


# Anleitung

bis v1.25.x



Open Source Software
entwickelt durch JK Effects
von Julian Krauser

16. November 2026

# Inhaltsverzeichnis

Inhaltsverzeichnis ..... II
1 Einleitung ..... 1
2 Installation ..... 2
2.1 Docker ..... 2
2.2 Docker-Compose ..... 3
2.3 Git ..... 7
2.4 Datenbank ..... 9
2.5 Konfiguration ..... 10
2.6 Serverzustand ..... 15
2.7 Secret-Rotation ..... 15
2.8 Logging ..... 16
2.9 Health-Checks ..... 17
2.10 API-Dokumentation (OpenAPI) ..... 17
2.11 Dateiablage ..... 18
2.12 Update der Version ..... 19
2.12.1 Support-Linien ..... 19
2.13 WebApp ..... 20
2.14 Einrichtung ..... 20
3 Konzepte ..... 22
3.1 Stammdaten ..... 22
3.2 Berechtigungen ..... 22
3.3 Zeitformate ..... 23
3.3.1 Gültigkeitsdauer ..... 23
3.3.2 Zeitintervalle ..... 23
3.4 Rechtschreibprüfung ..... 24
3.5 Dateien und Bilder ..... 24
3.6 Engines ..... 25
3.6.1 Notification-Engine ..... 25
3.6.2 Template-Engine ..... 25
3.6.3 Query-Engine ..... 26
4 Module ..... 27
4.1 Mitgliederverwaltung & Stammdaten ..... 27
4.1.1 Nachweisdokumente ..... 30
4.2 Kalender ..... 31

4.3 Protokolle ... 34
4.4 Newsletter ... 36
4.5 Führerscheinkontrolle ... 38
4.6 Nachweisverwaltung (Requirements & Compliance Tracking) ... 40
4.6.1 Nachweis-Vorschriften ... 40
4.6.1.1 Zuordnungen ... 42
4.6.1.2 Zuordnungsregeln ... 42
4.6.2 Personennachweise ... 43
4.6.2.1 Nachweistermine ... 45
4.7 Geräte- & Fahrzeug- & Kleidungsverwaltung ... 46
4.7.1 Objektfilter ... 47
4.7.2 Geräte, Fahrzeuge & Kleidung ... 47
4.7.3 Verbrauchsmaterial ... 49
4.7.4 Schadensmeldungen & Reparaturen ... 50
4.7.5 Wartungen ... 51
4.7.6 Prüfungen ... 54
4.8 Query Builder & Query Store ... 58
4.8.1 Aufbau einer Abfrage ... 58
4.8.2 Bedingungen im Detail ... 59
4.8.3 Verknüpfte Tabellen im Detail ... 60
4.8.4 Datenstruktur ... 60
4.8.5 Beispiel ... 61
4.9 Templates & Template Builder ... 62
4.9.1 Template Builder ... 63
4.9.2 Bausteine ... 64
4.9.3 Verschachtelungsregeln ... 65
4.9.4 Variablen & Pipes ... 65
4.9.5 Bedingungen ... 66
4.9.6 Render-Kontext ... 67
4.9.7 Bildergalerie ... 67
4.9.8 Template-Verwendung ... 68
4.10 Listen-Druck ... 69
4.11 Datenimport (Massendaten) ... 70
4.11.1 Import von Terminen ... 71
4.12 Anwendungsspezifische Einstellungen ... 73
4.13 Modulbezeichnungen ... 74
4.13.1 Bezeichnungen ... 75
4.13.2 Module deaktivieren ... 75

4.13.3 Abhängigkeiten ..... 75
4.13.4 Gesperrte Module ..... 76
4.14 Benutzerverwaltung ..... 76
4.14.1 Benutzer einladen ..... 77
4.14.2 Verwaltete Konten ..... 78
4.14.3 Konten deaktivieren ..... 79
4.14.4 Details ..... 79
4.14.5 Berechtigungen ..... 80
4.14.6 Rollen ..... 80
4.15 Rollenverwaltung ..... 81
4.15.1 Details ..... 82
4.15.2 Berechtigungen ..... 82
4.16 WebApi ..... 83
4.16.1 Details ..... 84
4.16.2 Berechtigungen ..... 85
4.16.3 WebApi Zugriffsablauf ..... 85
4.16.4 API-Clients ..... 86
4.17 Backups ..... 87
4.17.1 Backup wiederherstellen ..... 88
4.17.2 Daten exportieren ..... 91
4.17.3 Backup hochladen ..... 92
4.18 Logs ..... 93
4.18.1 Ansicht ..... 94
4.18.2 Protokollierung ändern ..... 94
4.19 Anwendungsinformationen und Versionsmanagement ..... 95
4.19.1 Anwendungsinformationen ..... 95
4.19.2 Versionsmanagement ..... 96
4.20 Modulspezifische Einstellungen ..... 97
5 Benutzerbereich und Accounteigenschaften ..... 99
5.1 Zugang und Anmeldedaten ..... 99
5.1.1 Anmeldeoptionen ..... 99
5.1.2 Änderung der Anmeldedaten ..... 100
5.2 Benachrichtigungen ..... 101
5.2.1 Benachrichtigungsanzeige ..... 101
5.2.2 Benachrichtigungseinstellungen ..... 101
5.2.3 Geplante Benachrichtigungen ..... 102
5.3 Push-Benachrichtigungen ..... 103
5.3.1 Funktionsweise ..... 103

5.3.2 Push aktivieren ... 103
5.3.3 Übersicht und Verwaltung ... 104
5.3.4 Deaktivieren und Löschen ... 104
5.3.5 Lebenszyklus einer Push-Subscription ... 105
5.3.6 Mehrere Benutzer auf einem Gerät ... 105
5.4 Sessionverwaltung ... 105
5.4.1 Geräteerkennung und Datenschutz ... 106
5.4.2 Verwaltung aktiver Sessions ... 106
5.5 Passkeys ... 106
5.5.1 Erstellung eines Passkeys ... 107
5.5.2 Login mit Passkeys ... 107
5.6 Übertragung Administration ... 108
5.6.1 Durchführung ... 108
5.7 Wissensartikel ... 109
5.7.1 Anlegen und Bearbeiten ... 109
5.7.2 Freigaben ... 109
5.7.3 Löschen ... 110
6 Ökosystem FF Admin ... 111
7 Roadmap ... 112

# 1 Einleitung

**FF Admin - Die zentrale Verwaltungssoftware für Feuerwehren und Vereine**

FF Admin ist eine vielseitige Mitgliederverwaltungssoftware, die als Herzstück eines wachsenden Ökosystems dient. Neben der Mitgliederverwaltung ermöglicht das Programm die Organisation von Terminkalendern, die Erstellung von Newslettern und Protokollen sowie die Verwaltung von Gerätschaften und Prüfplänen.

Obwohl FF Admin in erster Linie für Feuerwehren konzipiert ist, kann es dank seines modularen Aufbaus auch für andere Organisationen angepasst werden. Die frei definierbaren Stammdaten ermöglichen einen flexiblen Einsatz, so dass die Software optimal an die individuellen Bedürfnisse angepasst werden kann.

# 2 Installation

FF Admin kann über mehrere Wege betrieben werden. Zum einen werden Docker-Images versioniert zur Verfügung gestellt. Weiterhin kann auch das Release Projekt heruntergeladen und verwendet werden.

## 2.1 Docker

**Disclaimer:** Die Anleitung zum Betrieb von FF Admin mit Docker setzt Kenntnisse mit Docker und Docker-Compose voraus.

Die Docker-Images können gemeinsam über eine Compose-File konfiguriert und gestartet werden. Auch können die Images einzeln gestartet werden.

Die Docker-Images sind versioniert. Für den `<tag>` des Images stehen folgende Varianten zur Verfügung:

- `vX` (z.B. `v1`): neueste Version der Hauptversion. **Empfohlen**, da alle neuen Funktionen und Fehlerbehebungen ohne grundlegende Umstellungen enthalten sind.
- `vX.Y` (z.B. `v1.6`): neueste Patch-Version einer Support-Linie (Abschnitt 2.12.1). Geeignet, um bewusst auf einem Funktionsstand zu bleiben und trotzdem Sicherheits- und Fehlerbehebungen zu erhalten.
- `vX.Y.Z` (z.B. `v1.6.4`): genau diese Version, z.B. für ein Rollback.
- `latest`: höchste veröffentlichte Version insgesamt, auch über eine neue Hauptversion hinweg.

Jeder Tag existiert zusätzlich mit den Endungen `-arm` und `-amd` für eine einzelne Architektur. Die Versionen können auch in den Releases der Repositories der Anwendungen nachgeschlagen werden. Dort lassen sich auch auch Informationen zu neuen Funktionen, Änderungen oder Fehlerbehebungen der jeweiligen Funktion finden.

## 2.2 Docker-Compose

### App

```yaml
ff-admin-app:
  image: code.jk-effects.cloud/ff-admin/ff-admin/app:<version | latest>
  container_name: ff_admin
  restart: unless-stopped
  ports:
    - "80:80"
  environment:
    - SERVERADDRESS=<backend_url> # optional, bei abweichender URL des Backends
    - FRAME_ANCESTORS=<erlaubte Einbett-Origins, leerzeichengetrennt> # optional
```

Die Verwendung der Werte des Typs Environment werden unter dem Punkt Konfiguration (Abschnitt 2.5) erklärt.
Die Environment-Variable `SERVERADDRESS` ist optional und kann für eine Abweichende URL des Backends verwendet werden. Ist die Serveradresse nicht angegeben, wird versucht das Backend unter der selben URL mit dem PathPrefix `/api` zu erreichen.

Die Environment-Variable `FRAME_ANCESTORS` ist optional und standardmäßig leer. Dadurch darf FF Admin ausschließlich von sich selbst per `<iframe>` eingebettet werden (Content-Security-Policy `frame-ancestors 'self'`). Um das Einbetten auf anderen Webseiten zu erlauben, können die erlaubten Origins mit vollständigem Schema (`https://...`), ohne Pfad und leerzeichengetrennt angegeben werden, z.B. `https://verein.de https://www.verein.de`.

Eine erweiterte Personalisierung der App mit eigenem Logo der Feuerwehr oder des Vereins ist bei der ersten Einrichtung oder unter dem Modul Einstellungen möglich. Hiervon betroffen ist das Icon im Browser-Tab, jede Anzeige des FF Admin Logos innerhalb der App und das Icon, wenn die WebApp auf einem Gerät installiert wird.

### Server

```yaml
ff-admin-server:
  image: code.jk-effects.cloud/ff-admin/ff-admin/server:<version | latest>
  container_name: ff_admin_server
  restart: unless-stopped
  ports:
    - "5000:5000"
  environment:
    - DB_HOST=<database host>
    - DB_PORT=<database port> # optional, da default 5432
    - DB_SCHEMA=<database schema> # optional, da default public
    - DB_NAME=<database name>
    - DB_USERNAME=<database username>
    - DB_PASSWORD=<database password>
    - APPLICATION_SECRET=<application_secret> # einmalig generieren, Wechsel nur über den Rotationsmodus (siehe Handbuch)
    - APPLICATION_SALT=<application_salt> # einmalig generieren, Wechsel nur über den Rotationsmodus (siehe Handbuch)
    - SERVER_URL=<server_url>
    - CLIENT_URL=<server_url>
    # \/ optional \/
    - JWT_SECRET=<jwt_secret> # signiert Sitzungen und WebAPI-Tokens statt APPLICATION_SECRET
    - APPLICATION_STORAGE=<application_storage>
    - USE_SECURITY_STRICT_LIMIT=<boolean>
    - SECURITY_STRICT_LIMIT_WINDOW=<time window>
    - SECURITY_STRICT_LIMIT_REQUEST_COUNT=<strict_request_count>
    - USE_SECURITY_LIMIT=<boolean>
    - SECURITY_LIMIT_WINDOW=<time window>
    - SECURITY_LIMIT_REQUEST_COUNT=<request_count>
    - TRUST_PROXY=<proxy config>
    - LOG_LEVEL=<error|warn|info|debug>
    - LOG_MAX_FILES=<count | days>
    - LOG_MAX_SIZE=<size>
    - LOG_SLOW_QUERY_MS=<milliseconds>
    - FILE_ORPHAN_REPORT_ONLY=<true|false>
    - FILE_KEY_VERSION=<number>
    - BACKUP_UPLOAD_MAX_SIZE=<megabytes>
    - API_DOCS=<off|public>
  volumes:
    - <volume|local path>:/app/files
    - <volume|local path>:/app/logs # optional, ohne Mount gehen die Logs beim Neuerstellen des Containers verloren
```

Die Verwendung der Werte des Typs Environment werden unter dem Punkt Konfiguration (Abschnitt 2.5) erklärt.

Environment Werte können optional sein oder haben Standard-Werte.
Das Fehlen einer geforderten Variable oder die falsche Angabe eines Variablen-Werts verhindert das Starten des der Anwendung.

Innerhalb dem Ordner, der dem Volume zugeordnet ist, werden Backups und Ausdrucke der geschriebenen Protokolle, Newsletter und alle weiteren Dokumente abgelegt, die hochgeladen oder erstellt werden können. Unter `/app/logs` schreibt der Server seine Log-Dateien (siehe Abschnitt 2.8). Der Mount dieses Ordners ist optional: Ohne Volume funktioniert das Logging ebenfalls, die Dateien gehen aber verloren, sobald der Container neu erstellt wird (z.B. bei einem Update).

**Datenbank**
Als Datenbank wird Postgres verwendet:

```yaml
ff-db:
  image: postgres:<version (bsp 16)>
  container_name: ff_db
  restart: unless-stopped
  ports:
    - "5432:5432"
  environment:
    - POSTGRES_DB=<database name>
    - POSTGRES_USER=<username>
    - POSTGRES_PASSWORD=<user password>
  volumes:
    - <volume|local path>:/var/lib/postgresql/data
```

`POSTGRES_DB` erstellt direkt eine Datenbank, die durch einen angelegten `POSTGRES_USER` verfügbar ist.

**Hinweis** Wenn eine Docker-Compose Datei verwendet wird, kann zusätzliche ein Netzwerk angelegt werden. Dadurch ist das Veröffentlichen der Datenbank-Ports nicht mehr notwendig. Das Entfernern der Port-Exposes verhindert den direkten Zugriff auf die Ports von außerhalb.
Hierfür muss das verwendete `network` dem Backend und dem Datenbank-Container bekanntgegeben werden:

1. Ergänzung zu Server und Datenbank Container:

```yaml
#volumes:
#  ...
networks:
  - ff_internal
```

2. Ergänzung zur Docker-Compose:

```yaml
networks:
  ff_internal:
```

3. Neue Host-URL der Datenbank des Servercontainers:

```yaml
environment:
  - DB_HOST: ff-db
  # ...
```

Wenn die Datenbank über ein Netzwerk in einer Compose-Datei freigegeben wird, kann als Host der Service-Name der Datenbank angegeben werden. In der angegebenen Datenbank-Konfiguration wäre das `ff-db`.

Optionale Ergänzung zum Servercontainer:

```yaml
#volumes:
#  ...
#networks:
#  - ff_internal
depends_on:
  - ff-db
```

Hierdurch kann der Server nicht starten, wenn die verwendete Datenbank nicht läuft.

**Proxy**

Damit die App und der Server aus dem Internet über URLs erreichbar sind, kann Traefik oder Nginx als Reverse-Proxy verwendet werden.
Bei der Konfiguration des Proxies ist Folgendes zu beachten:

- Die Tatsächliche IP sollte an den Server weitergeleitet werden
- Alle HTTP-Header müssen an den Server weitergeleitet werden
- Die maximale Größe für Dateiuploads sollte nicht zu stark eingeschränkt werden

Die Weiterleitung der Header kann im Serverlog überprüft werden. API-Anfragen werden im Format `:remote-addr :method :url :status - :response-time ms | isPWA: :headers["x-pwa-client"] - device: :headers["x-device-name"]` protokolliert. Fehlen die Werte nach „isPWA:“ oder „device:“, werden die entsprechenden Header nicht weitergeleitet.

**Hinweis:** Bei Anfragen von externen APIs oder beim Zugriff auf Logo bzw. Favicon existieren keine benutzerdefinierten Header, da diese nur von der WebApp über den Browser gesendet werden.

## 2.3 Git

Eine Alternative zu Docker ist die direkte Ausführung der Anwendungen auf dem Server oder einem Desktop Gerät.
Hierzu müssen die App und der Server als Quellcode auf das ausführende System geladen und dort direkt verwendet werden. Das System muss NodeJs 24 und Postgres installiert haben.

Wie bei den Docker-Tags kann auch im Git ein Versionsstand gewählt werden, der mit `git checkout <branch/tag>` ausgecheckt wird:
- `main`: neueste veröffentlichte Version, entspricht dem Docker-Tag `latest`.
- `release/X.Y` (z.B. `release/1.6`): neueste Patch-Version einer älteren Support-Linie (Abschnitt 2.12.1), entspricht dem Docker-Tag `vX.Y`. Die aktuelle Linie liegt in `main`.
- `vX.Y.Z` (z.B. `v1.6.4`): Git-Tag für genau diese Version, z.B. für ein Rollback.

Der Branch `develop` enthält den Entwicklungsstand und ist nicht für den Betrieb gedacht.

Um die Konfiguration mittels ENV-Variablen an die Anwendungen weitergeben zu können, müssen `.env` Dateien erstellt werden. Hierzu kann die `.env.example` Datei kopiert und die definierten Werte ausgefüllt werden. Nicht benötigte Einträge sollten entfernt werden.
Die env-Datei im Frontend muss vor dem build-Prozess erstellt sein, da dort die Werte fest in den Code übernommen werden. Weiterhin muss die Datei im Frontend `.env.production` heißen. Die bestehende Datei kann modifiziert werden. Bei einer Änderung muss die App neu gebaut werden.
Wird der Server unter der gleichen URL wie die App verwendet, muss in der `.env.production` Datei des Frontends der Wert zum Eintrag von `SERVERADDRESS` auf den leeren String gesetzt werden.
Die env-Datei im Backend muss vor der Ausführung von `npm run start` angelegt sein. Bei einer Änderung der Einträge muss der Server lediglich neu gestartet werden.

Die NodeJs Prozesse können auch durch Tools wie pm2 verwaltet werden.
Damit die App und der Server aus dem Internet erreichbar sind, muss das Routing gesondert eingerichtet werden.

#### App

```shell
git clone https://code.jk-effects.cloud/ff-admin/ff-admin.git
cd ff-admin
npm install
npm run build
```

Der mit `npm run build` erstellte dist Ordner kann mit Apache oder Nginx zur veröffentlicht werden.

Für das Hosting der statischen Inhalte der App kann Apache oder Nginx verwendet werden. Eine Konfiguration für Nginx ist im Repo der App enthalten:

```conf
worker_processes 4;

events { worker_connections 1024; }

http {
  include mime.types;
  server_tokens off;

  server {
    root /usr/share/nginx/html; # Pfad zum dist Ordner des Frontends
    index index.html;

    # __SERVERADDRESS__ env var used for the app's own API base URL
    # __SERVERADDRESS_WS__ the wss:// (or ws://) equivalent of __SERVERADDRESS__
    # __FRAME_ANCESTORS__ space-separated origins allowed to embed this app in <iframe>
    add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'wasm-unsafe-eval'; style-src 'self' 'unsafe-inline'; img-src 'self' data: blob: __SERVERADDRESS__; font-src 'self' data:; connect-src 'self' __SERVERADDRESS__ __SERVERADDRESS_WS__ https://lang.ff-admin.de; frame-src 'self' blob:; manifest-src 'self' __SERVERADDRESS__; object-src 'none'; base-uri 'self'; form-action 'self'; frame-ancestors 'self' __FRAME_ANCESTORS__" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header Referrer-Policy "strict-origin-when-cross-origin" always;
    # Only enable if this app is always served over HTTPS
    # add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;

    location / {
      try_files $uri $uri/ /index.html;
    }
  }
}
```

Da die Nginx Konfiguration den Zugriff und die Einbettung einschränken, müssen die Variablen `__SERVERADDRESS__` und `__FRAME_ANCESTORS__` manuell angepasst werden.

### Server

```Shell
git clone https://code.jk-effects.cloud/ff-admin/ff-admin-server.git
cd ff-admin-server
npm install
npm run build
npm run start
```

Die Log-Dateien werden im Ordner `logs` innerhalb des Arbeitsverzeichnisses des Servers abgelegt (siehe Abschnitt 2.8).

**Proxy**
Damit die App und der Server aus dem Internet über URLs erreichbar sind, kann Traefik oder Nginx als Reverse-Proxy verwendet werden.
Bei der Konfiguration des Proxies ist Folgendes zu beachten:
- Die Tatsächliche IP sollte an den Server weitergeleitet werden
- Alle HTTP-Header müssen an den Server weitergeleitet werden
- Die maximale Größe für Dateiuploads sollte nicht zu stark eingeschränkt werden

Die Weiterleitung der Header kann im Serverlog überprüft werden. API-Anfragen werden im Format `:remote-addr :method :url :status - :response-time ms | isPWA: :headers["x-pwa-client"] - device: :headers["x-device-name"]` protokolliert. Fehlen die Werte nach „isPWA:“ oder „device:“, werden die entsprechenden Header nicht weitergeleitet.

**Hinweis:** Bei Anfragen von externen APIs oder beim Zugriff auf Logo bzw. Favicon existieren keine benutzerdefinierten Header, da diese nur von der WebApp über den Browser gesendet werden.

## 2.4 Datenbank

Die Anwendung FF Admin unterstützt ausschließlich PostgreSQL als Datenbank, da diese Vorteile bei der Migrationsverwaltung und Rollbacks in Transaktionen mit sich bringt. Es wird empfohlen die Datenbank stets auf der neuesten Version zu halten, um auch deren neueste Features nutzen zu können.
Mit Version 1.6.x der Core-Packages wurde eine datenbank-versionsabhängige Funktionalität eingeführt, welche die Verfügbarkeit der lokalen UUID Funktionen betrifft. So wird ab PostgreSQL v18 die Generierung von UUIDv7 ermöglicht, welches Optimierungen bei dem Einfügen von Daten mit sich bringt. FF Admin prüft bei jedem Start des Servers die Datenbankversion und passt die Generierung der UUIDs intern an.

## 2.5 Konfiguration

Folgende Werte können zu einem Container konfiguriert werden:

| Variable | Zweck | Default | Optional |
|---|---|---|---|
| ⬇️ App-Variablen | | | |
| SERVERADDRESS | URL, über welche das Backend erreicht werden kann. Die URL muss mit http:// oder https:// starten. Jeder angegebene Pfad wird automatisch entfernt. Die App versucht das Backend unter <SERVERADDRESS>/api zu erreichen. Laufen Backend und App auf der gleichen URL, kann diese Variable weggelassen werden. | | ✅ |
| FRAME_ANCESTORS | Leerzeichengetrennte Liste erlaubter Origins (vollständiges Schema, ohne Pfad), die die App zusätzlich zu sich selbst per <iframe> einbetten dürfen (CSP frame-ancestors). | leer (nur 'self') | ✅ |
| | | | |
| ⬇️ Server-Variablen | | | |
| DB_HOST | URL zur Datenbank | | 💥 |
| DB_PORT | Port der Datenbank | 5432 | ✅ |
| DB_SCHEMA | Schema der Datenbank | public | ✅ |
| DB_NAME | Name der Datenbank | | 💥 |
| DB_USERNAME | Nutzername für Zugang zu Datenbank | | 💥 |
| DB_PASSWORD | Passwort zum Zugang zur Datenbank | | 💥 |
| APPLICATION_SECRET | Zufällige Zeichenkette zur Validierung der Session-Tokens und Kodierung von Passwörtern und Zugangsdaten. | | 💥 |

| Variable | Zweck | Default | Optional |
|---|---|---|---|
| APPLICATION_SALT | Zufällige Zeichenkette zur Kodierung von Zugangsdaten. | | ✴️ |
| JWT_SECRET | Zufällige Zeichenkette zum Signieren der Session- und WebAPI-Tokens. Ist sie nicht gesetzt, wird das APPLICATION_SECRET verwendet. Ein Wechsel meldet alle Nutzer ab und macht WebAPI-Tokens ungültig. | | ✅ |
| SECRET_ROTATION | Startet den Rotationsmodus mit true oder dessen Rückabwicklung mit rollback (siehe @secret-rotation). | false | ✅ |
| APPLICATION_SECRET_OLD | Bisheriges APPLICATION_SECRET, nur für den Rotationsmodus. | | ✅ |
| APPLICATION_SALT_OLD | Bisheriges APPLICATION_SALT, nur für den Rotationsmodus. | | ✅ |
| APPLICATION_STORAGE | Angabe des verfügbaren Speichers, alternativ zur berechneten Größe des freien Festplattenspeichers. Fromat: [0-9]*(B\|KB\|MB\|GB\|TB) | | ✅ |
| SERVER_URL | URL, über welche das Backend erreicht werden kann. Die URL muss mit http:// oder https:// starten. Ein angegebener Pfad verursacht eine Fehlermeldung beim Start. | | ✴️ |
| CLIENT_URL | URL, über welche der Client erreicht werden kann. Die URL muss mit http:// oder https:// starten. Ein angegebener Pfad verursacht eine Fehlermeldung beim Start. | | ✴️ |
| USE_SECURITY_STRICT_LIMIT | Soll ein Anfrage-Limit für Login, Reset und Co gesetzt werden? | true | ✅ |

| Variable | Zweck | Default | Optional |
|---|---|---|---|
| | In diesem Fall ist der Nutzer nicht angemeldet, sondern versucht es. | | |
| SECURITY_STRICT_LIMIT_WINDOW | Über welches Zeitfenster soll das Limit angewandt werden?<br>Format: [0-9]*(y\|d\|h\|m\|s) | 15m | ✅ |
| SECURITY_STRICT_LIMIT_REQUEST_COUNT | Wie viele fehlerhafte Anfragen müssen gesendet werden, bis das Limit aktiviert ist? | 15 | ✅ |
| USE_SECURITY_LIMIT | Soll ein Anfrage-Limit für Anfragen innerhalb der App gesetzt werden?<br>In diesem Fall ist der Nutzer angemeldet. | true | ✅ |
| SECURITY_LIMIT_WINDOW | Über welches Zeitfenster soll das Limit angewandt werden?<br>Format: [0-9]*(y\|d\|h\|m\|s) | 1m | ✅ |
| SECURITY_LIMIT_REQUEST_COUNT | Wie viele fehlerhafte Anfragen müssen gesendet werden, bis das Limit aktiviert ist? | 500 | ✅ |
| TRUST_PROXY | Wird der Server hinter einem Proxy betrieben und Rate-Limit verwendet?<br>Ist dieser Wert nicht gesetzt, wird davon ausgegangen, dass kein Proxy verwendet wird.<br><br>Folgende Werte können gesetzt werden:<br>true / false<br>Anzahl der Proxies: [0-9]*<br>IP-Adresse des Proxy: ip<br>IP-Adressen der Proxy: ip1,ip2,... | | ✅ |
| VAPID_PUBLIC_KEY | Base64url-kodierter öffentlicher VAPID-Schlüssel für Web-Push-Benachrichtigungen. | | ✅ |

| Variable | Zweck | Default | Optional |
|---|---|---|---|
| VAPID_PRIVATE_KEY | Base64url-kodierter privater VAPID-Schlüssel (Gegenstück zu VAPID_PUBLIC_KEY). | | ✅ |
| LOG_LEVEL | Detailgrad der Logs: error, warn, info oder debug. Jede Stufe enthält die vorherigen. SQL-Abfragen werden nur bei debug protokolliert. | info | ✅ |
| LOG_MAX_FILES | Aufbewahrung der Log-Dateien als Anzahl an Dateien oder Tagen. Format: [0-9]+(d) | 14d | ✅ |
| LOG_MAX_SIZE | Maximale Größe einer Log-Datei, bevor eine neue Datei begonnen wird. Format: [0-9]+(k\|m\|g) | 20m | ✅ |
| LOG_SLOW_QUERY_MS | Datenbankabfragen, die länger als dieser Wert in Millisekunden dauern, werden als Warnung protokolliert. 0 deaktiviert die Prüfung. | 1000 | ✅ |
| FILE_ORPHAN_REPORT_ONLY | Gespeicherte Dateien ohne Bezug zu einem Datenbank-Eintrag werden bei true nur im Log gemeldet, bei false täglich gelöscht (siehe @file-storage). | true | ✅ |
| VAPID_SUBJECT | Kontakt-URI für Push-Dienste (Format: mailto:admin@example.com). Wird benötigt, wenn VAPID-Schlüssel gesetzt sind. | | ✅ |
| FILE_KEY_VERSION | Aktive Version der Dateischlüssel (siehe @file-storage). | 1 | ✅ |
| BACKUP_UPLOAD_MAX_SIZE | Maximale Größe hochgeladener Backups und Exporte in Megabyte. | 200 | ✅ |
| API_DOCS | Mit `public` liefert der Server die OpenAPI-Spezifikation und die API- | off | ✅ |

