquellen arbeitet, der sollte folgenden Workflow erarbeiten:

1. BFFH Server herunterfahren
2. Aktuelle Benutzerkonfiguration per `/usr/bin/bffhd --dump-users users.toml` sichern
3. die soeben gesicherte `users.toml` Datei benutzen, um die aktuellsten Änderungen an der Benutzerbasis hinzuzufügen, zu löschen oder zu ändern. Hier muss selbst entschieden werden, welche Benutzerdaten aktuell sein sollen: Die durch die App geänderten Daten, oder die aus der externen Quelle? Eine von beiden muss die andere überschreiben bzw. überlagern. Entsprechend muss das verwendete Script programmiert sein: bei Vorhandensein von Nutzern, Rollen, Passwörter, Cardkeys rückfragen, ob bei Differenzen der bestehende oder der neue Datensatz verwendet werden soll oder per Config-Parameter automatisiert durchführen.
4. die neue `users.toml` Datei laden: `/usr/bin/bffhd --load-users users.toml` (siehe auch [Nutzerdatenbank laden / hashen / prüfen](#))
5. BFFH Server starten

Unter Umständen ist die Arbeit mit der [pyfabapi](#) sinnvoller!

## LDAP Anbindung

Hierzu gibt es einen [dedizierten Artikel](#).

## Authentik

Nutzer geben ein zweites Passwort an und ein Skript checkt jede Minute Authentik ob es neue Nutzer gibt, sich Gruppen/Passwort geändert haben und nimmt entsprechend Änderungen über die FabAccess API vor. Das Passwort wird dann aus den Nutzerattributen gelöscht. Die Daten liegen in einer MySQL Datenbank.

Mögliche Ansprechpartner dafür könnten das [Creative Lab der Hochschule München](#) oder Michael Prange vom [MakerSpace Gütersloh](#) sein!

## CiviCRM

CiviCRM erlaubt den Export von Nutzern als CSV-Tabelle. Diese kann verwendet werden, um Nutzer in FabAccess zu importieren.

Roy Böttcher vom [MakerSpace Leipzig](#) fragen!

## VereinOnline

Nutzer und Gruppen aus [VereinOnline](#) lassen sich mit dem Tool [csv-to-fabaccess-user-toml](#) vom [Sternenlabor Plauen](#) direkt als `users.toml` ausgeben. Dazu wird eine Datei namens `Mitglieder.csv` mit Spaltenseparator `;` und den Spalten `Login` und `Gruppen` (mit Separator `,`) erzeugt. Für das Tool muss [Node.js](#) installiert sein.

```
cd ~
git clone https://github.com/Sternenlabor/csv-to-fabaccess-user-toml.git
cd csv-to-fabaccess-user-toml/

sudo apt install nodejs

npm install csv-parser
```

Mitglieder-CSV Datei - Beispiel:

<https://docs.fab-access.org/data-csv-to-fabaccess-user-toml/Mitglieder.csv>

```
# Ausführen
node run.mjs
```

```
[
  { Login: 'Anton', Gruppen: 'Admins' },
  { Login: 'Peter', Gruppen: 'Mitglieder,Admins' },
  { Login: 'Jonny', Gruppen: 'Mitglieder' },
  { Login: 'Sarah', Gruppen: 'Mitglieder' }
]
users: 4
File created successfully
```

## Beispiel-Output:

```
cat user.toml
```

```
[Admin1]
roles = [ "Admin" ]
passwd = "secret"

[Anton]
roles = [ "User", "Admins" ]
passwd = "secret"

[Peter]
roles = [ "User", "Mitglieder,Admins" ]
passwd = "secret"

[Jonny]
roles = [ "User", "Mitglieder" ]
passwd = "secret"

[Sarah]
roles = [ "User", "Mitglieder" ]
passwd = "secret"
```

# easyVerein

Für das Programm easyVerein sind noch keine Schnittstellen in der Community bekannt. Jedoch nutzen es viele offene Werkstätten.

Zuletzt aktualisiert: 2025-03-03 15:03:59 CET von Mario Voigt (Stadtfabrikanten e.V.)