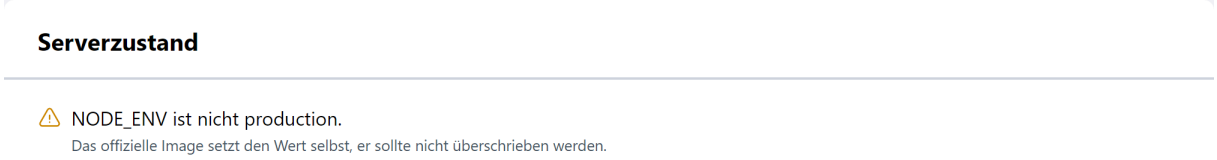
| Variable | Zweck | Default | Optional |
|---|---|---|---|
| | Dokumentation aus (siehe Abschnitt 2.10). Mit `off` sind beide nicht erreichbar. | | |
| | | | |
| ⬇️ Database-Variablen | | | |
| POSTGRES_DB | Name der Datenbank, die bei Erstellung direkt angelegt wird. | | ✴️ |
| POSTGRES_USER | Benutzername des Users, der bei Erstellung direkt angelegt wird. | | ✴️ |
| POSTGRES_PASSWORD | Passwort zum User, das bei Erstellung gesetzt wird. | | ✴️ |

✴️: Ein Fehlen dieser Variable verhindert das Starten der Anwendung!

**Hinweis:** Eine fehlerhafte Konfiguration der optionalen oder geforderten Variable verhindert das Starten der Anwendung.
**Hinweis:** Ist eine Datenbank leer, wird das neueste prüfbare Backup automatisch wiederhergestellt, sollte ein solches existieren.
**Hinweis:** APPLICATION_SECRET und APPLICATION_SALT werden zur Verschlüsselung von Zugangsdaten in der Datenbank und von Dateien (Abschnitt 2.11) verwendet. Beide Werte dürfen nur über den Rotationsmodus gewechselt werden (Abschnitt 2.7). Werden sie einfach geändert, sind Zugangsdaten und verschlüsselte Dateien nicht mehr lesbar, bis die bisherigen Werte wieder eingetragen sind. Bekannte Platzhalter wie `<application_secret>` oder `mysecret` verhindern das Starten der Anwendung.
Daten können entsprechend über ein Backup wiederhergestellt werden, jedoch werden dabei Zugangsdaten (Credentials) nicht überschrieben oder entfernt. Diese können nur über die Benutzeroberfläche geändert werden, da spezielle Zugänge wie der E-Mail-Versand bei Änderungen validiert werden müssen. Nutzer-Passwörter werden nach einem Backup-Restore nicht zurückgesetzt, um Fremdzugriffe zu verhindern.
In einem Fall, bei dem Zugangsdaten nicht mehr stimmen und der Mailversand nicht klappt, muss die Anwendung neu aufgesetzt werden.
**Hinweis:** Die SERVER_URL wird benötigt, damit die Backend-Anwendung Links zu sich selbst korrekt setzen kann. Sie schränkt nicht ein, über welche Domains das Backend erreichbar ist.
**Hinweis:** Die CLIENT_URL wird benötigt, damit die Backend-Anwendung Informationen zur Version des Clients auslesen kann. Zusätzlich ist die CLIENT_URL die einzige erlaubte CORS-Origin für die REST-API und die WebSocket-Verbindung: Der Wert muss exakt der Origin entsprechen, unter welcher die App im Browser aufgerufen wird (Protokoll, Host und ggf. Port), da sonst Anfragen des

Browsers an das Backend blockiert werden. Server-zu-Server-Aufrufe, z.B. über die API-Clients (Abschnitt 4.16.4), sind von dieser Einschränkung nicht betroffen.
**Hinweis:** Sind keine VAPID-Schlüssel (`VAPID_PUBLIC_KEY`, `VAPID_PRIVATE_KEY`) konfiguriert, generiert der Server bei jedem Start ein neues Schlüsselpaar und gibt dieses im Serverlog aus. Diese generierten Schlüssel können aus dem Log kopiert und als Umgebungsvariablen gesetzt werden. Ohne persistente Schlüssel werden bestehende Push-Subscriptions bei jedem Neustart ungültig, da sich der öffentliche Schlüssel ändert.

## 2.6 Serverzustand




Unter Verwaltung → Serverzustand erklärt die Anwendung Administratoren und dem Besitzer Probleme der Konfiguration, z.B. zu schwache Secrets, `TRUST_PROXY=true`, fehlende VAPID-Schlüssel, einen nicht erreichbaren Mailserver oder WebAPI-Zugänge mit Admin-Rechten ohne Ablaufdatum. Bei Fehlern erscheint zusätzlich ein Hinweis im Admin-Bereich. Auch die Berichte des Rotationsmodus werden hier angezeigt und quittiert.

## 2.7 Secret-Rotation

Der Rotationsmodus wechselt APPLICATION_SECRET und/oder APPLICATION_SALT und schlüsselt alle davon abhängigen Daten um, z.B. nach einem Verdacht auf Kompromittierung oder um einen schwachen Wert zu ersetzen.

1. Backup der Datenbank und des Ordners `/app/files` anlegen. Die Anwendung nicht im selben Schritt aktualisieren: Update und Rotation müssen getrennt erfolgen, die Anwendung sollte vorher einmal normal mit der neuen Version gestartet sein.
2. Neue Werte eintragen und die Rotation einschalten. Für Werte, die gleich bleiben, entfällt die jeweilige `_OLD`-Variable:

```yaml
- SECRET_ROTATION=true
- APPLICATION_SECRET=<neuer Wert, mindestens 32 zufällige Zeichen>
- APPLICATION_SALT=<neuer Wert, mindestens 32 zufällige Zeichen>
- APPLICATION_SECRET_OLD=<bisheriger Wert>
- APPLICATION_SALT_OLD=<bisheriger Wert>
```

3. Container neu erstellen. Die Anwendung legt einen Restore-Punkt an, schlüsselt die Datenbank in einer Transaktion und danach die Dateien um. Der Fortschritt steht im Log. Die API

bleibt währenddessen und danach gesperrt, der Prozess läuft aber weiter, damit `restart: unless-stopped` ihn nicht erneut startet.

4. Erscheint im Log `secret rotation completed`: `SECRET_ROTATION`, `APPLICATION_SECRET_OLD` und `APPLICATION_SALT_OLD` entfernen und den Container erneut erstellen.
5. Die im Serverzustand (Abschnitt 2.6) gemeldeten Schritte ausführen und den Bericht quittieren.

Stimmen die alten Werte nicht, bricht die Rotation ohne Änderung ab. Schlägt die Umschlüsselung einzelner Dateien fehl, setzt ein Neustart mit denselben Variablen sie fort. Ein wiederholter Start nach einer abgeschlossenen Rotation ändert nichts.
Der Bericht liegt als Datei unter `/app/files/secret-rotation` und enthält keine geheimen Werte.

| Gewechselt | Daten umgeschlüsselt | Nutzer abgemeldet | WebAPI-Tokens ungültig |
|---|---|---|---|
| nur Salt | ja | nein | nein |
| Secret oder beides | ja | ja | ja, außer `JWT_SECRET` ist gesetzt |

**Rotation zurücknehmen:** `SECRET_ROTATION=rollback` stellt Datenbank, Dateien und Backups im aktuellen Format auf die bisherigen Schlüssel zurück, Änderungen seit der Rotation bleiben erhalten. Der Ablauf entspricht der Rotation, nur sind die Werte vertauscht: `APPLICATION_SECRET` und `APPLICATION_SALT` enthalten die bisherigen Werte, `APPLICATION_SECRET_OLD` und `APPLICATION_SALT_OLD` die der Rotation. Soll auch der Inhalt zurück, wird danach der Restore-Punkt der Rotation eingespielt. Der Restore-Punkt allein reicht nicht, da er weder Dateien noch gespeicherte Zugangsdaten der Nutzer zurückstellt.
**Achtung:** Backups im aktuellen Format stellt die Rotation auf die neuen Schlüssel um, ausgenommen ihren eigenen Restore-Punkt. Backups im alten Format werden bei einer leeren Datenbank nicht mehr automatisch wiederhergestellt, ihre verschlüsselten Werte sind nur lesbar, solange die `_OLD`-Variablen gesetzt sind. Exporte bleiben lesbar, da sie nur durch ihr eigenes Passwort geschützt sind.
Ein Wechsel von `JWT_SECRET` allein benötigt keinen Rotationsmodus.

## 2.8 Logging

Der Server schreibt alle Meldungen gleichzeitig auf die Konsole und in Log-Dateien im Ordner `logs` (im Docker-Container `/app/logs`). Pro Tag wird eine Datei `app-JJJJ-MM-TT.log` angelegt; wird die mit `LOG_MAX_SIZE` festgelegte Größe überschritten, wird eine weitere Datei begonnen. Ältere Dateien werden entsprechend `LOG_MAX_FILES` automatisch gelöscht. Erfasst werden Meldungen der Anwendung, aller API-Anfragen und der Datenbank.
Stürzt der Server durch einen unerwarteten Fehler ab, wird dieser zusätzlich sofort in die Datei `crash.log` geschrieben. Bei einem Abbruch von außen (z.B. durch fehlenden Arbeitsspeicher)

können die letzten Einträge fehlen. Wird der Server ohne Docker betrieben, empfiehlt sich deshalb ein Prozessmanager (z.B. systemd, pm2 oder unter Windows NSSM), welcher die Konsolenausgabe zusätzlich speichert und den Server nach einem Absturz neu startet.
Die Logs können zusätzlich in der Anwendung unter `Verwaltung > Logs` eingesehen, gefiltert und heruntergeladen werden. Dort lässt sich der Detailgrad auch vorübergehend bis zum nächsten Neustart ändern (siehe Abschnitt 4.18).
**Fehler-ID:** Jede Anfrage erhält eine kurze ID, die im Log bei der Anfrage (`id:`) und beim zugehörigen Fehler steht. Bei einem Serverfehler zeigt die App diese ID in der Fehlermeldung an, z.B. „(Fehler-ID: Xk3a_9Qe)“. Mit der Suche unter `Verwaltung > Logs` lässt sich der Fehler so direkt finden. Die ID wird auch im Header `X-Request-Id` jeder Antwort mitgeschickt.

## 2.9 Health-Checks

Die Docker-Images von Server und App enthalten einen Health-Check, den Docker und Plattformen wie Coolify oder Portainer automatisch auswerten. Eine zusätzliche Konfiguration in der compose-Datei ist nicht nötig.

- **Server:** `GET /api/health` antwortet, solange der Server läuft, auch während eine Datensicherung wiederhergestellt wird.
- **Server:** `GET /api/ready` prüft zusätzlich die Datenbankverbindung und antwortet mit Status `503`, solange die Datenbank nicht erreichbar oder die API vorübergehend deaktiviert ist. Dieser Endpunkt eignet sich für externes Monitoring (z.B. Uptime Kuma).
- **App:** `GET /health` wird direkt von nginx beantwortet.

Die Endpunkte benötigen keine Anmeldung und erscheinen nicht im Log.

## 2.10 API-Dokumentation (OpenAPI)

Der Server beschreibt seine gesamte REST-API in einer OpenAPI-Spezifikation (Version 3.1). Sie wird bei jeder Version aus dem Code erzeugt und enthält alle Routen, Parameter, Anfrage- und Antwortformate, Fehlercodes sowie die benötigten Berechtigungen. Die Routen des Core sind darin enthalten.
Standardmäßig ist die Spezifikation nicht erreichbar. Mit `API_DOCS=public` liefert der Server aus:

- `GET /api/openapi.json`: die Spezifikation, `info.version` entspricht der Version des Servers.
- `GET /api/docs`: eine lesbare Dokumentation der API (Scalar). Alle Dateien dafür kommen vom eigenen Server, es werden keine externen Dienste geladen.

Beide Adressen sind ohne Anmeldung abrufbar. Daten der Anwendung enthalten sie nicht, sie verraten aber Aufbau und Version der API. Wer die Dokumentation nur für die Entwicklung eigener

Anbindungen braucht, kann sie dafür vorübergehend aktivieren oder die Spezifikation aus dem Repository des Servers (`openapi.json`) bzw. aus den API-Clients (Abschnitt 4.16.4) verwenden.
**Fehlerformat:** Fehler beantwortet der Server standardmäßig mit einer Meldung als Text. Sendet ein Client den Header `Accept: application/problem+json`, erhält er ein Objekt nach RFC 9457:

```json
{
  "type": "about:blank",
  "title": "Forbidden",
  "status": 403,
  "detail": "missing permission for management.user.update",
  "code": "permission.missing",
  "params": { "permission": "management.user.update" },
  "requestId": "Xk3a_9Qe"
}
```

`code` ist stabil und eignet sich zur Auswertung in eigenen Anwendungen, `detail` ist ein Text für Entwickler. Validierungsfehler haben den Code `validation.failed` und listen unter `errors` je Feld den Pfad (z.B. `body.firstname`) und eine deutsche Meldung. `requestId` ist die Fehler-ID aus dem Log (Abschnitt 2.8).

## 2.11 Dateiablage

Alle dauerhaft gespeicherten Dateien wie Bilder, Dokumente, Ausdrucke und Backups liegen im Ordner `files` (im Docker-Container `/app/files`). Hochgeladene Dateien werden zunächst in einen temporären Ordner des Betriebssystems geschrieben und erst nach erfolgreicher Prüfung in den Ordner `files` verschoben. Übrig gebliebene temporäre Dateien werden täglich entfernt. Ein zusätzliches Volume für den temporären Ordner ist nicht nötig.
**Verschlüsselte Dateien:** Dateien mit personenbezogenen oder Gesundheitsdaten werden verschlüsselt (AES-256-GCM) im Ordner `files` abgelegt. In FF Admin betrifft das die Dokumente der Mitglieder (Aufnahmeantrag, Ehrungen, Ausbildungen), die Nachweis-Dateien und die Bilder der Führerscheinkontrolle.. Der Schlüssel wird aus `APPLICATION_SECRET` und `APPLICATION_SALT` abgeleitet. Eine Sicherung des Ordners `files` ist daher nur zusammen mit diesen Werten nutzbar.
Wird `FILE_KEY_VERSION` erhöht, stellt der tägliche Lauf die Dateischlüssel auf die neue Version um, ohne dass sich Secret oder Salt ändern. Alle Versionen leiten sich aus denselben Werten ab, bei einem Verdacht auf Kompromittierung hilft deshalb nur der Rotationsmodus (Abschnitt 2.7).
Durch abgebrochene Anfragen oder gelöschte Datenbank-Einträge können Dateien zurückbleiben, auf die kein Eintrag mehr verweist. Der Server sucht täglich um Mitternacht nach solchen Dateien, sofern sie älter als 24 Stunden sind. Standardmäßig werden die gefundenen Dateien nur im Log gemeldet (`FILE_ORPHAN_REPORT_ONLY=true`), damit vor dem ersten Löschen geprüft werden kann, welche Dateien betroffen wären. Mit `FILE_ORPHAN_REPORT_ONLY=false` werden diese Dateien anschließend automatisch gelöscht, leere Ordner werden ebenfalls entfernt.

## 2.12 Update der Version

Um eine Version auf eine Neuere zu aktualisieren, muss meist nur der Docker-Tag oder das Repo ersetzt werden.
Wer Docker mit `vX`, `vX.Y` oder `latest` nutzt, kann das neue Image direkt mit `docker pull` neu beziehen und dann den Container neustarten.

Informationen zu neuen Versionen können innerhalb der App unter `Verwaltung > Version` oder in den Release-Pages gefunden werden.
Die Releases beinhalten Informationen zu einem Update und was zu beachten ist. So enthalten die Release-Informationen beispielsweise Vorbereitungen vor einem Update.

Bei Verwendung mittels Git, müssen die Repos neu bezogen werden. Anschließend müssen die Dependencies neu installiert und die Anwendungen neu gebaut werden. Bei einem ausgecheckten Branch (`main` oder `release/X.Y`) genügt dazu `git pull`, bei einem Tag muss der neue Tag ausgecheckt werden.

### 2.12.1 Support-Linien

Jede Minor-Version (z.B. `1.6`) bildet eine eigene Support-Linie. Neben der aktuellen Version werden ältere Linien weiter gepflegt, sodass ein Update auf eine neue Minor-Version nicht sofort erfolgen muss.
Unterstützt werden die letzten drei Minor-Versionen der aktuellen Hauptversion sowie jeweils die neueste Minor-Version jeder älteren Hauptversion. Ist z.B. `1.7` aktuell, werden `1.7`, `1.6` und `1.5` unterstützt.
Unterstützte ältere Linien erhalten Sicherheits- und Fehlerbehebungen sowie Aktualisierungen von Abhängigkeiten und Basis-Images als Patch-Version (z.B. `1.6.5`), jedoch keine neuen Funktionen. Diese Updates erscheinen automatisch unter dem Docker-Tag `vX.Y` der Linie und im Git im Branch `release/X.Y`.
Erscheint eine neue Minor-Version, endet der Support der ältesten Linie. Sie erhält danach keine Updates mehr. Das Image der letzten Patch-Version bleibt erhalten, ältere Patch-Versionen einer beendeten Linie werden aus der Registry entfernt. Instanzen auf einer beendeten Linie sollten zeitnah auf eine unterstützte Linie aktualisiert werden.
Welche Linien aktuell unterstützt werden, ist für alle Anwendungen zentral unter `support.json` einsehbar (Abschnitt `supported`, beendete Linien unter `eol`).

## 2.13 WebApp

FF Admin ist als WebApp verfügbar. Dadurch lässt sich die Anwendung auf einem Smartphone oder Desktop über den Browser installieren.
Sollte die Version des FF Admin aktualisiert werden, wird dem Nutzer in der Anwendung eine Information angezeigt, dass die Anwendung aktualisiert werden kann. Wird im Client eine neue Version erkannt, wird auch eine Meldung zum manuellen Download der neuen Version angezeigt.



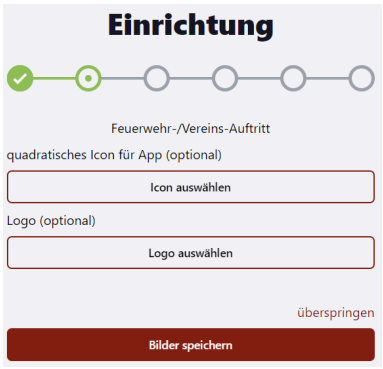

Updates werden im Hintergrund heruntergeladen, aber erst auf Klick installiert, da die Seite dabei neu geladen wird und ungespeicherte Eingaben sonst verloren gingen. Auf neue Versionen prüft die App beim Start, beim Zurückkehren in die App, nach einer Unterbrechung der Verbindung und alle 30 Minuten. Wurde das Update bereits in einem anderen Fenster installiert, weist die App darauf hin, offene Eingaben zu speichern und die Seite neu zu laden.
Kann die App eigene Dateien nicht laden oder schlägt ein Update fehl, bietet der Hinweis `App reparieren` an. Dabei werden die zwischengespeicherten App-Dateien gelöscht und neu geladen. Anmeldung, Push-Benachrichtigungen und lokal gespeicherte Daten bleiben erhalten. Ein manuelles Leeren des Browser-Caches ist dafür nicht mehr nötig.

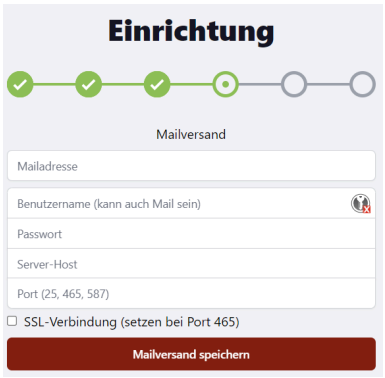
## 2.14 Einrichtung

Um die Anwendung nutzen zu können, kann ein erster Administrator-Account wie folgt erstellt werden:




**Admin Benutzer erstellen**: Um einen ersten Benutzer mit Administrator-Berechtigungen zu erstellen, muss der Einrichtungs-Assistent unter dem Pfad `/setup` aufgerufen werden. Nach der initialen Einrichtung wird der Pfad automatisch geblockt. Der Nutzer, welcher über die Einrichtung erstellt wird, wird anschließend als `Owner` gekennzeichnet und hat unabhängig der gesetzten Berechtigungen Vollzugriff auf die gesamte Anwendung.
Der Einrichtungs-Assistent ermöglicht das Setzen erster Einstellungen wie Vereinsdaten und Logos. Auch ermöglicht der Assistent das Einrichten der Mail-Adresse, die von FF Admin verwendet werden soll. Über diese Mailadresse können andere Nutzer eingeladen, Zugänge zurückgesetzt oder Newsletter versendet werden.







**Rollen und Berechtigungen**: Unter `Verwaltung > Rollen` können die Rollen und Berechtigungen für die Benutzer erstellt und angepasst werden.

**Nutzer einladen**: Unter `Verwaltung > Benutzer` können weitere Nutzer eingeladen werden. Diese erhalten dann eine E-Mail mit einem Link. Über diesen Link können die Nutzer einen Account mit Zugangsdaten erstellen. Für den Zugang können entweder TOTP oder Passwörter verwendet werden.

# 3 Konzepte

FF Admin basiert auf mehreren Konzepten, die das System modular und flexibel machen, so dass es an verschiedene Anwendungsfälle angepasst werden kann.

## 3.1 Stammdaten

Stammdaten sind grundlegende Basisdaten, die als Grundlage für weitere Einträge dienen. Dazu gehören z.B. Anrede, Mitgliedsstatus oder verschiedene Arten von Kalendereinträgen.

Diese Daten sind frei definierbar, so dass die Benutzer ihre eigene Namensgebung festlegen und sicherstellen können, dass alle benötigten Einträge zur Verfügung stehen.

Damit Einträge wie Mitglieder oder Termine angelegt oder verwendet werden können, müssen **zuvor** die entsprechenden Stammdaten angelegt worden sein.

## 3.2 Berechtigungen

Das Berechtigungssystem ist tief in FF Admin integriert und steuert den Zugriff von Benutzern oder API-Clients auf verschiedene Sektionen und Module. Berechtigungen bestimmen, ob jemand Daten lesen, erstellen, aktualisieren oder löschen kann.

Berechtigungen werden immer summiert:
Ein Benutzer erhält alle Berechtigungen, die ihm direkt oder indirekt über Rollen zugewiesen wurden.
Ein Beispiel: Erhält ein Nutzer Leserechte zu den Mitgliedern über eine Rolle aber auch Rechte zum Erstellen direkt zugewiesen, so kann der Benutzer letztendlich Mitglieder erstellen. Die höhere Berechtigung zählt in dem Fall.

Der Berechtigungseditor ist in Sektionen und Module unterteilt. Wenn ein Benutzer eine Berechtigung für einen Abschnitt erhält, gilt diese automatisch auch für alle untergeordneten Module.

**Hinweis:** Um Einträge in einem Modul, das auf Stammdaten zugreift, anlegen oder bearbeiten zu können, erhält der Benutzer automatisch Leserechte auf die Stammdaten. Dies bedeutet jedoch nicht, dass der Benutzer die Stammdaten direkt in der Anwendung sehen kann - hierfür benötigt er eine explizite Berechtigung für den entsprechenden Abschnitt oder das entsprechende Modul.

## 3.3 Zeitformate

Es gibt verschiedene Formate für Zeitangaben. Das eine Format wird für Einstellungen wie der Session-Gültigkeitsdauer, das andere für wiederkehrende Termine und Erinnerungen verwendet.

### 3.3.1 Gültigkeitsdauer

Angaben im Format **<zahl>(mins|hrs|days|weeks|months|yrs|m|h|d|w|mo|y)** definieren sich durch eine Zahl gefolgt von einer Zeiteinheit und beschreiben Zeitlängen.
Die unterstützten Zeiteinheiten sind:

- **m** bzw. **mins**: Minuten - Beispiel: **15m** bzw. **15mins** für 15 Minuten gültig
- **h** bzw. **hrs**: Stunden - Beispiel: **2h** bzw. **2hrs** für 2 Stunden gültig
- **d** bzw. **days**: Tage - Beispiel: **1d** bzw. **1days** für 1 Tag gültig
- **w** bzw. **weeks**: Minuten - Beispiel: **15w** bzw. **15weeks** für 3 Wochen gültig
- **mo** bzw. **months**: Monate - Beispiel: **1mo** bzw. **1months** für 1 Monat gültig
- **y** bzw. **yrs**: Minuten - Beispiel: **15y** bzw. **15yrs** für 15 Jahre gültig

Dieses Format wird hauptsächlich für die Gültigkeitsdauer von Nutzersessions verwendet.

### 3.3.2 Zeitintervalle

Angaben im Format **<zahl>-(d|m|y)** oder **DD/MM** oder **DD/*** bzw. **DD/MM+** oder **DD/*+** beschreiben Zeitintervalle. Diese Angaben dienen der Berechnung eines künftigen oder vergangenen Datums, welches zum Erinnern oder als nächstes Datum eines Intervalls genutzt werden kann.

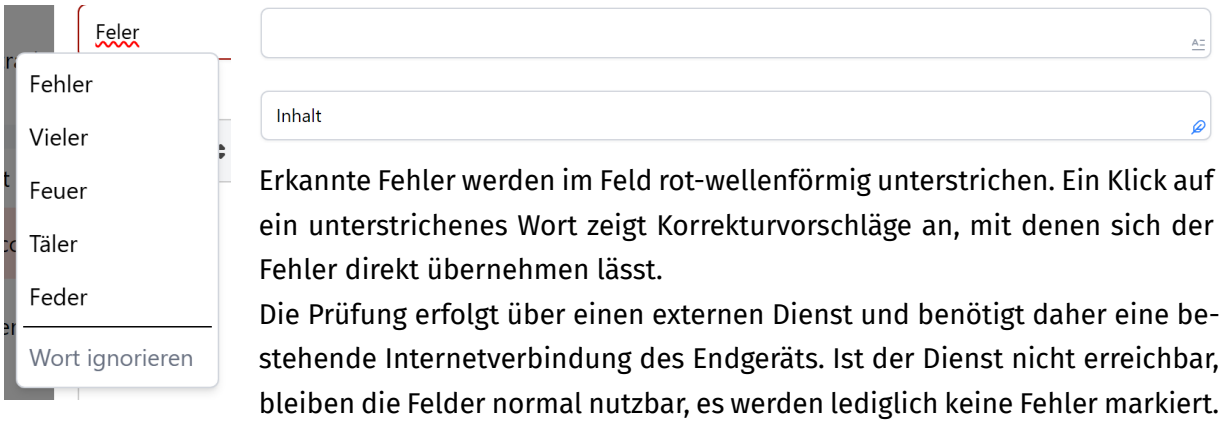
| **Format** | **als Intervall** | **als Erinnerung** |
|---|---|---|
| **<zahl>-(d\|m\|y)** | Intervall für alle x Tage, alle x Monate oder alle x Jahre.<br>z.B.: **7-d** für alle 7 Tage | Erinnerung für x Tage, x Monate oder x Jahre vor dem Fälligkeitsdatum. |
| **DD/MM** | Intervall für jedes Jahr an einem festen Tag.<br>z.B.: **15/06** für den 15. Juni | Erinnerung an einem bestimmten Tag vor dem Fälligkeitsdatum. |
| **DD/*** | Intervall für bestimmten Tag jeden Monat.<br>z.B.: **01/*** für den ersten Tag jeden Monat | Erinnerung an einem bestimmten Tag eines Monat vor dem Fälligkeitsdatum. |

| Format | als Intervall | als Erinnerung |
|---|---|---|
| **DD/MM+** oder **DD/*+** | Erzwingt einen Mindestabstand zum Ausgangsdatum: **15/06+** liegt garantiert nicht im selben Jahr, **01/*+** nicht im selben Monat. Mit Zahl, z.B. **15/06+2**, für einen größeren Abstand (hier mind. 2 Jahre). | Erzwingt einen Mindestabstand zum Fälligkeitsdatum: **15/06+** liegt garantiert mindestens ein Jahr, **01/*** mindestens einen Monat vorher — auch wenn das Datum sonst nur wenige Tage vorher läge. Mit Zahl, z.B. **15/06+2**, für einen größeren Abstand (hier mind. 2 Jahre). |

Dieses Format wird hauptsächlich für die Intervalle von Wartungen und Prüfungen und den Zeitpunkt der Erinnerung vor dem nächsten Check verwendet.

## 3.4 Rechtschreibprüfung

In vielen Textfeldern steht eine integrierte Rechtschreibprüfung zur Verfügung. Ein Symbol am Rand des Feldes zeigt an, ob die Prüfung für das Feld aktiv ist bzw. gerade läuft. Schreibgeschützte oder deaktivierte Felder werden nicht geprüft.




Erkannte Fehler werden im Feld rot-wellenförmig unterstrichen. Ein Klick auf ein unterstrichenes Wort zeigt Korrekturvorschläge an, mit denen sich der Fehler direkt übernehmen lässt.
Die Prüfung erfolgt über einen externen Dienst und benötigt daher eine bestehende Internetverbindung des Endgeräts. Ist der Dienst nicht erreichbar, bleiben die Felder normal nutzbar, es werden lediglich keine Fehler markiert.

## 3.5 Dateien und Bilder

Hochgeladene Dateien werden vom Server anhand ihres tatsächlichen Inhalts geprüft, nicht anhand der Dateiendung. Je nach Modul werden Bilder, PDF-Dokumente oder beides akzeptiert.

**Bilder** können als JPEG, PNG, WebP, GIF, TIFF oder AVIF hochgeladen werden. Sie werden einheitlich im WebP-Format gespeichert, entsprechend der Aufnahme-Ausrichtung gedreht und auf

maximal 2000 Pixel Kantenlänge verkleinert. Dadurch bleiben Speicherbedarf und Ladezeiten gering. HEIC-Bilder, wie sie z.B. von iPhones erstellt werden, werden nicht unterstützt und müssen vorher als JPEG oder PNG exportiert werden. Ausgenommen sind Logo und Icon in den Einstellungen, die weiterhin als PNG hochgeladen werden.

**PDF-Dokumente** werden unverändert gespeichert.

Dateien, die nicht öffentlich bereitgestellt werden, sind nur für angemeldete Nutzer mit entsprechender Berechtigung abrufbar. Bilder und PDFs werden direkt in der Anwendung angezeigt und können von dort heruntergeladen werden.

## 3.6 Engines

FF Admin stellt eine Reihe von systemweiten Funktionen zur Verfügung, die vom Benutzer konfiguriert und in verschiedenen Modulen verwendet werden können.

### 3.6.1 Notification-Engine

Die Notification-Engine ermöglicht es dem Benutzer, Benachrichtigungen für verschiedene Ereignisse in der Anwendung zu konfigurieren. Dies umfasst sowohl systemweite Vorgänge als auch Erinnerungen an Fristen und Fälligkeiten.
Benachrichtigungen können entweder direkt in der Anwendung angezeigt oder per E-Mail erhalten werden. Der Benutzer kann dabei selbst festlegen, welche Ereignisse auf welchem Weg mitgeteilt werden sollen.
Dabei kann ein Nutzer nur Benachrichtigungen für Module konfigurieren, für die er über entsprechende Berechtigungen verfügt. Werden einem Nutzer nachträglich Berechtigungen entzogen, prüft das System beim Versand einer Benachrichtigung automatisch, ob der Zugriff auf das betreffende Modul weiterhin besteht.

Eine detaillierte Anleitung zur Verwendung finden Sie unter **Benutzerbereich und Accounteigenschaften -> Benachrichtigungen** (Abschnitt 5.2).

### 3.6.2 Template-Engine

Die Template-Engine ermöglicht es dem Benutzer, eigene Formatvorlagen zu erstellen und Standardvorlagen damit zu ersetzen. Ein Modul kann Vorlagen für Kopf- und Fußzeilen sowie für den Hauptteil eines Dokuments verwenden.
Diese Vorlagen werden zum Drucken von PDFs oder zum Versenden von E-Mails verwendet. Sie werden mit Daten aus FF Admin gefüllt und anschließend exportiert oder versendet.

In den Vorlagen können Platzhalter eingefügt werden, die beim Druck mit den entsprechenden Daten ersetzt werden.

Eine detaillierte Anleitung zur Verwendung finden Sie unter **Module -> Templates & Template Builder** (Abschnitt 4.8).

### 3.6.3 Query-Engine

Die Query-Engine ermöglicht das Erstellen und Ausführen von individuellen Abfragen auf die Mitgliederdaten. Diese Abfragen können gespeichert und in anderen Modulen wieder verwendet werden.

Beispiele von Abfragen wären:
- Mitglieder eines bestimmten Mitgliedschaftstyps
- Mitglieder einer bestimmten Mitgliedschaftsdauer
- Mitglieder eines bestimmten Alters
- ...

Eine detaillierte Anleitung zur Benutzung finden Sie unter **Module -> Query Builder & Query Store** (Abschnitt 4.9).

**Hinweis:** Die Query-Engine ermöglicht nur das Lesen von Mitglieder- und mitgliederbezogenen Daten. Das Schreiben oder Verändern von Daten ist nicht möglich.

# 4 Module

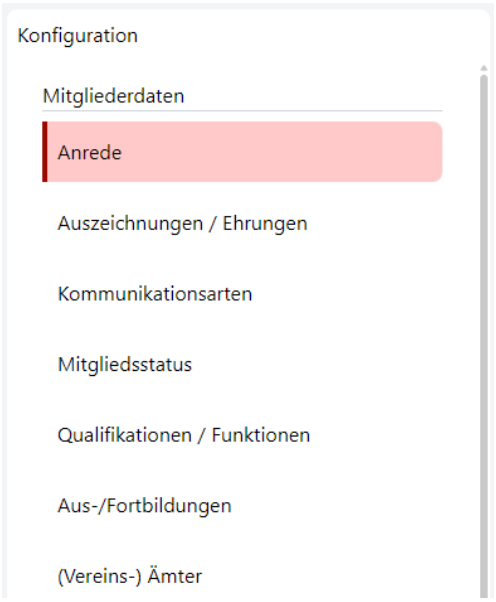
FF Admin nutzt verschiedenste Module, um Daten zu organisieren, die Verwaltung zu vereinfachen und Berechtigungen sinnvoll setzen zu können.
Diese Module ermöglichen die Anpassung der Anwendung an die eigenen Bedürfnisse, sodass die Daten der Feuerwehr oder des Vereins optimal abgebildet werden können.




Die Module gliedern sich in vier Teilbereiche, die sich an der Navigationsleiste orientieren:

- Zum einen die Verwaltung von personenbezogenen Vereinsinhalten wie Mitgliederdaten, Kalendereinträgen, Newslettern oder Protokollen.
- Ein weiterer Bereich umfasst die Organisation von Objekten der Feuerwehr oder des Vereins, etwa Gerätschaften, Fahrzeuge oder Kleidung.
- Die übrigen Teilbereiche dienen der Verwaltung von Basisdaten sowie der Konfiguration und Steuerung der Anwendung selbst.

## 4.1 Mitgliederverwaltung & Stammdaten

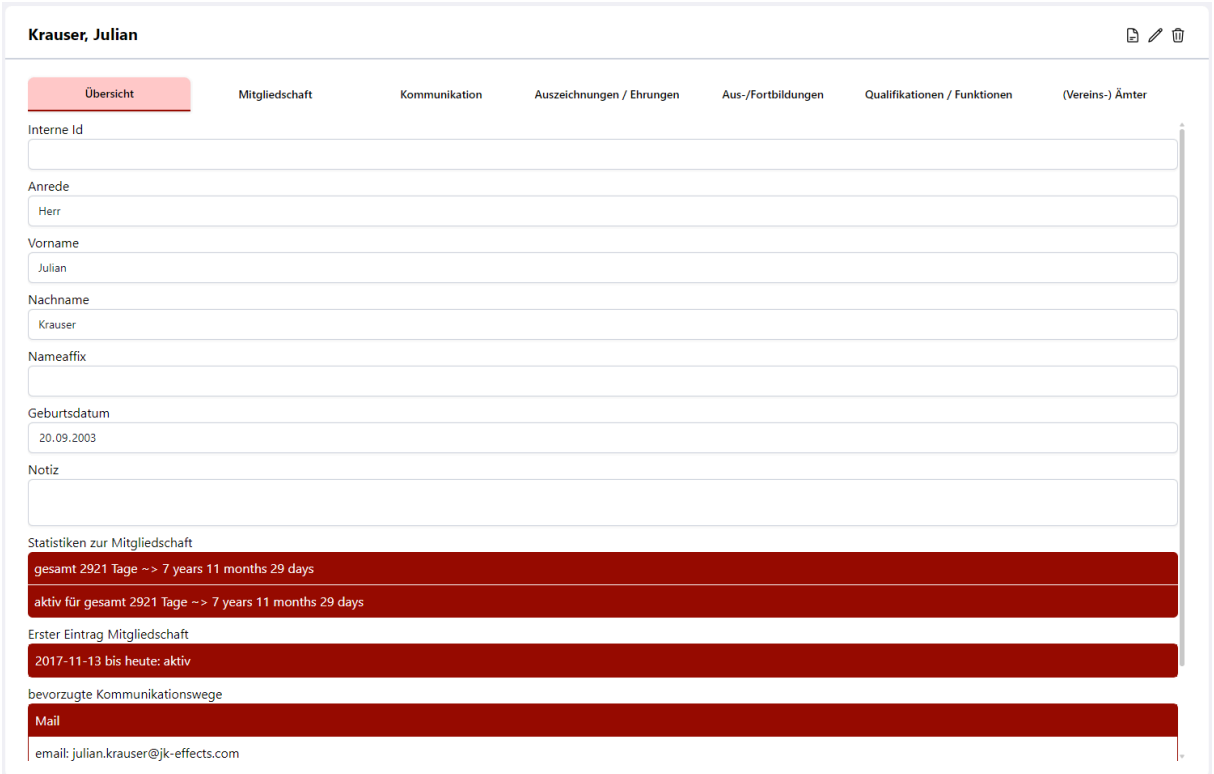



Die Mitgliederverwaltung ermöglicht die strukturierte Erfassung und Verwaltung von Mitgliederdaten. Bevor einzelne Mitgliederdatensätze angelegt werden können, müssen zunächst die Mitgliederstammdaten definiert werden. Diese Stammdaten dienen als Grundlage für die spätere Datenerfassung und ermöglichen eine einheitliche, standardisierte Verwaltung. Sind die Stammdaten einmal hinterlegt, können sie bei der Erfassung neuer Mitglieder verwendet werden.
Dazu gehören grundlegende Kategorien wie Anrede, Auszeichnungen/Ehrungen, Kommunikationskanäle, Mitgliedsstatus, Qualifikationen/Funktionen, Aus-/Fortbildungen und Vereinsämter.

Alle Stammdaten können frei ausgefüllt werden und einem Mitglied je nach Typ mehrfach zugewiesen werden. Dabei können diese unterschiedliche Inhalte speichern. Der Titel oder die

Bezeichnung eines Stammdaten-Typs kann dabei nicht doppelt vorkommen.

- **Anrede:** Anreden können je nach Bedarf mit Werten wie Herr, Frau, Dr. oder Prof. erstellt werden.
- **Auszeichnungen/Ehrungen:** Auszeichnungen und Ehrungen können mit individuellen Bezeichnung angelegt werden.
- **Kommunikationskanäle:** Kommunikationskanäle enthalten sowohl die Bezeichnung als auch die dem Kanal zugeordneten Kommunikationsarten. Folgende Felder können zugeordnet werden: Telefonnummer, E-Mail-Adresse, Postleitzahl, Stadt/Ort, Straße, Hausnummer und Hausnummernzusatz.
- **Mitgliedsstatus:** Mitgliedsstatus sind frei wählbar und können die Zugehörigkeit zu einem Verein oder einer Vereinsgruppe widerspiegeln. Beispiele wären „aktiv“, „passiv“, „fördernd“ oder „Tischtennis“ oder „Fußball“.
- **Qualifikationen/Funktionen:** Qualifikationen/Funktionen können frei vergeben werden. Zusätzlich kann eine Beschreibung angegeben werden. Beispiele sind „Atemschutzgeräteträger“, „Gruppenführer“...
- **Aus-/Fortbildungen:** Aus-/Fortbildungen können frei vergeben werden. Zusätzlich kann eine Beschreibung angegeben werden. Beispiele sind Lehrgänge zu den Qualifikationen/Funktionen.
- **(Vereins-) Ämter:** Vereinsämter können ebenfalls frei belegt werden. Dabei können alle Funktionen im Verein von „Vorstand“ bis „EDV“ angelegt werden.



Innerhalb der Mitgliederansicht können dann individuelle Mitgliederprofile angelegt und gepflegt werden. Diese enthalten persönliche Daten wie Name, Geburtsdatum oder eine Notiz sowie spezifische Informationen zur Mitgliedschaft wie Eintrittsdatum, Qualifikationen/Funktionen oder verliehene Auszeichnungen. Darüber hinaus bietet das System eine automatische Berechnung der Mitgliedschaftsdauer zur Gesamtdauer und je zugewiesenem Status, um die aktive Zeit eines Mitglieds jederzeit nachvollziehen zu können. Daten zu einem Mitglied können als pdf exportiert werden. Das Basis-Template beinhaltet alle Mitgliederdaten außer den Kommunikationskanälen. Das Dokument-Icon in der Kopfzeile in der Mitglied-Detailansicht löst den Druck aus. Das erstellte pdf wird nicht auf dem Server gespeichert und muss deshalb direkt heruntergeladen werden, sollte dieses für später gespeichert werden.

Die Einträge zu einem Mitglied umfassen dann jeweils den Basisdatensatz und weitere Details, wie z.B. einen Zeitraum, in dem dieser Eintrag gültig ist oder war. Einige Eingabemöglichkeiten weisen Besonderheiten auf.

**Mitgliedschaften** werden als Zeiträume eines bestimmten Status angegeben. Durch die Angabe eines neuen Zeitraums wird ein bisher aktiver Zeitraum beendet und das Enddatum auf den Tag vor Beginn des neuen Zeitraums gesetzt. Derzeit wird nicht geprüft, ob sich Zeiträume überschneiden. Zusätzlich kann bei einem Eintrag ein Grund für das Ende des Eintrags angegeben werden. Beispiele hierfür wären der Austritt oder der Tod eines Mitglieds.

Bei einem **Kommunikationskanal** können nur die Felder ausgefüllt werden, die in den Stammdaten hinterlegt wurden. Zusätzlich kann angegeben werden, welche der Adressen bevorzugt verwendet werden soll. Dadurch ist es möglich, mehrere Telefonnummern oder Adressen zu erfassen, ohne die Übersicht zu verlieren. Weiterhin kann angegeben werden, ob ein Kommunikationskanal für den Versand eines Newsletters verwendet werden soll. Hierbei ist zu beachten, dass ein Newsletter derzeit nicht an eine Telefonnummer versendet werden kann.
Im Feuerwehrkontext kann bei einer Telefonnummer angegeben werden, ob diese für eine SMS-Alarmierung verwendet werden kann. Der Versand einer SMS an Mitglieder im Alarmfall ist nicht in FF Admin integriert und muss extern konfiguriert werden. Die Angabe erleichtert jedoch die Identifikation von Telefonnummern, die für diesen Anwendungsfall verwendet werden können.

**Auszeichnungen/Ehrungen** können einem Mitglied mit einem Übergabedatum zugeordnet werden. Dabei kann auch vermerkt werden, ob die Annahme vom Mitglied verweigert oder vom Verein verweigert wurde. Optional kann eine Notiz hinzugefügt werden. Die Notiz kann die Gründe für die Nichtannahme oder eine verspätete Übergabe enthalten.

**Qualifikationen/Funktionen** können auch mit einem Zeitraum angelegt werden. Ein Eintrag kann

weiterhin eine Notiz enthalten, z.B. eine Einschränkung der Qualifikationen/Funktionen. Außerdem kann angegeben werden, warum die Qualifikationen/Funktionen zurückgezogen wurde oder nicht mehr gültig ist.

**Aus-/Fortbildungen** können auch mit einem Zeitraum angelegt werden. Ein Eintrag kann weiterhin einen Ausbildungsort oder eine Notiz enthalten, z.B. eine Einschränkung der Aus-/Fortbildungen.

**(Vereins-) Ämter** können ebenfalls mit den Zeiträumen der Ausübung angelegt werden. Auch hier besteht die Möglichkeit, eine Notiz hinzuzufügen.

### 4.1.1 Nachweisdokumente

Zu einem Mitglied sowie zu jeder Auszeichnung/Ehrung und jeder Aus-/Fortbildung kann jeweils ein Nachweisdokument hinterlegt werden:

- **Aufnahmeantrag** beim Anlegen oder Bearbeiten eines Mitglieds,
- **Urkunde** bei einer Auszeichnung/Ehrung,
- **Nachweis** (z.B. Lehrgangszeugnis) bei einer Aus-/Fortbildung.

Erlaubt sind pdf-Dateien und Bilder bis 20 MB. Bilder werden beim Speichern automatisch in ein platzsparendes Format umgewandelt. Pro Eintrag ist genau ein Dokument möglich. Ein hinterlegtes Dokument kann beim Bearbeiten über *Datei ersetzen* ausgetauscht oder über das Papierkorb-Symbol entfernt werden. Die Auswahl wird erst beim Speichern des Formulars übernommen und kann bis dahin rückgängig gemacht werden.

Ist ein Dokument hinterlegt, erscheint in der Mitglied-Übersicht bzw. am jeweiligen Listeneintrag ein Dokument-Symbol. Darüber wird das Dokument in einer Vorschau geöffnet und kann von dort auch heruntergeladen werden. Die Dateien werden im Ordner `files` des Servers je Mitglied abgelegt und beim Löschen des Eintrags entfernt.

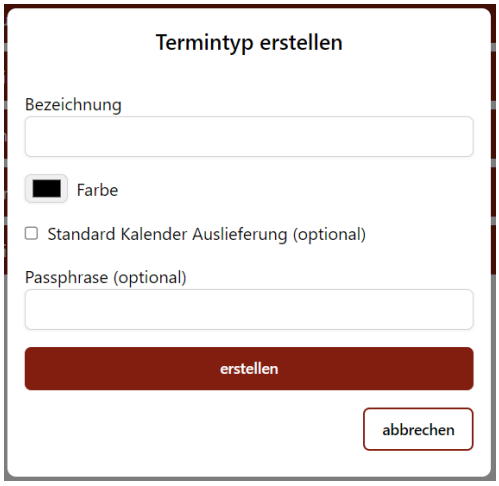
Über die Moduleinstellungen (Abschnitt 4.20) lässt sich festlegen, dass Aufnahmeantrag, Urkunden und Aus-/Fortbildungsnachweise an den pdf-Ausdruck eines Mitglieds angehängt werden. Der Ausdruck enthält dann nach den Mitgliederdaten eine Seite *Anhang* mit einer Übersicht aller angehängten Dokumente, gefolgt von den Dokumenten selbst: pdf-Dateien mit allen Seiten, Bilder jeweils auf einer eigenen Seite mit Überschrift. Urkunden und Nachweise werden chronologisch sortiert. Kann ein Dokument nicht eingebunden werden, wird an seiner Stelle eine Hinweisseite eingefügt.

**Anmerkung:** In jeder erfassbaren Datenkategorie können Einträge mehrfach vorkommen. Eine

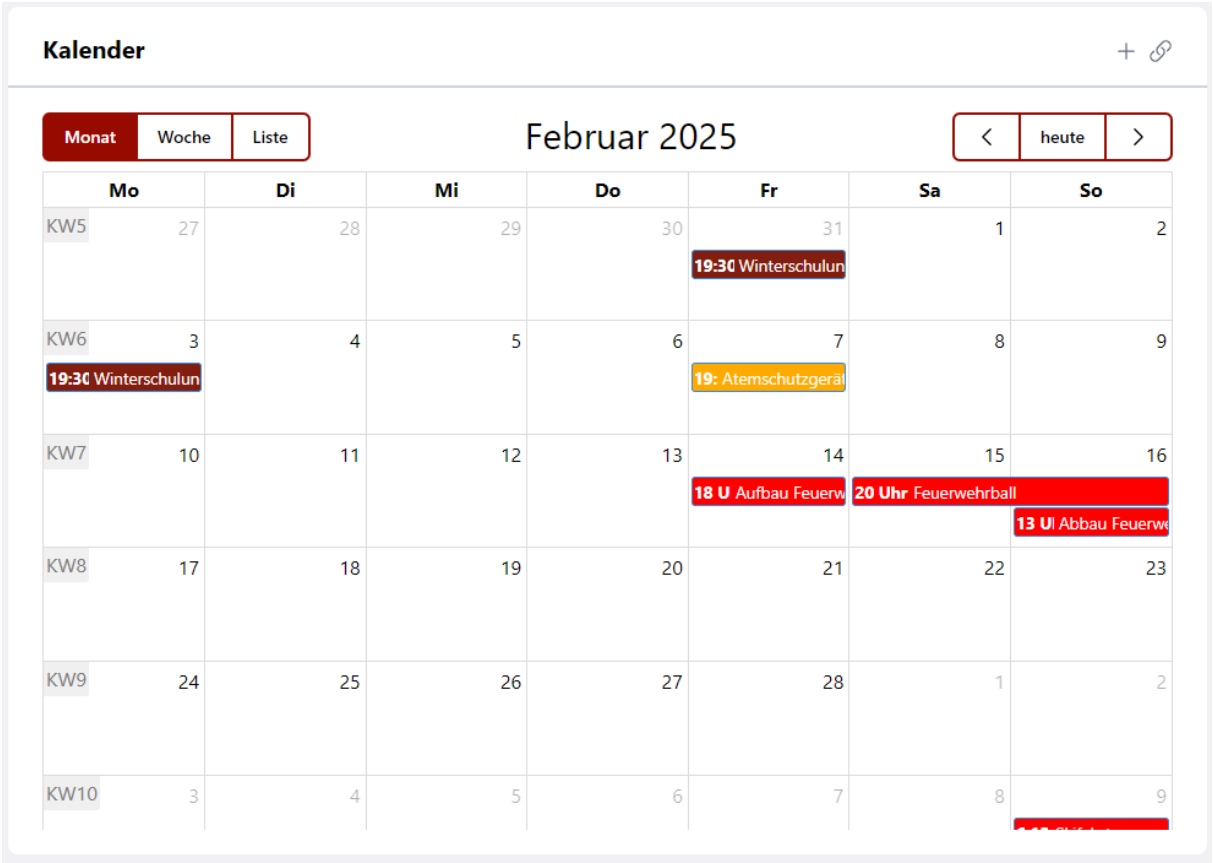
Überprüfung auf sich überschneidende Zeiträume oder Doppelerfassungen muss durch den Anwender erfolgen.

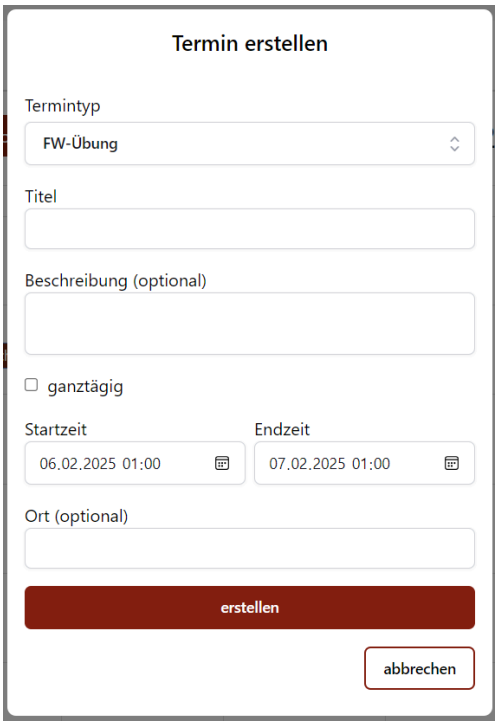
## 4.2 Kalender

FF Admin beinhaltet ein Modul für die Verwaltung von Terminen. Diese Termine können den Mitgliedern oder anderen Interessierten über die integrierte öffentliche Ansicht oder direkt als WebCal-Link zur Verfügung gestellt werden.




Auch im Kalendermodul müssen Stammdaten angelegt werden.
Die Stammdaten dienen dazu, die Kalendereinträge bestimmten Kategorien zuzuordnen. Die Kategorien enthalten eine Farbe und können dauerhaft öffentlich oder durch ein Passwort geschützt sein. Der Schutz einer Kategorie mit einem Passwort erschwert das automatisierte Testen und Auslesen von Terminen, da dann neben dem Namen der Kategorie auch das Passwort im Link angegeben werden muss.

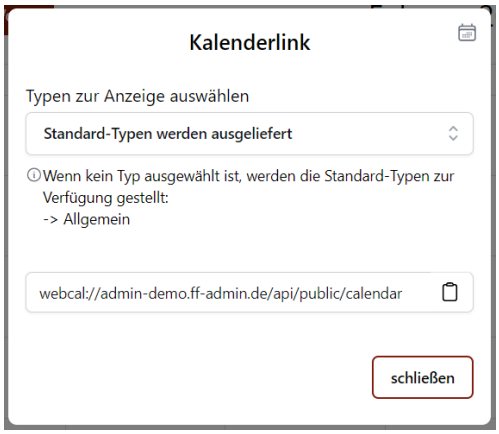



Der Kalender ermöglicht die direkte Verwaltung von Terminen. Durch die Auswahl von Tagen in der Monatsansicht oder Zeiträumen in der Wochenansicht können Zeitbereiche direkt in das Formular zur Erstellung eines Termins übernommen werden.
Der vorgewählte Zeitraum kann im erscheinenden Formular jederzeit geändert werden. Bei ganztägigen Terminen können dann nur noch direkte Datumsangaben und bei zeitbegrenzten Terminen Daten mit Uhrzeit ausgewählt werden. Wird die Option **ganztägig** wieder abgewählt, wird als Zeitraum standardmäßig 08:00 bis 09:00 Uhr vorbelegt.

Weiterhin besteht ein Termin aus einem Termintyp bzw. einer Kategorie, einem Titel, einer Beschreibung und einem Ort.
Die Textfelder sind frei wählbar und werden nicht auf Duplikate oder Korrektheit geprüft.

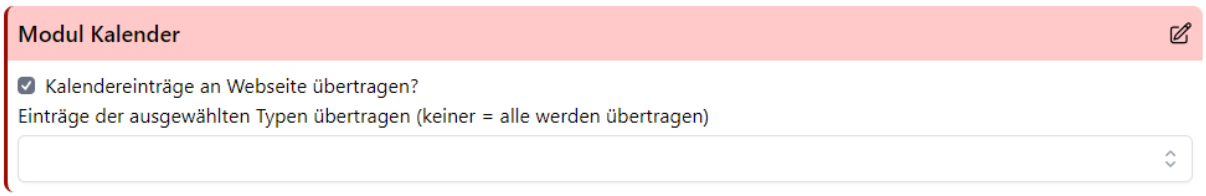
Alle Daten eines Termins sind analog zu diesem Formular in der öffentlichen Kalenderansicht sichtbar. Die Daten werden auch vollständig über WebCal übertragen.
Die öffentliche Ansicht des Kalenders ist in FF Admin unter dem URL-Pfad `/public/calendar` verfügbar und zeigt alle Termine der Kategorien an, die mit `Standard Kalender Auslieferung` erstellt wurden. Die Ansicht kann als angemeldeter Nutzer über das Kalender-Icon im Kalender-Link-Dialog erreicht werden.
Die Kategorien können weiterhin verwendet werden, um spezielle Links anzulegen, die Termine nur bestimmten Mitgliedern zur Verfügung stellen.




Die Freigabe des Kalenders mit einem WebCal-Link kann über eine eigene Dialogbox verwaltet werden. Die Dialogbox kann über das Link-Icon in der rechten oberen Ecke der Kalenderansicht geöffnet werden.

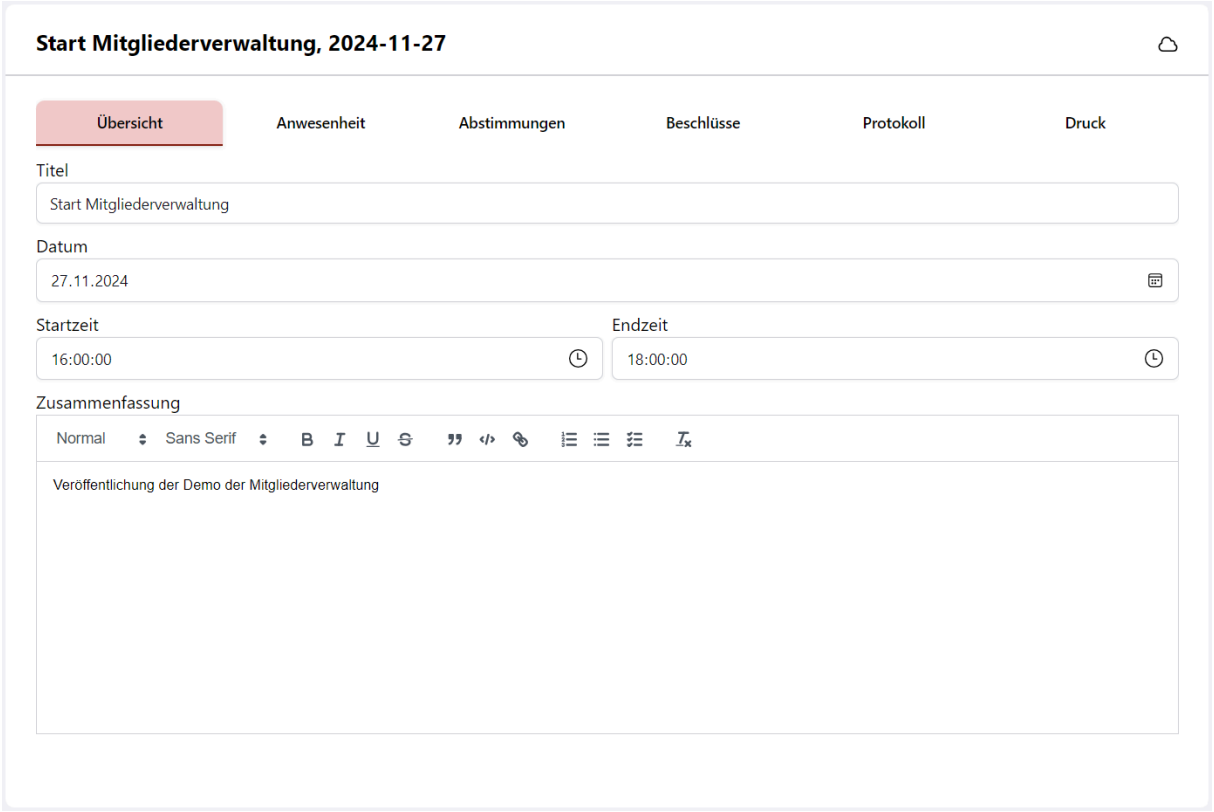
Hier kann ausgewählt werden, welche Kategorien über einen Link verfügbar sein sollen. Die Kategorien werden - bei Bedarf mit der Passphrase - automatisch in den Link geschrieben. Sind Kategorien ausgewählt, kann zusätzlich die Standard-Auslieferung aktiviert werden. Ansonsten werden nur Termine der ausgewählten Kategorien ausgeliefert. Dieser kann dann kopiert und direkt in einen Kalender wie der von Google oder von Apple eingefügt werden.

In den Modulspezifischen Einstellungen (Abschnitt 4.20) kann ein automatischer sync mit FF Webpage eingerichtet werden.




Für die Synchronisierung muss zusätzlich die Moduleinstellung `Webpage WebAPI Connect` eingerichtet sein. Wird die Synchronisierung für die Kalendereinträge aktiviert, kann zusätzlich noch ausgewählt werden, ob alle angelegten oder nur ausgewählte Terminarten synchronisiert werden sollen.

## 4.3 Protokolle




FF Admin bietet auch die Möglichkeit, Protokolle zu erfassen. Die Protokolle können An- und Abwesenheiten, Abstimmungen, Beschlüsse sowie die besprochenen Themen enthalten. Die Protokolle können auch als PDF exportiert werden.

Jedes Protokoll enthält einen Titel sowie ein Datum mit Anfangs- und Endzeit. Zusätzlich kann eine Zusammenfassung verfasst werden, die beim Ausdruck des Protokolls an den Anfang gestellt wird.
Die **Anwesenheit** kann nur Mitglieder erfassen, die in der Mitgliederverwaltung angelegt wurden. Bei den Anwesenheiten können Mitglieder gesucht und dem Protokoll hinzugefügt werden. Zusätzlich kann gespeichert werden, ob ein Mitglied entschuldigt oder unentschuldigt abwesend war. Je nach Anwesenheit werden die Mitglieder auf dem pdf unter verschiedenen Positionen ausgedruckt.
**Abstimmungen** können ebenfalls erfasst werden. Eine Abstimmung besteht aus einem Titel und einer Beschreibung. Zusätzlich kann das Ergebnis mit der Anzahl der Ja-, Nein- und Enthaltungsstimmen angegeben werden.
Die **Beschlüsse** sind als separate Auflistung von Sitzungen gedacht. Diese sollen noch einmal klar und gesondert hervorheben, was alles beschlossen wurde.

Das **Protokoll** selbst besteht aus vielen Punkten, die in der Sitzung besprochen wurden. Es wird immer mit Titel und Beschreibung erfasst.
Jedes Protokoll kann gedruckt werden. Die erstellten pdf-Dateien können heruntergeladen oder direkt in FF Admin angezeigt werden. Jedes pdf enthält die Daten des Protokolls und ordnet diese in Listen an.

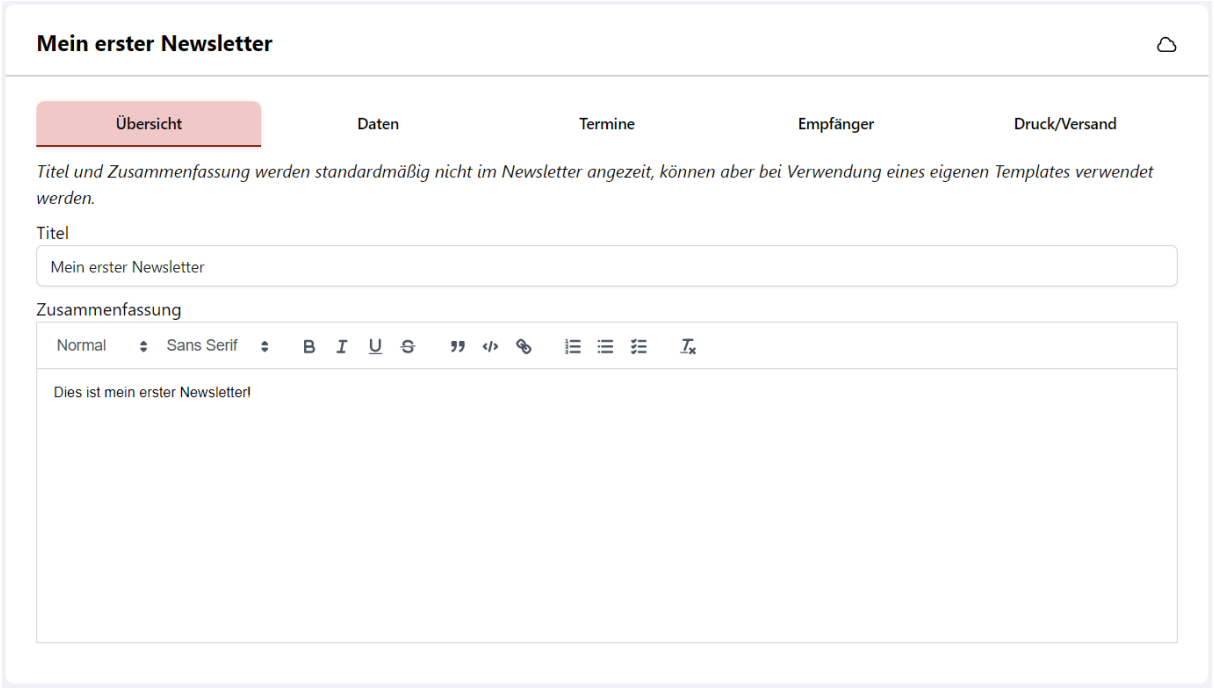


Die einzelnen Einträge können über Pfeile am Ende eines Eintrags nach oben oder unten verschoben werden. Dadurch kann die Reihenfolge geändert werden, wenn Inhalte nachträglich ergänzt werden.
Die Sortierung wird im Druck berücksichtigt.
Um Einträge löschen zu können, muss der Mülleimer mit einer Doppelte-Klick-Bestätigung angeklickt werden. Nach erstmaligem Klicken, beginnt der Mülleimer nach kurzer Verzögerung für 2 Sekunden weiß zu pulsieren. Während dieser Dauer wird ein zweiter Klick auf den Eimer als Bestätigung zum löschen gewertet.

**Hinweis:** Die Wolke in der rechten oberen Ecke zeigt den Status der Synchronisation an. Durch Klicken auf eine Wolke mit einem Pfeil kann die Synchronisation manuell durchgeführt werden. Ansonsten wird alle 10 Sekunden synchronisiert.

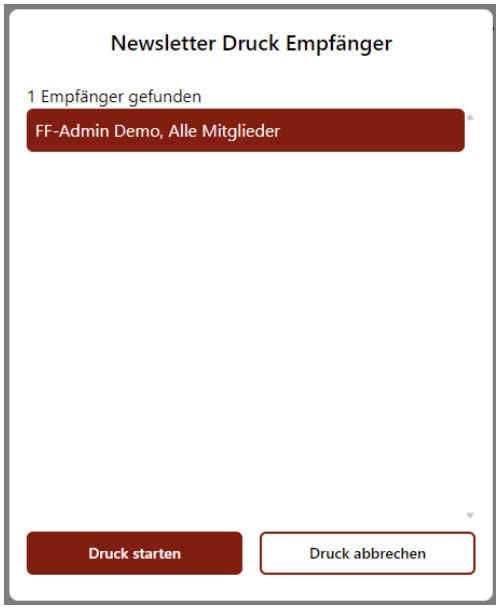
**Hinweis:** Ein Protokoll kann derzeit nicht kollaborativ geschrieben werden. Bei paralleler Arbeit am Protokoll überschreiben sich die Inhalte der Bearbeiter immer gegenseitig.

## 4.4 Newsletter



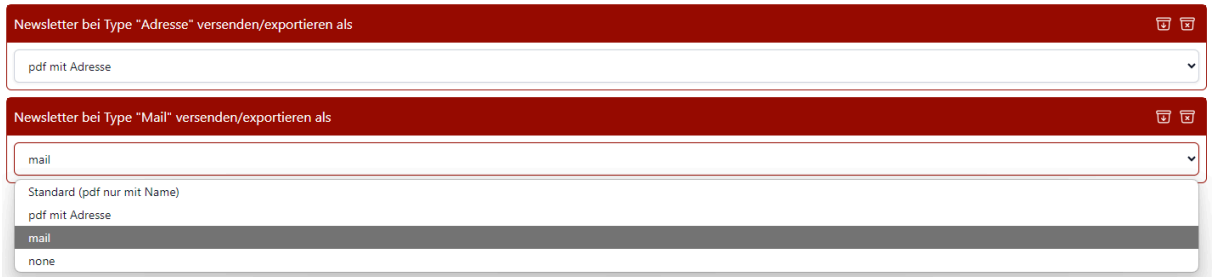

Ein weiteres Modul ermöglicht den Versand von Newslettern. Die Newsletter können per E-Mail versendet oder als PDF exportiert werden. Die Newsletter können mit eigenen Texten und Terminen gestaltet werden.

Ein Newsletter kann mit einem Titel und einer Zusammenfassung beschrieben werden. Diese Inhalte sind nicht direkt in den Exporten enthalten und dienen in erster Linie der besseren Übersicht.
Die **Daten**, die für einen Newsletter angegeben werden können, sind eine Überschrift, ein (Einleitungs-)Text und eine Signatur.
Zusätzlich können **Termine** ausgewählt werden, die dann direkt im Newsletter aufgelistet werden. Für jeden Termin können Titel und Beschreibung geändert werden. Dient ein Newsletter der reinen Informationsvermittlung, kann der Text verwendet werden. Es müssen keine Termine hinzugefügt werden.
Die **Empfänger** des Newsletters können automatisiert über Abfragen (Abschnitt 4.8) erfasst werden. Die Empfänger werden dann mit ihrem bevorzugten Empfangs-Kommunikationskanal angezeigt. Die Empfänger werden aus der Abfrage anhand einer vorhandenen `memberId` oder `id` Spalte ermittelt.
Zusätzlich können Empfänger manuell hinzugefügt werden, falls diese nicht in der Abfrage enthalten sind. Ein doppelter Versand aufgrund der manuellen Angabe und Bezug über einen Query ist nicht möglich.




Der **Export** kann als pdf oder als Mailversand erfolgen. Jedes Mitglied, das eine Mailadresse für den Empfang angegeben hat, erhält eine Mail mit dem Hauptteil des Newsletters. Für alle anderen wird ein PDF erzeugt. Das PDF enthält in der Fußzeile den Namen und optional eine Adresse, falls eine solche für den Versand des Newsletters beim Mitglied angegeben ist.

Bevor ein Druck oder Versand gestartet werden kann, wird ein Dialog angezeigt, über welchen die Empfänger eingesehen werden können.
Mit `Mail Vorschau` wird eine Testmail an die E-Mail-Adresse des eigenen Kontos gesendet. Für verwaltete Konten ohne E-Mail-Adresse ist das nicht möglich.
Ob ein Newsletter als pdf oder Mail erstellt wird, kann in der Newsletter-Konfiguration definiert werden.




Für jede Kommunikationsart kann festgelegt werden, ob eine E-Mail versendet, ein PDF mit Adresse oder ein PDF ohne Adresse erzeugt werden soll. Sind die erforderlichen Daten für die Auswahl Mailversand oder PDF-Druck mit Adresse nicht vorhanden, wird als Fallback auf ein PDF ohne Adresse umgeschaltet.

Um den Ausdruck zu vereinfachen, werden alle erzeugten PDFs in einem einzigen PDF zusammengefasst.

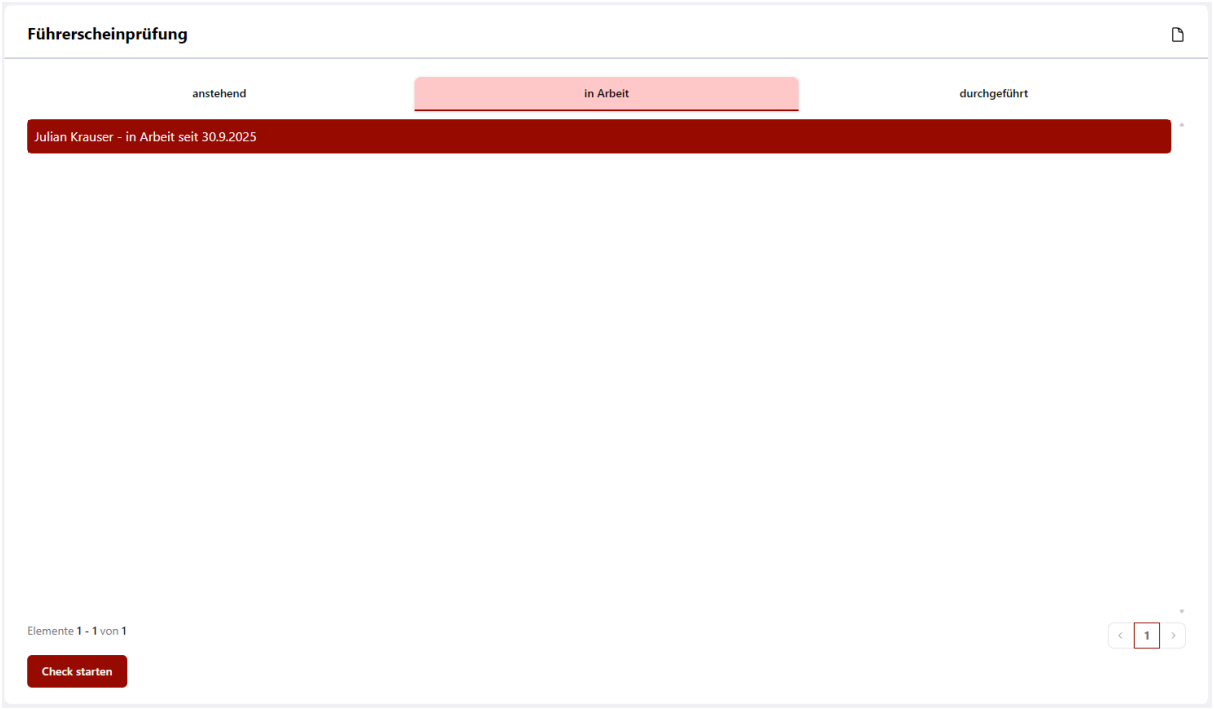
Zusätzlich wird eine ICS-Datei erstellt, die alle Termine enthält, die dem Newsletter hinzugefügt wurden. Diese wird ebenfalls direkt mit der Mail versendet. Zusätzlich wird noch ein PDF erstellt, welches in der Fußzeile an den Verein adressiert ist und somit an alle Mitglieder z.B. in einer WhatsApp Gruppe verteilt werden kann.

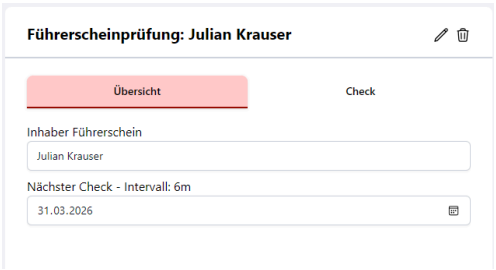
**Hinweis:** Die Wolke in der rechten oberen Ecke zeigt den Status der Synchronisation an. Durch Klicken auf eine Wolke mit einem Pfeil kann die Synchronisation manuell durchgeführt werden.

Ansonsten wird alle 10 Sekunden synchronisiert.

**Hinweis:** Ein Newsletter kann derzeit nicht kollaborativ geschrieben werden. Bei paralleler Arbeit am Newsletter überschreiben sich die Inhalte der Bearbeiter immer gegenseitig.

## 4.5 Führerscheinkontrolle



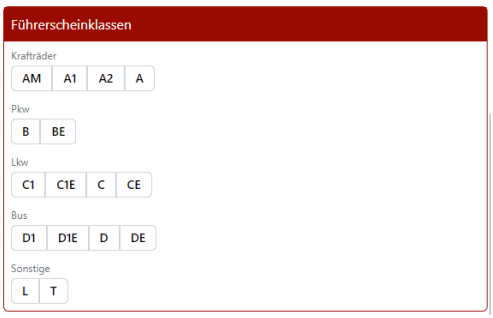

Mit diesem Modul können Führerscheine erfasst und dokumentiert werden. Fahrzeughalter erfüllen damit die gesetzliche Verpflichtung, die Führerscheine ihrer Fahrer regelmäßig zu überprüfen.
Das Modul zur Führerscheinkontrolle ermöglicht das Hochladen von Bildern der Vorder- und Rückseite des Führerscheins sowie das Hinzufügen von Anmerkungen. Erinnerungen zur erneuten Prüfung werden automatisch je nach Einstellung in den Moduleinstellungen (Abschnitt 4.20) angezeigt.



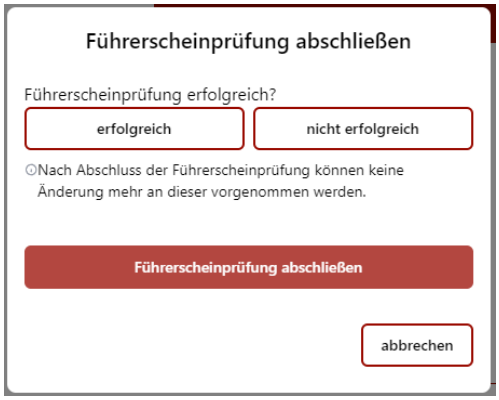




Hochgeladene Bilder können automatisiert zugeschnitten werden. Das Zuschneiden erfolgt mithilfe von OpenCV und ermöglicht oftmals eine Erfassung der relevanten Bereiche. Zusätzlich besteht die Möglichkeit, Bilder manuell zu drehen, um die Ausrichtung zu korrigieren.

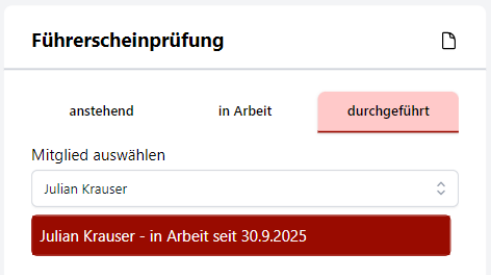



Weiterhin können die ausgestellten Führerscheinklassen eigens erfasst werden. Diese werden auch auf dem Listenausdruck mit dargestellt.




In dem Dialog zum Abschließen einer Führerscheinprüfung muss angegeben werden, ob diese erfolgreich war oder nicht. Auch in der Ansicht einer abgeschlossenen Führerscheinprüfung wird der Erfolgsstatus angezeigt.

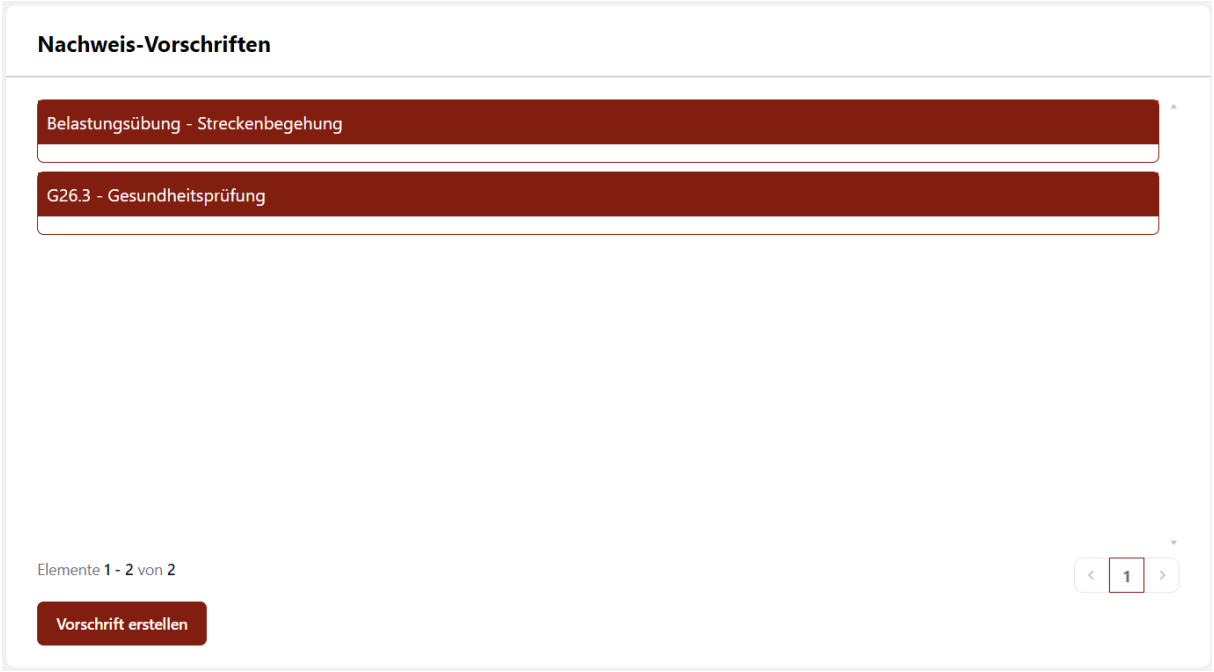
Auch können durchgeführte Führerscheinkontrollen zu einem Mitglied gesucht werden. Für jedes Mitglied werden alle durchgeführten Checks aufgelistet und mit dem jeweiligen Stand angezeigt.



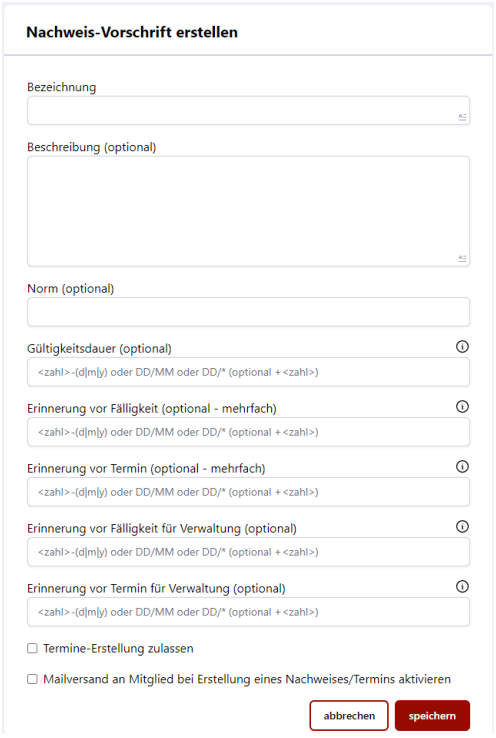
## 4.6 Nachweisverwaltung (Requirements & Compliance Tracking)

FF Admin ermöglicht die Verwaltung von Nachweisen, die Mitglieder erbringen müssen, z.B. Erste-Hilfe-Kurse, medizinische Untersuchungen oder sonstige Qualifikationsnachweise. Das Modul gliedert sich in zwei Bereiche: die **Nachweis-Vorschriften** (Vorlagen inkl. Zuordnung, wer einen Nachweis erbringen muss) und die **Personennachweise** (die tatsächlich erbrachten Nachweise und die zugehörigen Termine).
**Hinweis:** Die Führerscheinkontrolle (Abschnitt 4.5) ist ein eigenständiges, spezialisiertes Modul und nicht Teil der Nachweisverwaltung.

### 4.6.1 Nachweis-Vorschriften

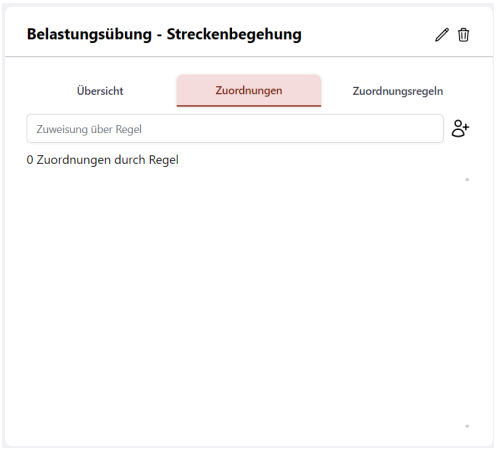



Eine Nachweis-Vorschrift definiert, welcher Nachweis benötigt wird, in welchem Turnus dieser erneuert werden muss und ob dafür ein Termin erforderlich ist.




Eine Vorschrift besteht aus einer Bezeichnung, einer optionalen Beschreibung sowie einer optionalen Norm-Angabe, über die z.B. auf eine gesetzliche Grundlage oder einen Standard verwiesen werden kann.
Die **Gültigkeitsdauer** ist optional und wird im Format aus Abschnitt 3.3 angegeben. Ist keine Gültigkeitsdauer hinterlegt, gilt ein erbrachter Nachweis unbefristet bzw. bis zu dem beim Nachweis individuell angegebenen Datum. Zusätzlich können mehrere **Erinnerungen vor Fälligkeit** sowie - sofern Termine für die Vorschrift zugelassen sind - mehrere **Erinnerungen vor Termin** hinterlegt werden. Diese Erinnerungen werden per Mail an die betroffenen Mitglieder gesendet.
Unabhängig davon können eine **Erinnerung vor Fälligkeit für Verwaltung** und eine **Erinnerung vor Termin für Verwaltung** (jeweils optional, nur ein Wert im Format aus Abschnitt 3.3) festgelegt werden. Diese richten sich nicht an die Mitglieder, sondern an Nutzer der Verwaltung, die die entsprechende Benachrichtigung in den Benachrichtigungseinstellungen (Abschnitt 5.2.2) aktiviert haben. Sie erhalten eine Sammelübersicht der bald fälligen Nachweise bzw. der bald anstehenden Termine.
Ist **Termine-Erstellung zulassen** aktiviert, können für Mitglieder mit dieser Vorschrift Nachweistermine angelegt werden (siehe unten).
Mit **Mailversand an Mitglied bei Erstellung eines Nachweises/Termins aktivieren** wird das betroffene Mitglied per Mail informiert, sobald ein Nachweis oder ein Termin für dieses angelegt wird. Wird ein Termin anschließend verschoben oder abgesagt, erhält das Mitglied ebenfalls eine Mail. Die Mails enthalten bei Terminen Datum und Uhrzeit, bei Nachweisen die Gültigkeit. Voraussetzung ist eine hinterlegte E-Mail-Adresse des Mitglieds.

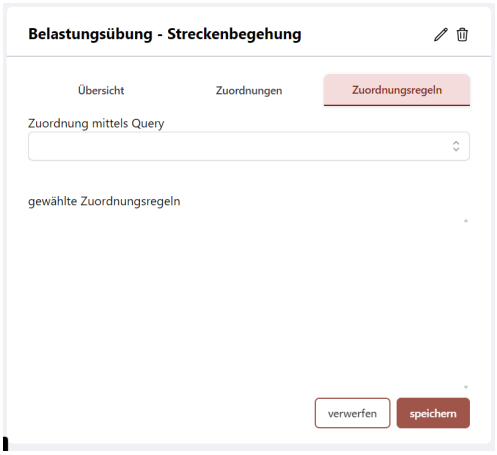
#### 4.6.1.1 Zuordnungen




Im Tab **Zuordnungen** wird festgelegt, welche Mitglieder einen Nachweis nach dieser Vorschrift erbringen müssen. Die Zuordnung erfolgt entweder automatisch über Zuordnungsregeln (siehe unten) oder - über das Icon in der Kopfzeile - manuell durch Auswahl einzelner Mitglieder. Beide Modi schließen sich pro Vorschrift gegenseitig aus: Sind Zuordnungsregeln hinterlegt, wird die manuelle Auswahl nicht verwendet.

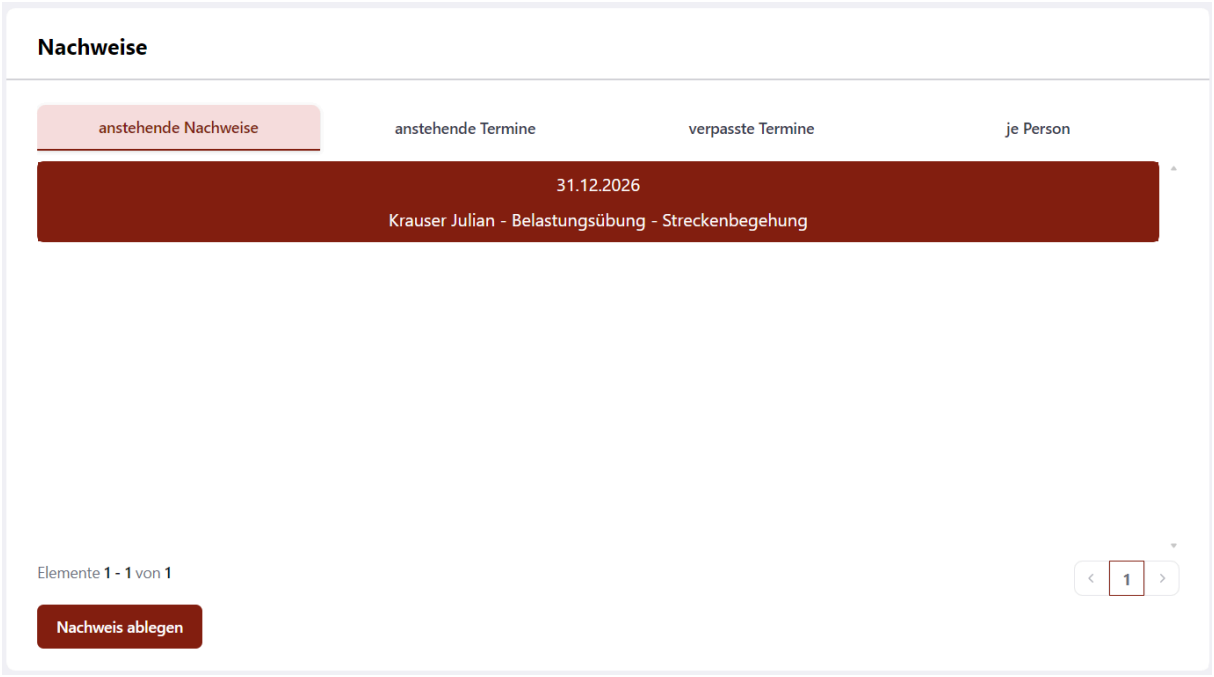
Bei jedem zugeordneten Mitglied wird angezeigt, ob eine E-Mail-Adresse hinterlegt ist. So ist erkennbar, welche Mitglieder bei aktiviertem Mailversand tatsächlich benachrichtigt werden können.

#### 4.6.1.2 Zuordnungsregeln

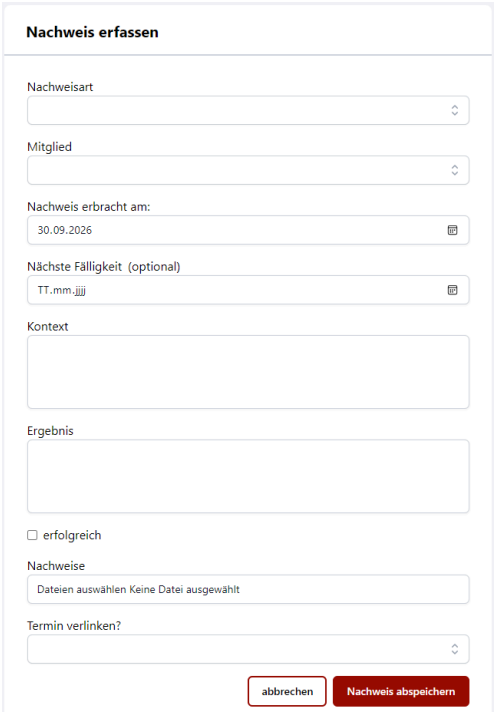



Über den Tab **Zuordnungsregeln** können eine oder mehrere gespeicherte Abfragen aus dem Query-Builder (Abschnitt 4.8) ausgewählt werden. Alle Mitglieder, auf die mindestens eine der gewählten Abfragen zutrifft, benötigen automatisch diesen Nachweis. Da Query-Ergebnisse sich mit den Mitgliederdaten laufend ändern können, wird die Zuordnung nicht gespeichert, sondern bei jedem Aufruf neu anhand der Abfragen ermittelt. So können z.B. alle Mitglieder mit einer bestimmten Qualifikation oder Funktion automatisch zur Nachweispflicht herangezogen werden.

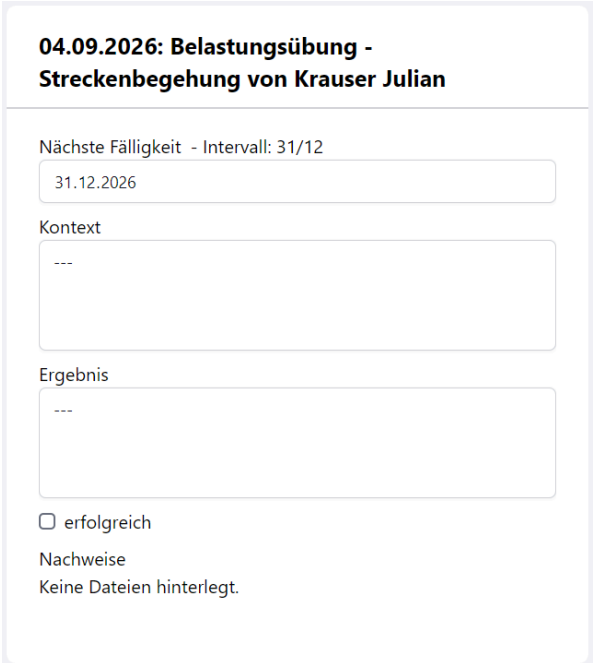
### 4.6.2 Personennachweise




Im Menü **Personennachweise** werden anstehende bzw. fällige Nachweise, anstehende und verpasste Termine sowie alle erbrachten Nachweise je Mitglied übersichtlich dargestellt.
Über das Filter-Symbol in der Kopfzeile kann eine **Vorschrift ausgewählt** werden, sodass nur Einträge zu dieser Vorschrift angezeigt werden (auch bei den verpassten Terminen). Ein Punkt am Symbol zeigt einen aktiven Filter an, über **abwählen** wird er entfernt.

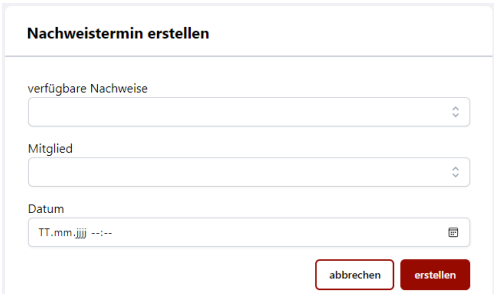



Aus der Übersicht der anstehenden Nachweise heraus kann direkt ein neuer Nachweis für das jeweilige Mitglied und die jeweilige Vorschrift erfasst werden. Ein Nachweis kann auch unabhängig davon frei über **Nachweis ablegen** erstellt werden. Dabei wird zuerst die **Nachweisart** und anschließend das **Mitglied** gewählt. Die Auswahl der Mitglieder beschränkt sich auf diejenigen, für die die gewählte Vorschrift gilt.
Pflichtangabe ist das Datum **Nachweis erbracht am**, das standardmäßig auf den aktuellen Tag gesetzt ist und bei Bedarf angepasst werden kann.
Neben Kontext und Ergebnis als Freitext kann angegeben werden, ob der Nachweis erfolgreich war, und es können Nachweis-Dateien (Bilder oder PDF) hochgeladen werden. Hat die Vorschrift ein Intervall, wird die **nächste Fälligkeit** automatisch ausgehend vom Datum **Nachweis erbracht am** vorbelegt, bleibt aber optional und kann angepasst werden. Ist der Nachweis das Ergebnis eines Termins, kann dieser Termin verlinkt werden.




Ein erfasster Nachweis zeigt alle Angaben inklusive der hochgeladenen Nachweis-Dateien in der Detailansicht an. Die hochgeladenen Dateien werden dabei mit ihrem ursprünglichen Dateinamen angezeigt.
Durch anklicken hochgeladener Dateien wird ein Dialog zur Anzeige geöffnet.

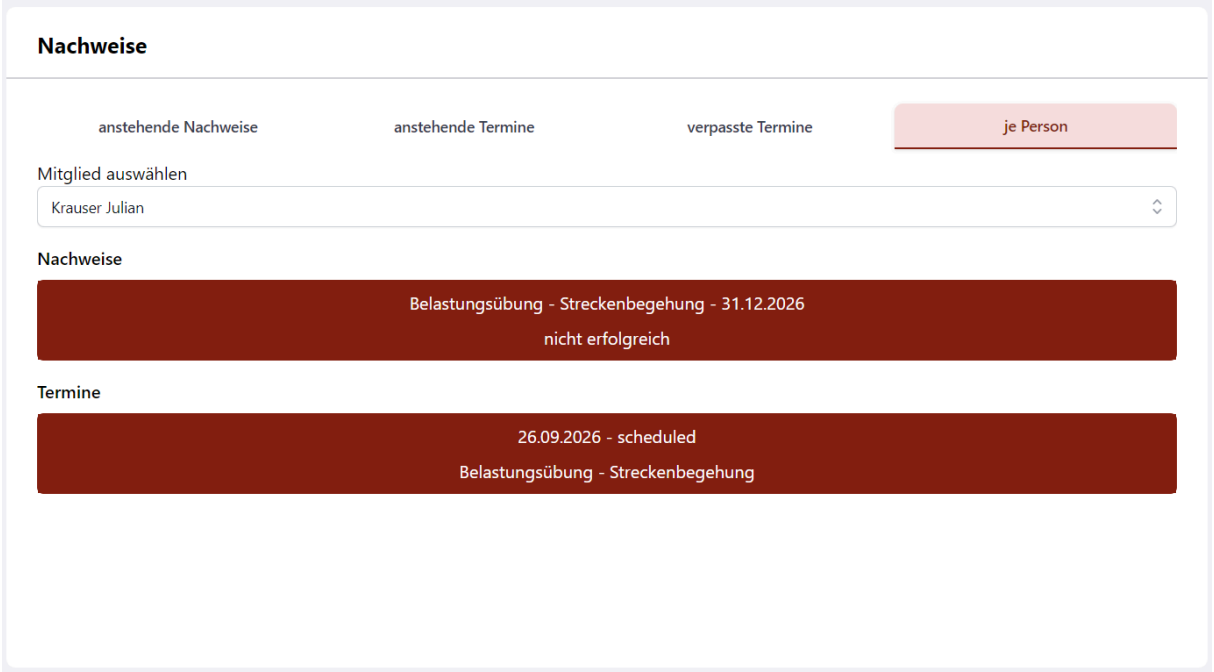
#### 4.6.2.1 Nachweistermine




Für Vorschriften mit aktivierter Termine-Erstellung können Nachweistermine angelegt werden. Ein Termin verknüpft eine Nachweisart, ein Mitglied und einen Zeitpunkt. Wie beim Nachweis wird zuerst die Nachweisart gewählt, danach werden nur die zugehörigen Mitglieder zur Auswahl angeboten.

Ein Termin durchläuft die Status `geplant`, `abgeschlossen`, `verpasst` oder `abgesagt`. Wird ein Termin abgeschlossen, kann direkt der resultierende Nachweis erfasst und mit dem Termin verknüpft werden.

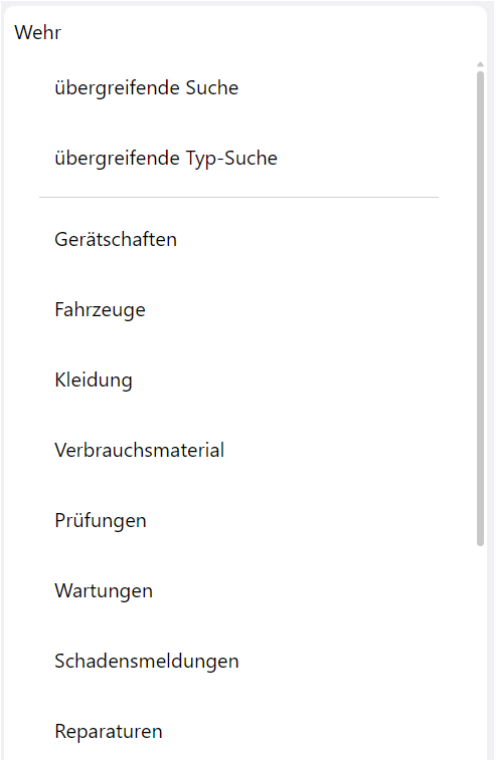


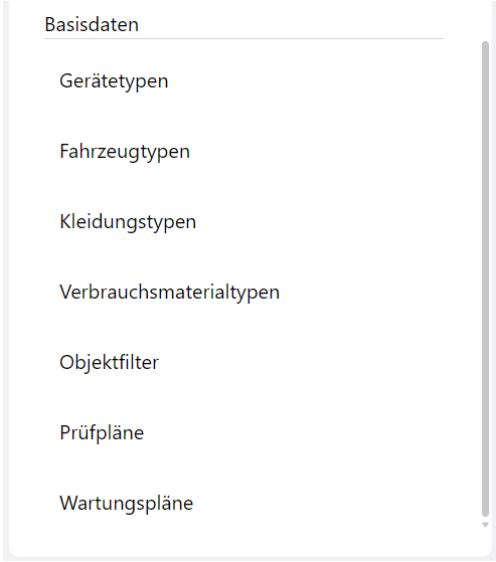




Im Tab **je Person** können alle bisher erbrachten Nachweise eines einzelnen Mitglieds gesucht und eingesehen werden.

**Hinweis:** Nutzer können in den Benachrichtigungseinstellungen (Abschnitt 5.2.2) festlegen, ob sie automatisch benachrichtigt werden möchten, wenn ein Nachweis gemäß den hinterlegten Erinnerungen fällig wird oder ein Termin ansteht.

## 4.7 Geräte- & Fahrzeug- & Kleidungsverwaltung



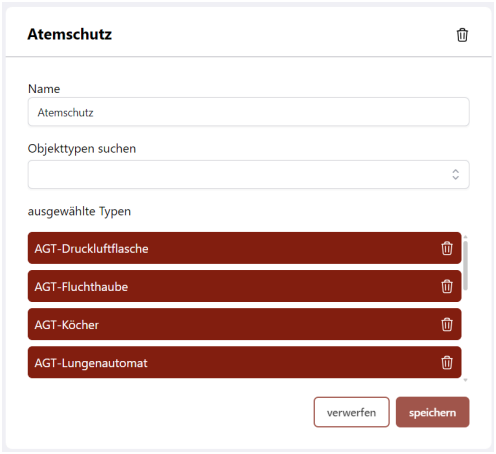




Die Verwaltung von Geräten, Fahrzeugen und Kleidung ermöglicht eine strukturierte Erfassung aller relevanten Objekte einer Feuerwehr oder eines Vereins. Für jedes dieser Objekte können Reparaturen, Schadensmeldungen, Wartungen und Prüfungen dokumentiert werden.
Um Einträge für Geräte, Fahrzeuge oder Kleidung anlegen zu können, müssen zunächst entsprechende Basisdaten definiert werden. Diese Basisdaten sorgen dafür, dass alle Einträge eines Typs über einheitliche Eigenschaften und Prüfungsoptionen verfügen.

Das Modul Verbrauchsmaterial ermöglicht die Verwaltung von Standorten, Mengen und Status aller Verbrauchsmaterialien und sorgt so für einen stets aktuellen Überblick.

**Hinweis:** Die Module zur Verwaltung von Geräten, Fahrzeugen und Kleidung gewährleisten eine nachvollziehbare Datenhistorie. Einträge können nach ihrer Erstellung nur noch eingeschränkt geändert und in der Regel nicht gelöscht werden.

### 4.7.1 Objektfilter

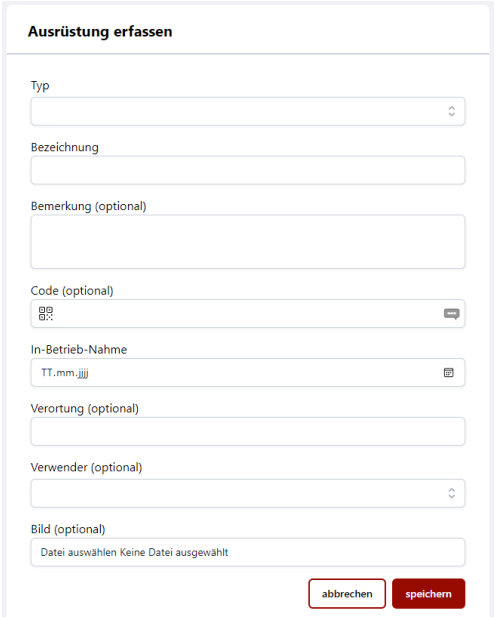



Ein Filter kann erstellt und Objekttypen diesem Filter zugeordnet werden. Diese zugeordneten Typen ermöglichen dann das Filtern auf sämtlichen Seiten, auf denen Daten mit Objekttypen verfügbar sind: in der Definition von den Typen, Wartungs- und Prüfpläne, bei Instanzen von Objekten sowie bei zugehörigen Daten wie Wartungen, Prüfungen, Schadensmeldungen und Reparaturen.

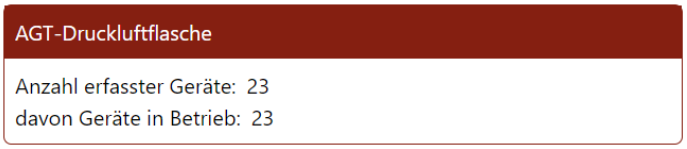
In den Ansichten, in denen Filtern verfügbar ist, kann über ein Icon mit Dropdown-Menü in der Kopfleiste gefiltert werden. Die Filter werden lokal auf dem Gerät gespeichert.

### 4.7.2 Geräte, Fahrzeuge & Kleidung

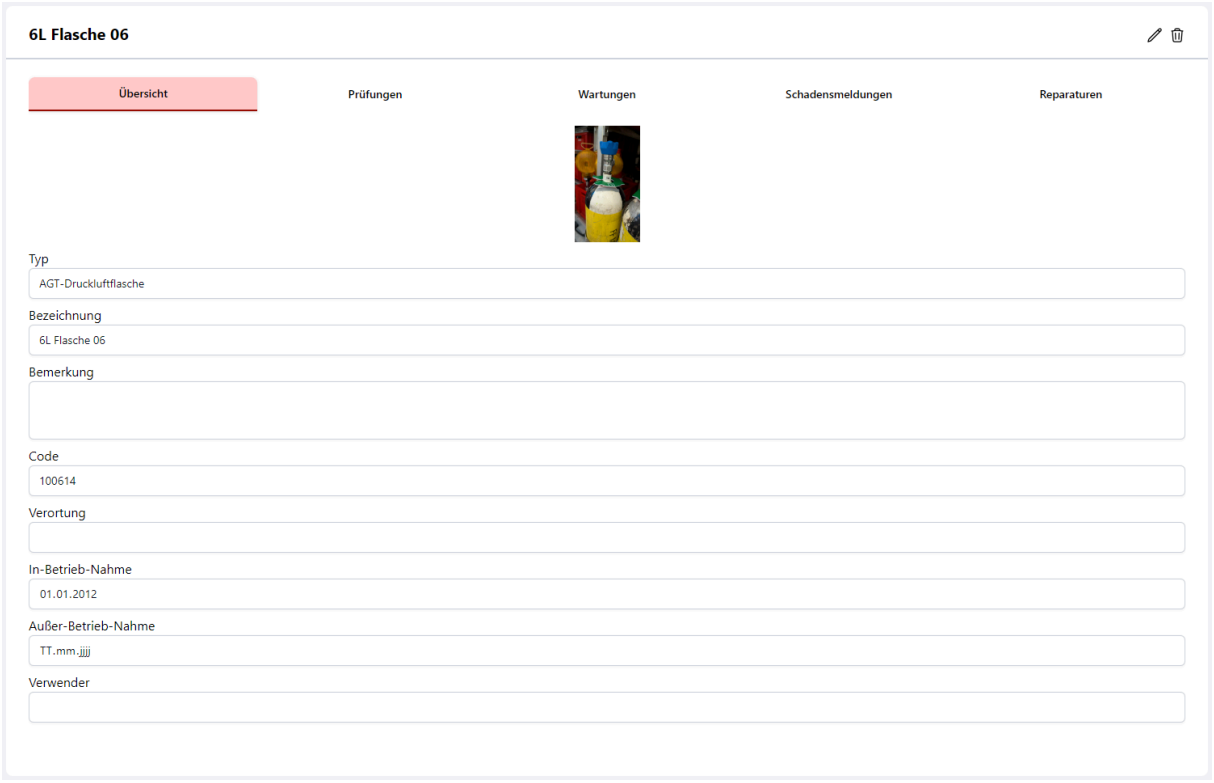
Im Folgenden werden die Eintragsarten Gerät, Fahrzeug und Kleidung exemplarisch anhand eines Geräteeintrags erläutert. Alle drei Arten verfügen über identische Datenfelder und bieten die gleichen Funktionen zur Verwaltung. Bei allen Objekten kann zusätzlich ein zugeordnetes Mitglied erfasst werden. Fahrzeuge haben zusätzliche Felder für das Kennzeichen und die FIN.




Nachdem ein Typ für ein Gerät, Fahrzeug oder Kleidungsstück angelegt wurde, können diesem Typ konkrete Instanzen zugewiesen werden. Ein Typ enthält dabei die zugehörigen Prüf- und Wartungspläne, die für alle zugeordneten Instanzen gelten.



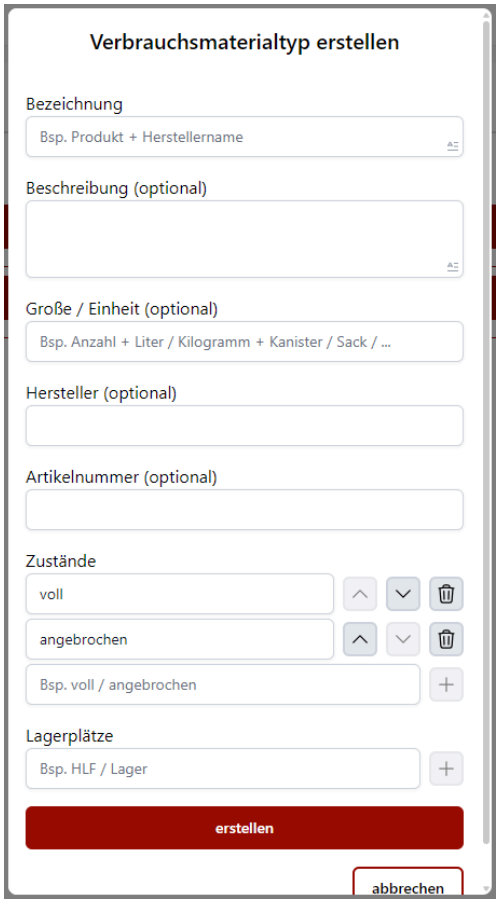

Zur besseren Übersicht zeigt der Typ zudem an, wie viele Instanzen es insgesamt gibt und wie viele davon aktuell in Betrieb sind.




Eine erstellte Instanz ermöglicht die Anzeige der zugeordneten Prüfungen, Wartungen, Schadensmeldungen und Reparaturen. Zusätzlich kann das Bild durch anklicken vergrößert werden.

### 4.7.3 Verbrauchsmaterial

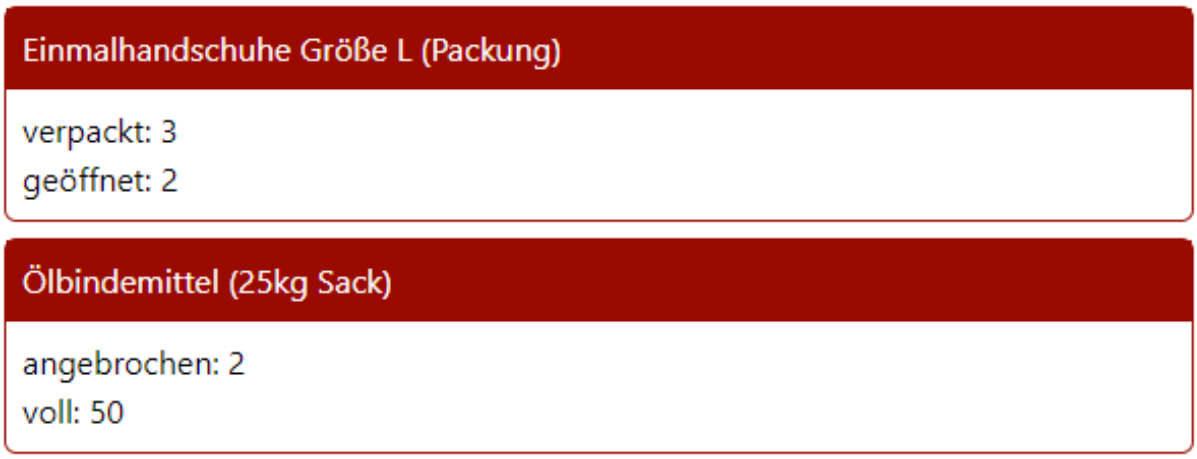
Das Modul Verbrauchsmaterial ermöglicht die Verwaltung von Material, das regelmäßig nachbeschafft werden muss.



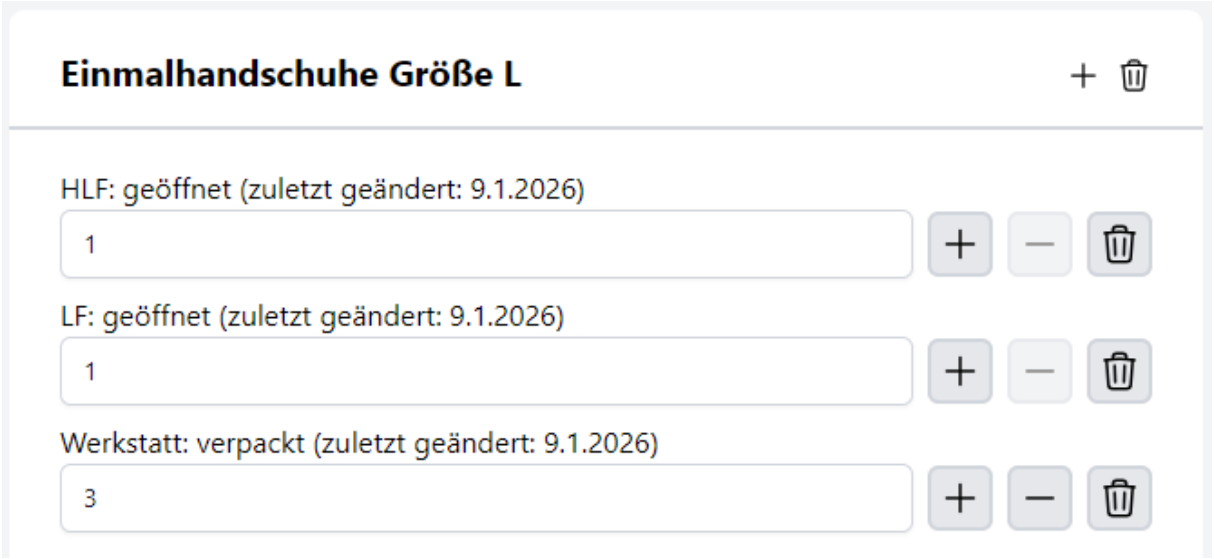

Bevor Material mit entsprechenden Mengen erfasst werden kann, muss zunächst ein Materialtyp angelegt werden, der die relevanten Produktinformationen enthält.

Am Materialtyp werden die möglichen Zustände (z.B. voll, angebrochen) und Lagerplätze gepflegt. Beim Erfassen eines Bestands werden diese nur noch ausgewählt, sodass keine abweichenden Schreibweisen entstehen. Wird ein Zustand oder Lagerplatz umbenannt, gilt der neue Name für alle Bestände. Solange noch Bestand darauf erfasst ist, kann er nicht entfernt werden. Anschließend können Materialbestände mit Zustand, Lagerplatz und Anzahl hinterlegt werden.

Zusätzlich lassen sich am Materialtyp Warnschwellen festlegen. Eine Warnschwelle bezieht sich immer auf genau einen Zustand und gilt entweder für den Gesamtbestand über alle Lagerplätze oder für einen einzelnen Lagerplatz. Sinkt der Bestand unter die angegebene Anzahl, wird einmalig benachrichtigt, erst nach erneutem Auffüllen wieder. Wer die Warnung erhält, legt jeder Benutzer in seinen Benachrichtigungseinstellungen unter Verbrauchsmaterial fest (In-App, E-Mail oder Push).

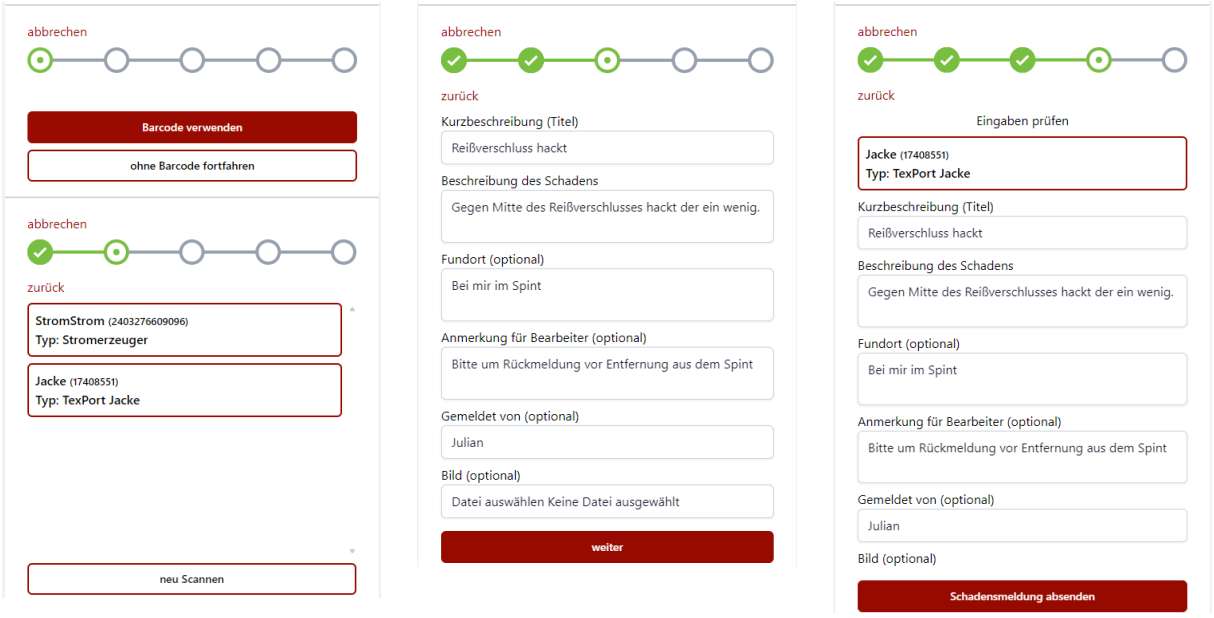


In der Vorschau wird das Material nach Zuständen gruppiert dargestellt. In der Detailansicht lassen sich die Mengen je Standort und Zustand anpassen.



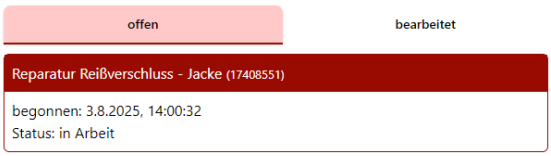

### 4.7.4 Schadensmeldungen & Reparaturen

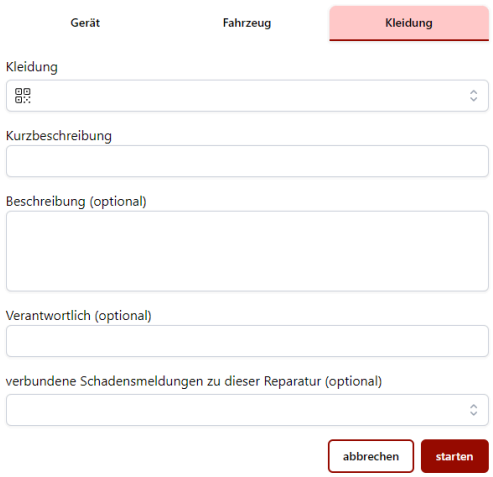
FF-Admin ermöglicht es, Schadensmeldungen auch von externen Personen entgegenzunehmen und direkt einem Objekt zuzuordnen. Für die Meldung ist keine Anmeldung erforderlich. Da jedem Objekt ein individueller Code (z.B. Barcode) zugewiesen werden kann, kann der Melder diesen scannen und die Schadensmeldung so eindeutig dem entsprechenden Objekt zuweisen. Anschließend können Reparaturen zu einer Schadensmeldung gezielt dokumentiert und verwaltet werden.






Eingereichte Schadensmeldungen werden übersichtlich nach ihrem Status angezeigt. Sobald eine Schadensmeldung zu einem Objekt vorliegt, erscheint sie auch in den Detailansichten des jeweiligen Objekts. Die Symbole in der roten Leiste geben Auskunft darüber, ob ein Bild, eine Position oder Kontaktdaten des Meldenden (Name) hinterlegt wurden und ob die Meldung bereits einer Reparatur zugeordnet ist. Bearbeiter können in der Schadensmeldung ausschließlich den Status und eine Anmerkung ergänzen. Der Verlauf der Änderungen wird dokumentiert und ist innerhalb einer Schadensmeldung einsehbar. Der ursprüngliche Inhalt der Meldung bleibt dabei unverändert und kann nur durch den Bearbeitungsprozess abgeschlossen werden. Eine Schadensmeldung gilt ebenfalls als abgeschlossen, sobald eine zugeordnete Reparatur beendet wurde.




Reparaturen können gezielt für ein bestimmtes Gerät, Fahrzeug oder Kleidungsstück angelegt werden. Bereits bei der Erstellung einer Reparatur lassen sich zugehörige Schadensmeldungen auswählen, die im Rahmen der Reparatur behoben werden sollen. Wie bei den Schadensmeldungen wird auch bei Reparaturen der Verlauf des gesetzten Status dokumentiert und ist innerhalb der Reparatur einsehbar. Die Auswahl der zugewiesenen Schadensmeldungen kann auch nachträglich angepasst werden.

**Hinweis:** Nutzer können in den Benachrichtigungseinstellungen (Abschnitt 5.2.2) festlegen, ob sie bei neu eingereichten Schadensmeldungen automatisch benachrichtigt werden möchten.

### 4.7.5 Wartungen

Wartungen sind planmäßige Instandhaltungs- oder Überholungsmaßnahmen, die in festgelegten Intervallen durchgeführt werden. Für jeden Typ von Gerät, Fahrzeug oder Kleidungsstück können individuelle Wartungspläne mit definierten Intervallen und Erinnerungsfristen hinterlegt werden.

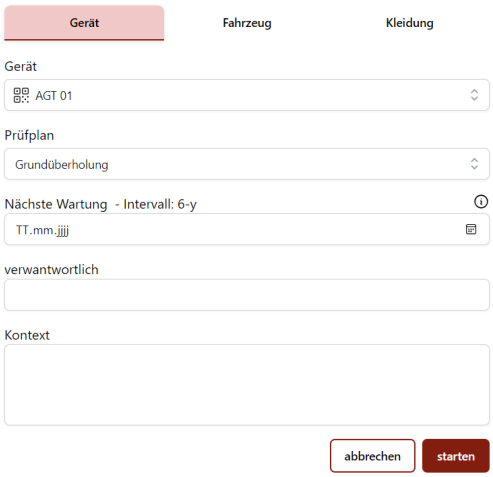
Die Intervalle sind optional und werden in Abschnitt 3.3 erklärt. So wird sichergestellt, dass alle erforderlichen Wartungen rechtzeitig erkannt und durchgeführt werden. Ein Wartungsplan beinhaltet zusätzlich eine Beschreibung, die die Wartungsinhalte beinhaltet.



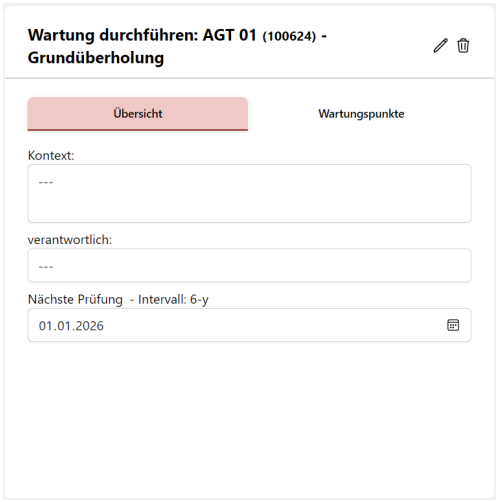


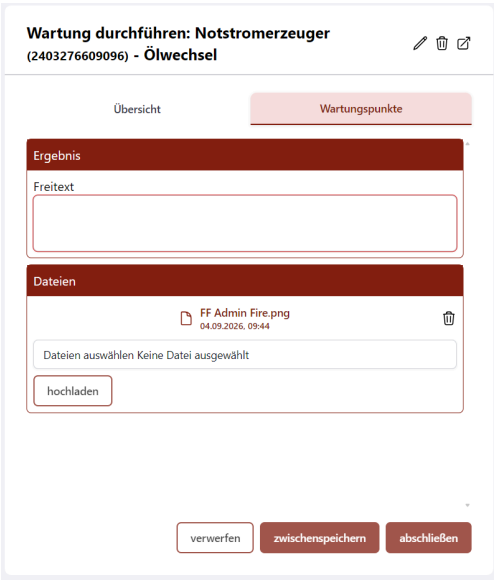

Wartungen werden in einem eigenen Menüpunkt übersichtlich nach den Kategorien „anstehend“, „in Arbeit“ und „nicht vorhanden“ dargestellt. Das Dokument-Icon oben rechts erlaubt den Druck einer Übersicht von bestehenden Wartung nach Geräte-, Fahrzeug- oder Kleidungstyp als Liste.




Anstehende oder fehlende Wartungen können direkt aus der Übersicht in die Erstellmaske übernommen werden. Dort lassen sich zusätzlich Verantwortlichkeiten, wie beispielsweise externe Dienstleister, sowie ein Kontext für die Wartung festlegen. Hat ein Wartungsplan ein Intervall, so muss das Datum für die nächste Wartung angegeben werden. Dieses Datum wird wiederum für die Anzeige innerhalb der anstehenden Wartungen verwendet.




Während eine Wartung läuft, können zusätzlich beliebige Dokumente (z.B. Fotos oder Belege) hochgeladen werden. Die hochgeladenen Dateien werden dabei mit ihrem ursprünglichen Dateinamen angezeigt.



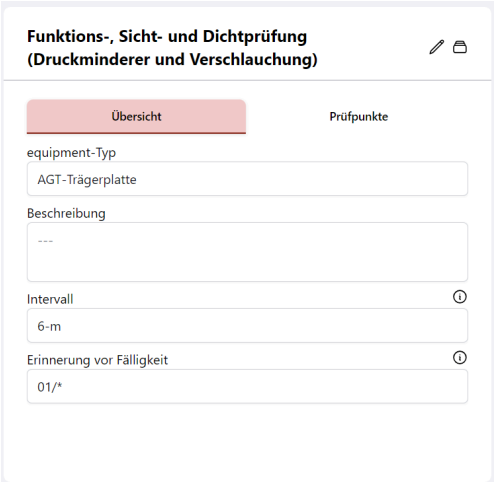

Wird eine Wartung abgeschlossen, wird ein pdf erstellt, welches über die Anwendung heruntergeladen und angezeigt werden kann.

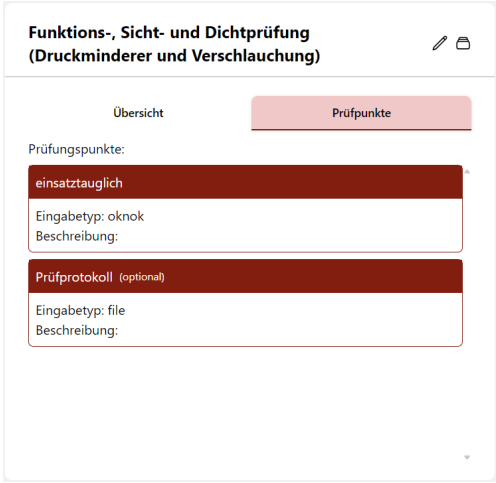
In dem Dialog zum Abschließen einer Wartung muss angegeben werden, ob diese erfolgreich war oder nicht. Sollte die Wartung nicht erfolgreich gewesen sein, kann direkt eine Reparatur angelegt werden. Der Ausdruck zur Wartung beinhaltet eine Zwischenüberschrift, welche den Erfolgsstatus angibt. Auch in der Ansicht einer abgeschlossenen Wartung wird der Erfolgsstatus angezeigt.

**Hinweis:** Nutzer können in den Benachrichtigungseinstellungen (Abschnitt 5.2.2) festlegen, ob sie automatisch benachrichtigt werden möchten, wenn eine Wartung gemäß dem hinterlegten Erinnerungsintervall fällig wird.

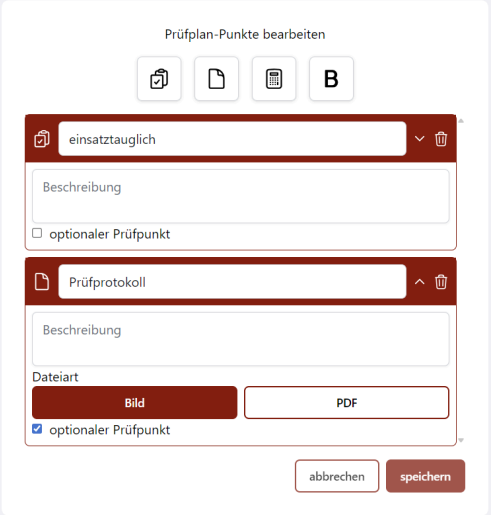
## 4.7.6 Prüfungen

Prüfungen sind regelmäßige Checks des Gerätezustands durch zum Beispiel einen Gerätewart oder externe Dienstleister. Für jeden Typ von Gerät, Fahrzeug oder Kleidungsstück können individuelle Prüfpläne mit definierten Intervallen und Erinnerungsfristen hinterlegt werden. Die Intervalle sind optional und werden in Abschnitt 3.3 erklärt. So wird sichergestellt, dass alle erforderlichen Prüfungen rechtzeitig erkannt und durchgeführt werden. Ein Prüfplan beinhaltet zusätzlich eine Beschreibung, die die Prüfinhalte beinhaltet. Zusätzlich können die Prüfpunkte selbst definiert werden.



Da Prüfungen individuell auf das jeweilige Gerät, Fahrzeug oder Kleidungsstück zugeschnitten sind, müssen für jeden Prüfplan spezifische Prüfpunkte definiert werden. Jeder Prüfpunkt kann einen Titel, eine Beschreibung sowie eine Option zur Kennzeichnung als „optional“ enthalten. Je nach Typ des Prüfpunkts stehen zusätzliche Felder zur Verfügung, beispielsweise zur Angabe eines optimalen Wertebereichs.

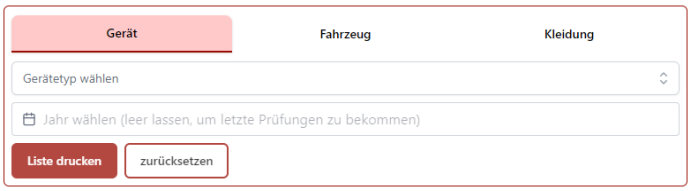


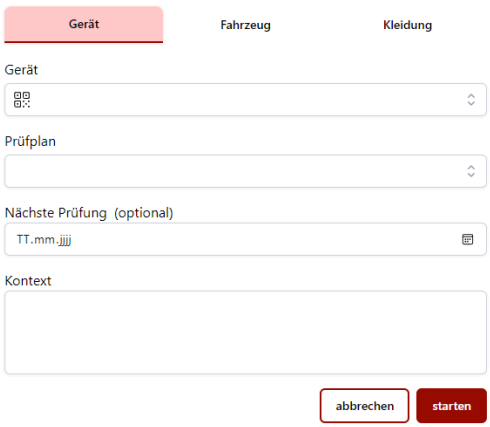

Folgende Prüfpunkttypen stehen zur Auswahl: Text, Zahl, Ok/Nicht-Ok und Bild-/PDF-Datei.

Die erfassten Prüfpunkte werden im Prüfprotokoll entsprechend dargestellt. Hochgeladene Dateien wie Bilder werden direkt im Protokoll angezeigt, PDF-Dateien werden als Anhang an das Prüfprotokoll angefügt.

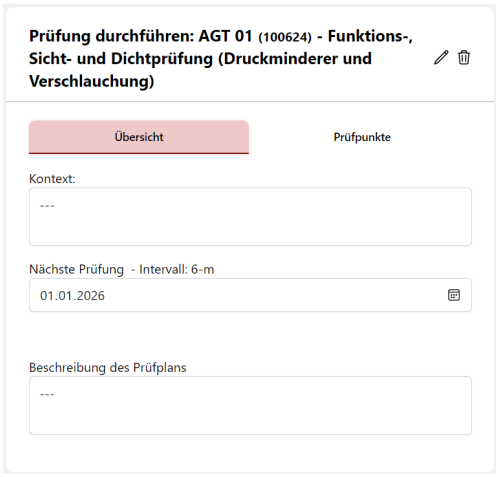


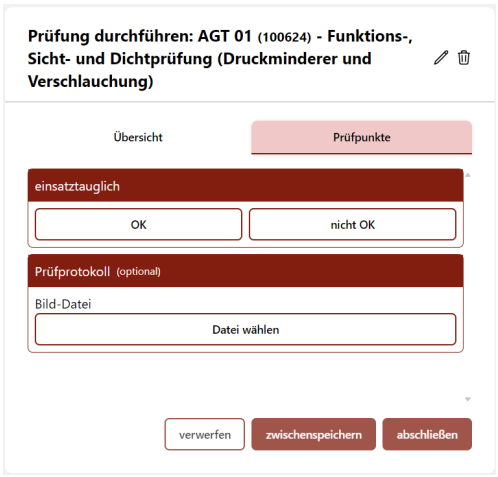
Prüfungen werden in einem eigenen Menüpunkt übersichtlich nach den Kategorien „anstehend“, „in Arbeit“ und „nicht vorhanden“ dargestellt. Das Dokument-Icon oben rechts erlaubt den Druck einer Übersicht von bestehenden Wartung nach Geräte-, Fahrzeug- oder Kleidungstyp als Liste.

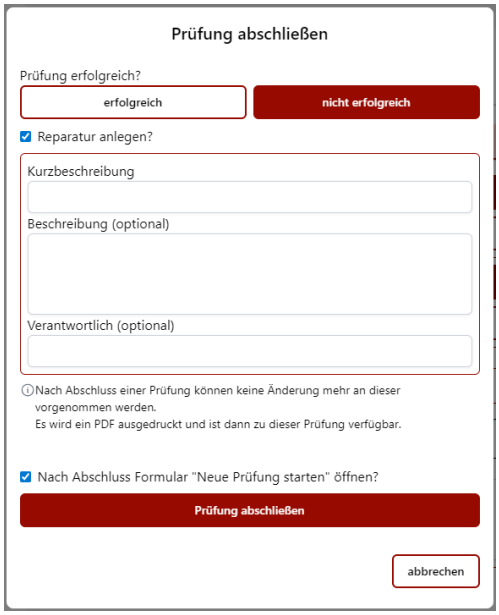



Anstehende oder fehlende Prüfungen können direkt aus der Übersicht in die Erstellmaske übernommen werden. Dort lässt sich ein Kontext für die Prüfung festlegen. Hat ein Prüfplan ein Intervall, so muss das Datum für die nächste Prüfung angegeben werden. Dieses Datum wird wiederum für die Anzeige innerhalb der anstehenden Prüfungen verwendet.





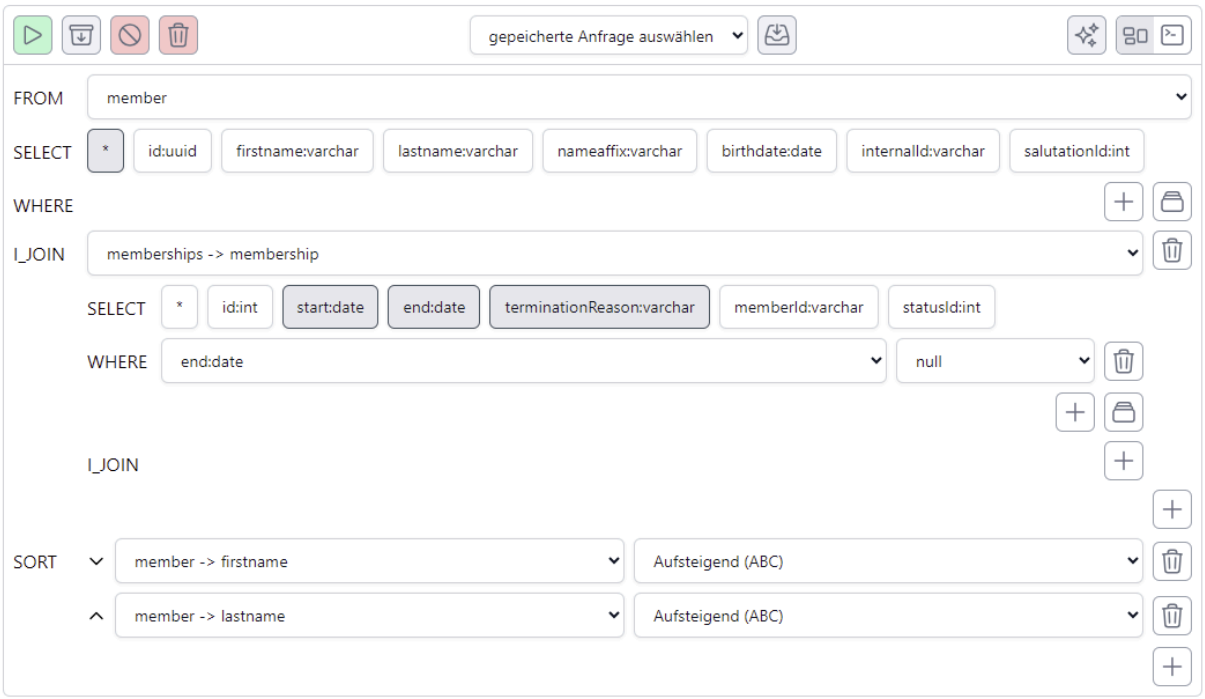

Wird eine Prüfung abgeschlossen, wird ein pdf erstellt, welches über die Anwendung heruntergeladen und angezeigt werden kann.

In dem Dialog zum Abschließen einer Prüfung muss angegeben werden, ob diese erfolgreich war oder nicht. Sollte die Prüfung nicht erfolgreich gewesen sein, kann direkt eine Reparatur angelegt werden. Der Ausdruck zur Prüfung beinhaltet eine Zwischenüberschrift, welche den Erfolgsstatus angibt. Auch in der Ansicht einer abgeschlossenen Prüfung wird der Erfolgsstatus angezeigt.

**Hinweis:** Nutzer können in den Benachrichtigungseinstellungen (Abschnitt 5.2.2) festlegen, ob sie automatisch benachrichtigt werden möchten, wenn eine Prüfung gemäß dem hinterlegten Erinnerungsintervall fällig wird.

## 4.8 Query Builder & Query Store

Queries sind Abfragen an die Mitgliederdaten, die von den Nutzern selbst definiert und in anderen Modulen wiederverwendet werden können, z.B. bei der Empfängerauswahl im Newsletter (Abschnitt 4.4) oder bei den Zuordnungsregeln der Nachweis-Vorschriften. Eine Query liefert immer eine Ergebnisliste von Datensätzen zurück, z.B. eine Liste von Mitgliedern, die bestimmte Bedingungen erfüllen.
Queries können entweder **visuell** mit dem Query Builder zusammengeklickt oder für erfahrene Nutzer direkt als SQL geschrieben werden. Im Folgenden wird ausschließlich der visuelle Builder erklärt - dafür sind keine SQL-Kenntnisse notwendig.
**Wichtig:** Mit Queries können Daten ausschließlich **gelesen**, niemals verändert werden.




### 4.8.1 Aufbau einer Abfrage

Jede Abfrage setzt sich aus bis zu fünf Bausteinen zusammen, die im Builder von oben nach unten sichtbar sind:

1. **Basistabelle (FROM):** Zuerst wird ausgewählt, auf welche Art von Daten sich die Abfrage bezieht, z.B. Mitglieder. Alle weiteren Bausteine beziehen sich auf diese Basistabelle.
2. **Ausgabe/Felder (SELECT):** Anschließend wird festgelegt, welche Felder der Basistabelle in der Ergebnisliste erscheinen sollen. * gibt alle Felder aus, alternativ können einzelne Felder angeklickt werden.

3. **Bedingungen (WHERE):** Bedingungen schränken die Ergebnisliste ein. Eine Bedingung besteht aus einem Feld, einem Vergleich (z.B. „gleich“, „größer als“, „enthält“) und einem Wert.
4. **Verknüpfte Tabellen (JOIN):** Sollen zusätzlich Daten aus einer verwandten Tabelle einbezogen oder für eine Bedingung genutzt werden (z.B. die Qualifikationen eines Mitglieds), wird über „Relation auswählen“ die passende Verknüpfung gewählt. Eine verknüpfte Tabelle verhält sich danach wie eine eigene kleine Abfrage: Auch für sie können wieder Felder, Bedingungen und weitere Verknüpfungen festgelegt werden.
5. **Sortierung (SORT):** Zuletzt kann festgelegt werden, nach welchen Feldern die Ergebnisliste sortiert wird (auf- oder absteigend).

### 4.8.2 Bedingungen im Detail

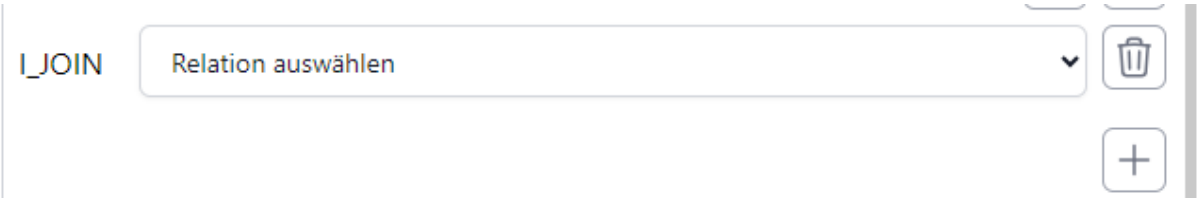



Jede Bedingung besteht aus **Feld**, **Vergleich** und **Wert**. Beim Vergleich „zwischen“ werden ein Von- und ein Bis-Wert angegeben, bei „ist leer“/„ist nicht leer“ entfällt der Wert.
Ab der zweiten Bedingung wird zusätzlich festgelegt, wie sie mit den vorherigen verknüpft wird: **UND** (alle verknüpften Bedingungen müssen zutreffen) oder **ODER** (mindestens eine muss zutreffen).
Über das zweite Icon kann eine **Gruppe** aus mehreren Bedingungen erstellt werden - vergleichbar mit Klammern in einer Rechnung. Damit lassen sich auch verschachtelte Kombinationen abbilden, z.B. „(Status gleich aktiv ODER Status gleich fördernd) UND Eintrittsdatum kleiner 01.01.2020“.
Folgende Vergleiche stehen zur Auswahl:

| **Vergleich** | **Bedeutung** |
|---|---|
| gleich / ungleich | Feld entspricht bzw. entspricht nicht genau dem Wert |
| größer / größer gleich | Feld ist größer bzw. größer oder gleich dem Wert (Zahlen, Daten) |
| kleiner / kleiner gleich | Feld ist kleiner bzw. kleiner oder gleich dem Wert (Zahlen, Daten) |
| enthält / enthält nicht | Text im Feld enthält bzw. enthält nicht den angegebenen Wert |
| beginnt mit / endet mit | Text im Feld beginnt bzw. endet mit dem angegebenen Wert |
| ist leer / ist nicht leer | Feld hat bzw. hat keinen Wert - hier wird kein Wert angegeben |

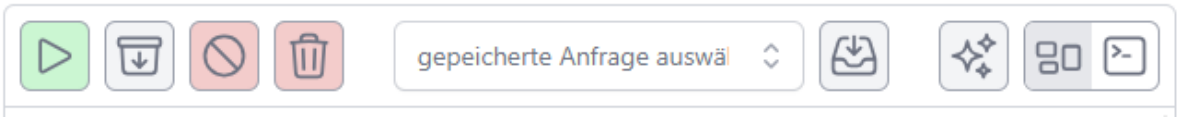
| Vergleich | Bedeutung |
|---|---|
| zwischen | Feld liegt zwischen einem Von- und einem Bis-Wert (Zahlen, Daten) |
| Zeitraum-Gleichheit | Vergleicht nur den Jahresanteil eines Datumsfelds, z.B. um alle Geburtstage in einem bestimmten Jahr zu finden, unabhängig vom Tag |

### 4.8.3 Verknüpfte Tabellen im Detail




Über „Relation auswählen“ werden nur Verknüpfungen angeboten, die für die aktuelle Tabelle sinnvoll sind (z.B. von Mitgliedern zu deren Qualifikationen oder Kommunikationskanälen). Die technische Verbindung zwischen den Tabellen wird dabei automatisch hergestellt - es muss keine eigene Verbindungsbedingung formuliert werden.
**Wichtig:** Eine Verknüpfung gibt nur Datensätze der Basistabelle aus, zu denen mindestens ein passender Eintrag in der verknüpften Tabelle existiert (INNER JOIN). Wird z.B. nach Mitgliedern mit einer bestimmten Qualifikation gesucht, schließt allein die Verknüpfung zu den Qualifikationen automatisch alle Mitglieder ohne jegliche Qualifikation aus - unabhängig von einer zusätzlichen Bedingung.

### 4.8.4 Datenstruktur



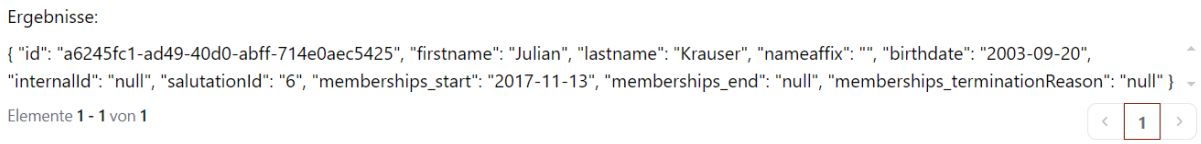

Die Symbole der Werkzeugleiste des Builders ermöglichen die Ausführung einer Abfrage, aber auch den Export oder die Speicherung einer Abfrage.

Der Export ermöglicht das Speichern der Abfrageergebnisse in einer CSV-Datei. Weiterhin können entweder die Abfrageergebnisse geleert oder die Abfrage und dessen Ergebnisse zurückgesetzt werden. Auch kann eine ausgewählte Abfrage überschrieben werden.

Über das Sternchen-Symbol in der Werkzeugleiste kann eine Übersicht aller Tabellen und ihrer

Verknüpfungen angezeigt werden. Sie hilft dabei, die passende Basistabelle sowie mögliche Verknüpfungen zu finden, ohne die Datenstruktur auswendig kennen zu müssen. Die Übersicht kann auch als Bild heruntergeladen werden.

Nach dem Ausführen (Play-Symbol) zeigt die **Ausgabe** die Ergebnisliste sowie die Anzahl der gefundenen Treffer an.




## 4.8.5 Beispiel

Gesucht werden alle aktiven Mitglieder mit der Qualifikation „Atemschutzgeräteträger“:

1. **Basistabelle:** Mitglieder auswählen.
2. **Verknüpfte Tabelle:** Relation zu den Qualifikationen der Mitglieder auswählen.
3. **Bedingung in der verknüpften Tabelle:** Feld „Qualifikation“ gleich „Atemschutzgeräteträger“.
4. **Bedingung in der Basistabelle:** Feld „Status“ gleich „aktiv“.
5. Ausführen - die Ergebnisliste zeigt nun alle passenden Mitglieder.

Weitere typische Anwendungsfälle wären z.B. Mitglieder unter bzw. über 18 Jahren oder Mitglieder mit einer bestimmten Mitgliedschaftsdauer. Über die angezeigte Trefferanzahl lassen sich so auch einfache Statistiken erstellen, ganz ohne die Ergebnisliste selbst auszuwerten.
Die Queries können gespeichert werden, damit diese dann zum Beispiel im Newsletter (Abschnitt 4.4) verwendet werden können.

**Hinweis:** Der Query-Builder beinhaltet 2 Key-Words, welche innerhalb des SQL-Editors verwendet werden können und einen gesamten Query ersetzen:

- `member`: Ausgabe aller Mitglieder
- `memberByRunningMembership`: Ausgabe aller Mitglieder mit aktuell laufendem Mitgliedschafts-Eintrag

## 4.9 Templates & Template Builder

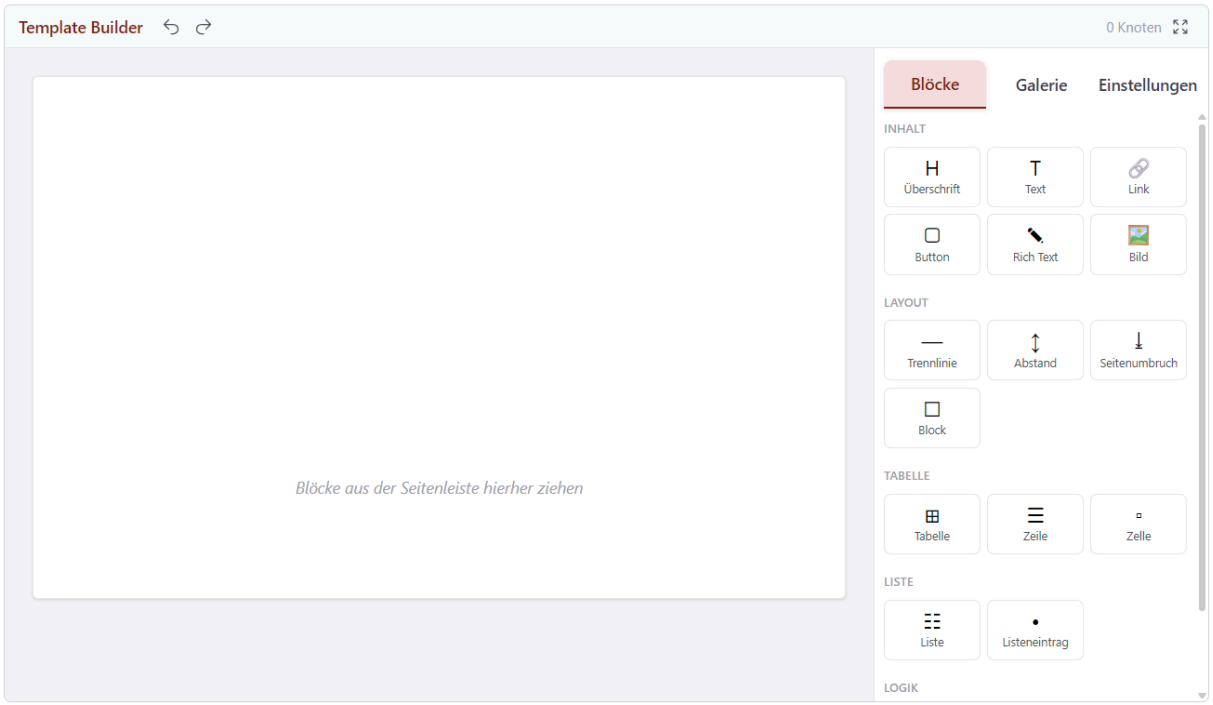
Templates sind Formatvorlagen, die beim Druck von Mitgliederdaten bzw. -listen, Protokollen (Abschnitt 4.3) und Newslettern (Abschnitt 4.4) verwendet werden können. Diese Templates können frei im Template Builder gestaltet werden und beinhalten Datenfelder, die dann beim Druck automatisch ausgefüllt werden.

**Ablauf von der Erstellung bis zur Verwendung:**

1. Im Template Builder wird ein neues Template angelegt und mit Bausteinen (Abschnitt 4.9) gestaltet - z.B. Text, Bild oder Tabelle.
2. In die Bausteine werden bei Bedarf Variablen eingefügt (Abschnitt 4.9), die beim Druck automatisch mit den echten Daten des jeweiligen Datensatzes gefüllt werden.
3. Über die Vorschau im Templateeditor lässt sich das Ergebnis mit Platzhaltern prüfen, ohne dass ein echter Druckvorgang notwendig ist.
4. In der Template-Verwendung (siehe unten) wird das fertige Template dann für einen konkreten Druckvorgang (z.B. Mitgliederliste, Protokoll-Ausdruck, Newsletter-Mail) als Kopf-, Fuß- oder Hauptbereich hinterlegt.

Ein Template muss dabei nicht für jeden Anwendungsfall neu erstellt werden: Bereits bestehende Templates können dupliziert und angepasst werden, um Varianten für ähnliche Druckvorgänge schnell zu erstellen.

### 4.9.1 Template Builder




Der Template Builder besteht aus drei Bereichen:

- **Canvas:** Der zentrale Bereich, in dem das Template zusammengebaut wird. Elemente können per Drag & Drop aus der Seitenleiste gezogen und frei angeordnet werden.
- **Blöcke-Palette:** Die Seitenleiste enthält alle verfügbaren Bausteine, sortiert nach Kategorien. Blöcke werden per Drag & Drop auf den Canvas gezogen.
- **Einstellungen:** Wird ein Element auf dem Canvas ausgewählt, erscheinen in der Seitenleiste die Eigenschaften dieses Elements zur Bearbeitung.

In der Toolbar befinden sich die Rückgängig- und Wiederholen-Buttons (Strg+Z / Strg+Y) sowie ein Vollbild-Modus.

### 4.9.2 Bausteine

Die Bausteine sind in fünf Kategorien unterteilt:

**Inhalt**

- **Überschrift:** Überschrift mit wählbarer Ebene (H1–H3), Farbe, Schriftgröße und Ausrichtung.
- **Text:** Textblock mit Schriftstil (Fett, Kursiv, Unterstrichen, Durchgestrichen), Farbe, Schriftgröße, Schriftart (Helvetica, Times, Courier), Ausrichtung, Einzug, Zeilenhöhe und optionaler Mehrspalten-Darstellung (bis zu 4 Spalten). Variablen können mit `{{variable}}` eingebettet werden, Links mit `[Text](URL)`.
- **Link:** Verlinkter Text mit URL, Farbe und Schriftstil.
- **Button:** Button mit Hintergrundfarbe, Textfarbe, Ausrichtung, Padding, Breite und Radius.
- **Rich Text:** Einbettung eines Quill-Delta-Feldes aus den Daten. Zur dynamischen Darstellung von formatierten Textinhalten.
- **Bild:** Bild aus drei möglichen Quellen: Galerie (hochgeladene Bilder), App-Logo (Vereins-Logo) oder dynamischer Pfad/Variable (z.B. `files/inspection/{{id}}/photo.webp`). Konfigurierbar mit Breite, Höhe, Fit-Modus (contain/cover/stretch) und Ausrichtung.

**Layout**

- **Trennlinie:** Horizontale Linie mit einstellbarer Farbe, Stärke und Abständen oben/unten.
- **Abstand:** Vertikaler Leerraum mit einstellbarer Höhe.
- **Seitenumbruch:** Erzwingt einen Seitenumbruch an dieser Stelle.
- **Block:** Container-Element mit Hintergrundfarbe, Rahmen (Breite, Farbe, Radius), Padding und Margin. Kann beliebige Kinder-Elemente enthalten.

**Tabelle**

- **Tabelle:** Container für Zeilen. Einstellbar sind Rahmenart (Solid/Keine), Rahmenbreite, Rahmenfarbe und die Spaltenbreiten. Spaltenbreiten werden als Liste definiert – `*` steht für flexible Breite, ansonsten können feste Werte in `%`, `px`, `pt`, `mm` oder `cm` angegeben werden.
- **Zeile:** Enthält die Zellen einer Tabellenzeile. Kann als Kopfzeile markiert werden. Hintergrundfarbe, Textfarbe und Schriftstil sind konfigurierbar.
- **Zelle:** Einzelne Tabellenzelle. Kann entweder einen Textinhalt oder beliebige Kinder-Elemente enthalten. Konfigurierbar mit ColSpan, Ausrichtung, Hintergrundfarbe, Textfarbe, Schriftgröße, Schriftstil und Padding.

**Liste**

- **Liste:** Container für Listeneinträge. Wahlweise als Aufzählung (Punkte) oder nummerierte Liste. Einstellbar sind Einzug und Punktgröße.
- **Listeneintrag:** Einzelner Eintrag in einer Liste. Kann einen Textinhalt oder beliebige Kinder-Elemente enthalten.

**Logik**

- **Schleife:** Iteriert über ein Datenfeld (z.B. `items`, `inspections`) und rendert die enthaltenen Kinder für jeden Eintrag. Das Datenfeld und ein Alias (Standard: `item`) werden konfiguriert. Über den Alias kann in Kind-Elementen auf die Daten des aktuellen Eintrags zugegriffen werden (z.B. `{{item.name}}`).

### 4.9.3 Verschachtelungsregeln

Nicht jeder Baustein kann überall platziert werden. Der Builder validiert beim Drag & Drop automatisch, welche Elemente wo erlaubt sind:

- **Tabelle** erlaubt nur Zeilen und Schleifen als direkte Kinder.
- **Zeile** erlaubt nur Zellen als direkte Kinder.
- **Liste** erlaubt nur Listeneinträge und Schleifen als direkte Kinder.
- **Block, Zelle, Listeneintrag** erlauben alle Inhalts- und Layout-Elemente.
- **Schleife** ist kontextabhängig: Befindet sich die Schleife innerhalb einer Tabelle, dürfen nur Zeilen eingefügt werden. Innerhalb einer Liste nur Listeneinträge. Ansonsten sind alle Inhalts- und Layout-Elemente erlaubt. Diese Regel gilt auch für mehrfach verschachtelte Schleifen.

### 4.9.4 Variablen & Pipes

In Textbausteinen können dynamische Datenfelder über die Syntax `{{feldname}}` eingebettet werden. Verschachtelte Felder werden mit Punkt-Notation angesprochen: `{{member.firstname}}`. Variablen können mit **Pipes** (Filtern) weiterverarbeitet werden. Ein Pipe wird mit `|` angehängt, optionale Argumente mit `:` getrennt:
`{{feldname | pipe}}` bzw. `{{feldname | pipe:argument}}`
Mehrere Pipes können verkettet werden: `{{feldname | pipe1 | pipe2:arg}}`

**Verfügbare Pipes:**

| Pipe | Beschreibung | Argument |
|---|---|---|
| `date` | Datum: TT.MM.JJJJ | – |
| `longdate` | Datum: Wochentag TT.MM.JJJJ | – |
| `datetime` | Datum + Uhrzeit: TT.MM.JJJJ HH:MM | – |
| `longdatetime` | Datum + Uhrzeit: TT.Monat.JJJJ HH:MM | – |
| `longdatetimeWithWeekday` | Wochentag TT.Monat.JJJJ HH:MM | – |
| `weekdayDayMonth` | Wochentag TT.Monat | – |
| `showDashAsFallback` | Zeigt „—“ wenn Wert leer ist | – |
| `fallback` | Zeigt Ersatztext wenn Wert leer ist | Ersatztext |
| `prepend` | Stellt Text voran (nur wenn Wert vorhanden) | Präfix |

| Pipe | Beschreibung | Argument |
|---|---|---|
| `append` | Hängt Text an (nur wenn Wert vorhanden) | Suffix |
| `ifTruthy` | Gibt Argument aus wenn Wert wahr ist | Ausgabetext |
| `joinNames` | Verbindet Array aus Objekten mit `firstname`/`lastname` zu kommaseparierter Liste | – |
| `length` | Gibt die Länge eines Arrays oder Strings aus | – |
| `json` | Gibt den Wert als JSON-String aus | – |

**Hinweise zu speziellen Pipes:**

- `joinNames`**:** Erwartet ein Array von Objekten mit den Keys `firstname` und `lastname`. Jedes Objekt wird zu „Vorname Nachname“ zusammengefügt (fehlende Keys werden übersprungen). Die Einträge werden mit „, „ verbunden.
  Beispiel: `{{attendees | joinNames}}` → „Max Mustermann, Anna Schmidt“
- `ifTruthy`**:** Gibt das Argument nur aus, wenn der Wert wahr ist (nicht `null`, `undefined`, `""`, `0` oder `false`). Nützlich für bedingte Texte inline.
  Beispiel: `{{isPdf | ifTruthy:Dieses Dokument wurde gedruckt.}}`
- `prepend` **/** `append`**:** Gibt Präfix bzw. Suffix nur aus, wenn der eigentliche Wert vorhanden ist. Verhindert überflüssige Leerzeichen oder Trennzeichen bei leeren Feldern.
  Beispiel: `{{member.nameaffix | prepend: }}` → „ von“ nur wenn Namenszusatz existiert
- `fallback`**:** Zeigt einen Ersatztext, wenn der Wert leer oder nicht vorhanden ist.
  Beispiel: `{{member.phone | fallback:nicht angegeben}}`
- **Datums-Pipes:** Erwarten einen gültigen Datumswert (ISO-String oder Date-Objekt). Formatierung erfolgt in deutscher Lokalisierung.

**Beispiele:**

- `{{member.birthDate | date}}` → „15.03.1990“
- `{{member.nameaffix | prepend: }}` → „ von“ (nur wenn vorhanden)
- `{{items | length}}` → „5“
- `{{isPdf | ifTruthy:Umstieg auf Mail}}` → „Umstieg auf Mail“ (nur im PDF)
- `{{member.phone | fallback:nicht angegeben}}` → „nicht angegeben“ (wenn leer)

### 4.9.5 Bedingungen

Jedem Baustein kann eine oder mehrere Bedingungen zugewiesen werden. Bedingungen steuern, ob ein Element beim Druck angezeigt wird oder nicht. Jede Bedingung besteht aus:

- **Feld:** Der Name des Datenfeldes (z.B. `status`, `check.type`).
- **Operator:** Vergleichsoperator – gleich, ungleich, wahr, falsch, größer, kleiner, größer-gleich, kleiner-gleich.
- **Wert:** Der Vergleichswert (entfällt bei wahr/falsch).

Mehrere Bedingungen werden als UND-Verknüpfung ausgewertet – alle müssen erfüllt sein, damit das Element angezeigt wird.

### 4.9.6 Render-Kontext

Beim Rendern eines Templates werden automatisch Kontext-Variablen gesetzt, die den Ausgabekanal kennzeichnen:

- `isPdf` – `true` wenn das Template als PDF gedruckt wird
- `isMail` – `true` wenn das Template als E-Mail (HTML) versendet wird

Diese Variablen stehen nicht explizit in den Daten, sondern werden automatisch je nach Ausgabekanal gesetzt. Ist eine Variable im aktuellen Kontext nicht vorhanden, wird sie als `false` gewertet.

**Verwendung in Bedingungen:**

Ein Element kann z.B. nur im PDF angezeigt werden, indem eine Bedingung mit Feld `isPdf` und Operator `wahr` gesetzt wird. Umgekehrt kann mit Feld `isMail` / Operator `wahr` ein Element nur in der Mail angezeigt werden.

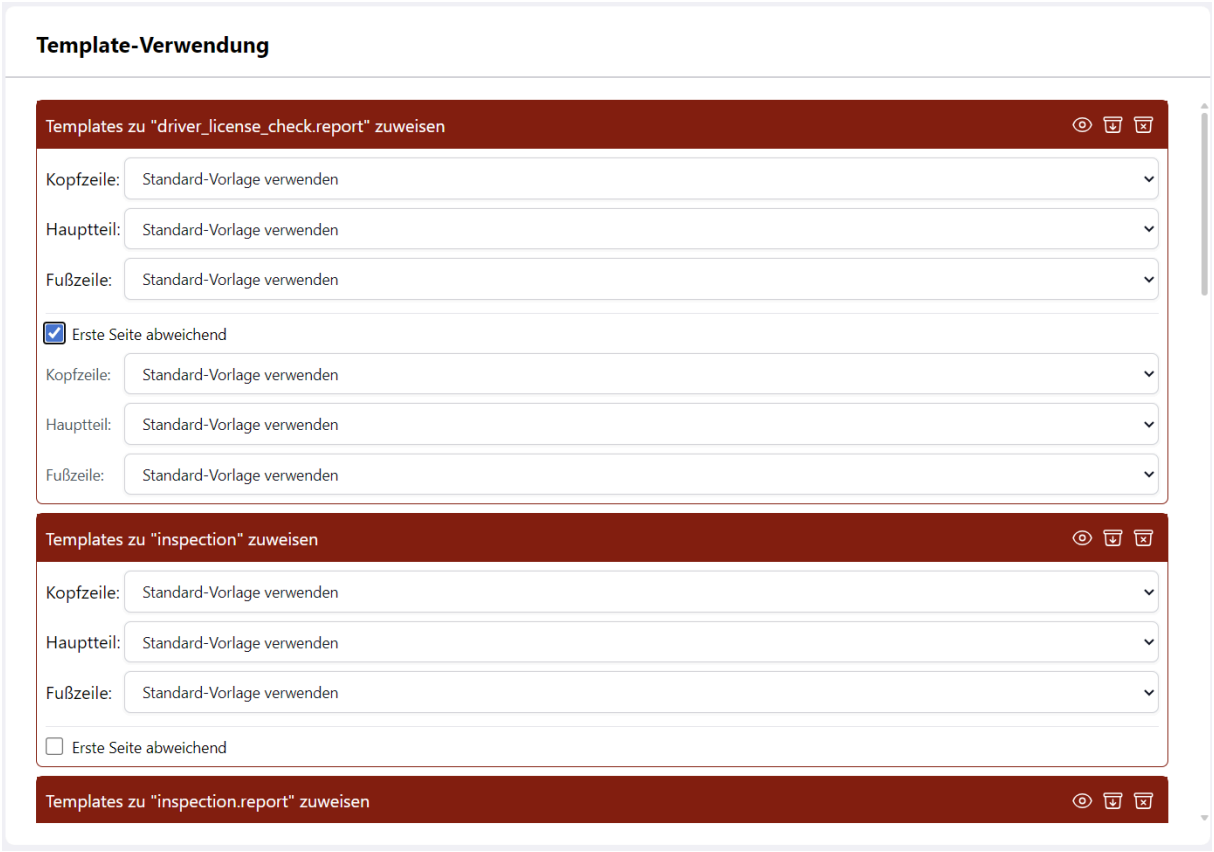
**Verwendung als Pipe in Texten:**

In Textbausteinen kann die Variable auch als Pipe genutzt werden: `{{isPdf | ifTruthy:Nur im PDF sichtbar}}`. Ist `isPdf` wahr, wird der angegebene Text ausgegeben – ansonsten bleibt die Stelle leer.

### 4.9.7 Bildergalerie

Über den Galerie-Tab in der Seitenleiste können Bilder hochgeladen und verwaltet werden. Diese stehen dann als statische Bilder im Template zur Verfügung. Bilder können per Drag & Drop direkt aus der Galerie auf den Canvas gezogen werden. Alternativ kann ein Bild-Element ausgewählt und über den Zuweisen-Button in der Galerie ein Bild zugeordnet werden.

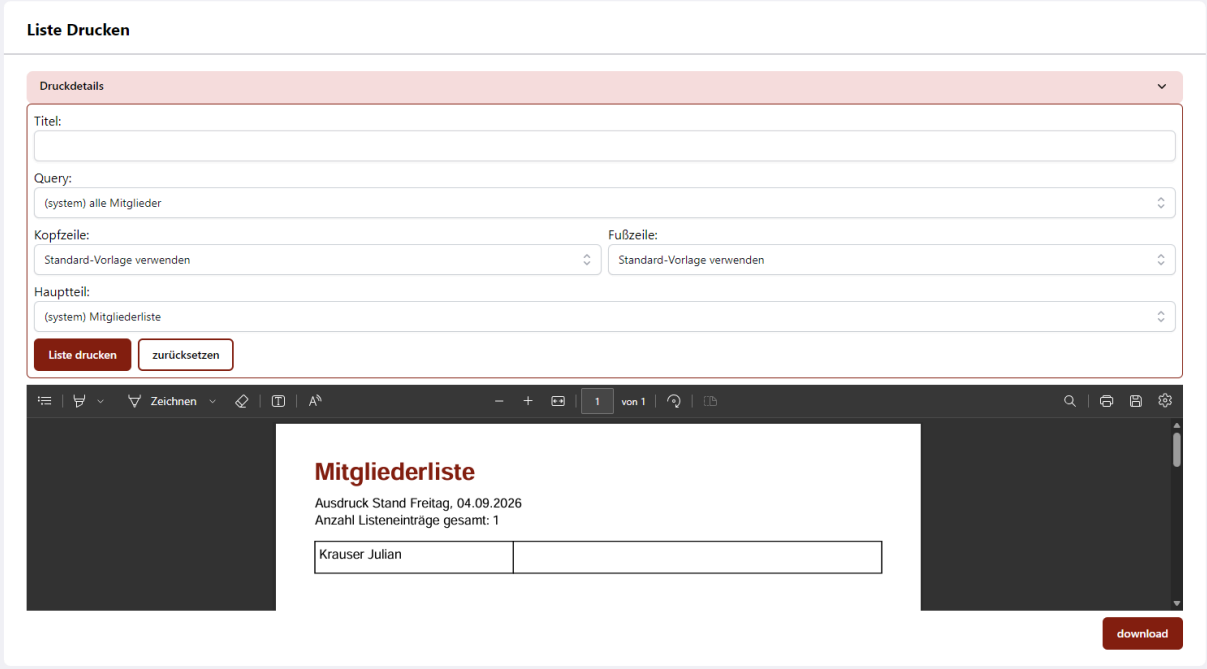
### 4.9.8 Template-Verwendung




Die Formatvorlagen können in der Template-Verwendung als Vorlagen für Kopf- und Fußzeile und den Hauptteil verwendet werden. Die Fußzeile gibt standardmäßig die Seitenzahl an. Eine abweichende erste Seite kann eingestellt werden.

Bei der Einstellung der Template-Verwendung kann für jeden verfügbaren Druckvorgang angegeben werden, welche eigenen bzw. Standard-Vorlagen verwendet werden sollen.
Die Änderungen können gespeichert oder vollständig zurückgesetzt werden.
Eine Vorschau des Resultats ist unter dem Auge-Symbol möglich. Dazu werden Demo-Daten geladen und in das Template eingefügt. Die Vorschau zeigt dann ein PDF, wie es bei einem finalen Export aussehen könnte.

## 4.10 Listen-Druck



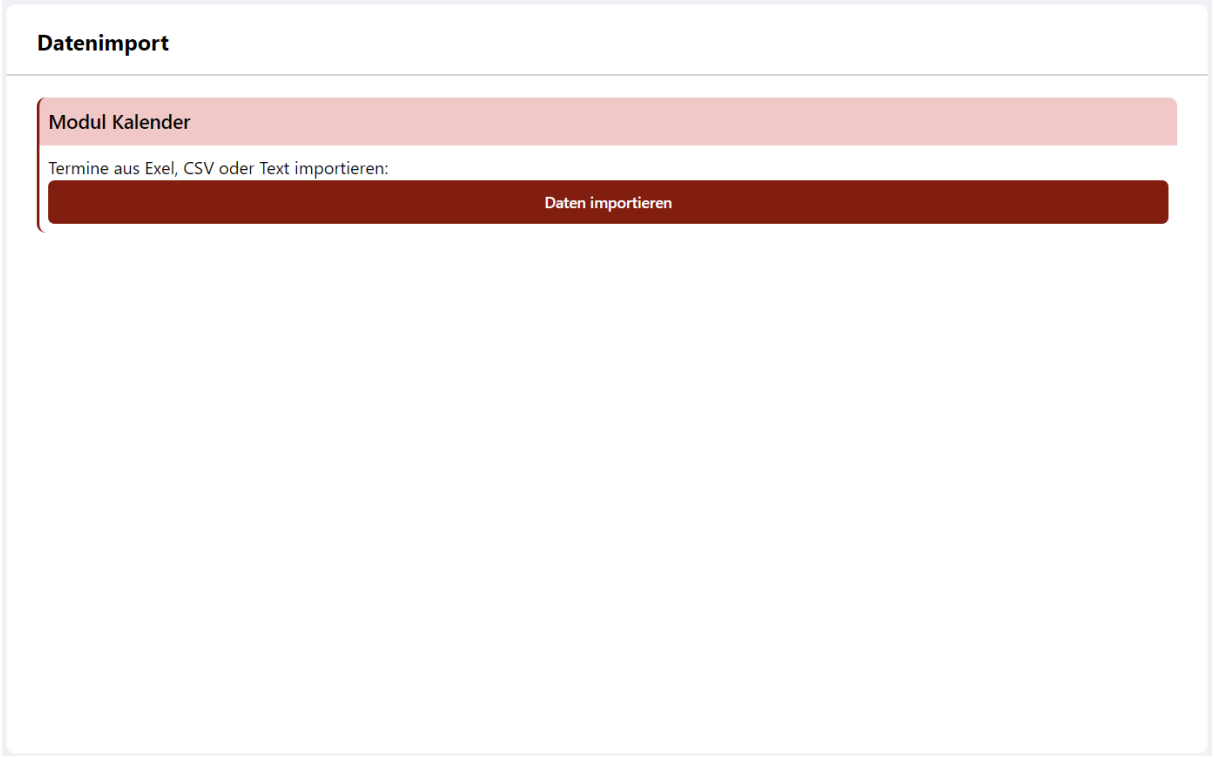

Das Modul Listen-Druck ermöglicht den pdf-Export von Daten, welche über den Query-Builder bezogen werden können. Dabei gibt es 2 System-Queries, welche alle Mitglieder und Mitglieder mit laufendem Mitgliedschafts-Eintrag ausgeben. Ausgewählt werden können Templates des Template-Builders und ein System-Template, welches eine Tabelle mit Vor- und Nachname der gewählten Mitglieder erstellt.

Das erstellte pdf wird nicht auf dem Server gespeichert und kann nur heruntergeladen werden.

Die Angabe eines Titels für den Ausdruck ist optional. Wird kein Titel angegeben, wird der Ausdruck ohne Titelzeile erstellt. Für eine höhere Darstellung des pdfs kann die Eingabe eingeklappt werden. Dies geschieht beim Absenden des Drucks automatisch.

## 4.11 Datenimport (Massendaten)

Das Modul Datenimport ermöglicht die schnelle Erfassung größerer Datenmengen. Es können entweder Daten aus Excel- oder CSV-Dateien importiert oder Einträge direkt in einer Tabelle erstellt und bearbeitet werden.



### 4.11.1 Import von Terminen

Termine können auf mehrere Arten importiert werden: aus Excel- oder CSV-Dateien, per Copy-Paste aus Excel oder durch manuelle Eingabe in einer Tabelle.




**Schritt 1: Übersichtsansicht**

Hier können neue Termine hinzufügt oder verwaltet werden. Es stehen folgende Icons zur Verfügung:

- „+“ Icon: Neue Spalte oder Zeile hinzufügen
- Datei-Icon: Daten aus Datei importieren
- Papierkorb-Icon: Alle Einträge löschen

Das Dropdown neben der Terminart hilft beim Ausfüllen: Ein auswählbarer Typ kann verwendet werden, um alle leeren Felder in dieser Spalte automatisch zu füllen. Ausgefüllte Felder werden dabei nicht überschieben.



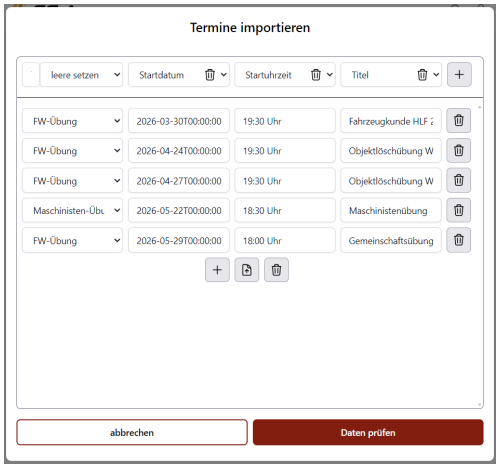

**Schritt 2: Dateiimport**

Importmöglichkeiten:

- Datei auswählen: CSV oder Excel-Dateien
- Copy-Paste: Auch können in Excel die gewünschten Zellen markiert und kopiert werden. Die kopierten Daten können in dem Textfeld eingefügt werden.

Die erkannten Zellen werden im unteren Textfeld angezeigt.

Über das Icon oben rechts kann zurück zur Übersichtsansicht gewechselt werden, ohne Daten zu übernehmen. Mit dem „Daten übernehmen“-Button werden die Daten in die Tabelle übernommen.

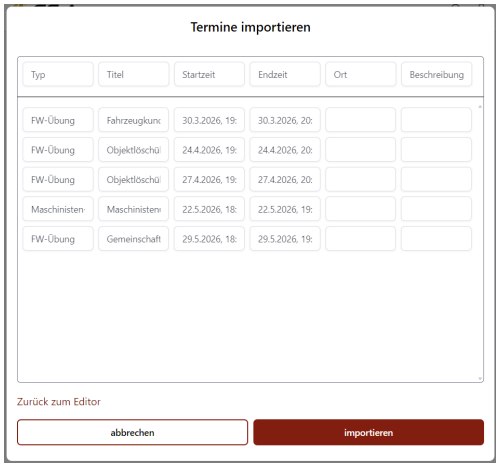



**Schritt 3: Zuordnung und Bearbeitung**

Die importierten Daten müssen noch zugeordnet werden:

- Spaltenzuordnung: Welche importierte Spalte gehört zu welchem Termin-Feld?
- Typzuordnung: Welchem Termintyp gehört welche Zeile an?

Die Tabelle zeigt alle Spalten und Zeilen. Es kann jedes Feld manuell bearbeitet und korrigiert werden.




**Schritt 4: Datenprüfung und Speicherung**

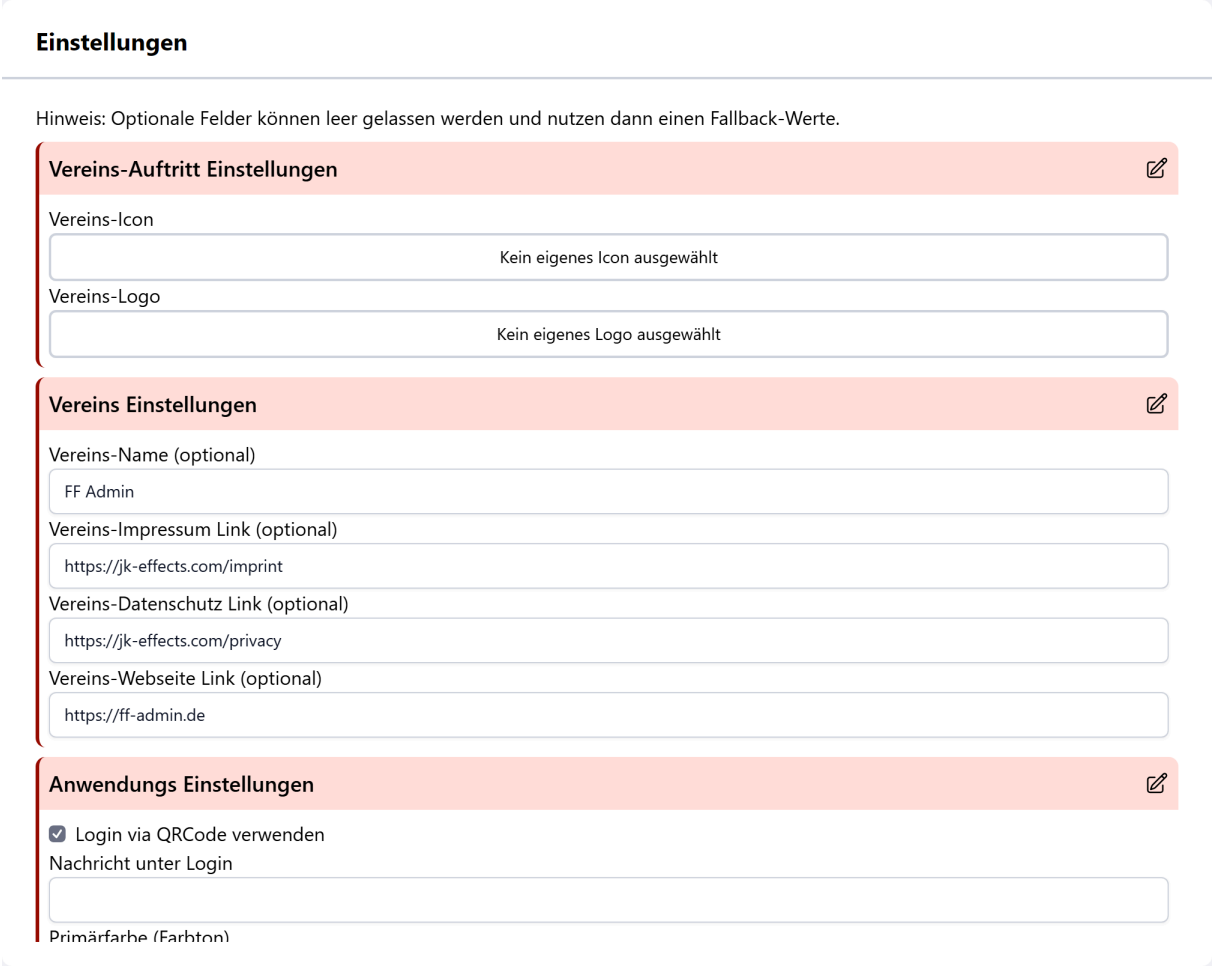
Über den Knopf „Daten prüfen“ müssen alle Eingaben verifiziert werden. Die erkannten Zeitdaten werden dann in einer nachfolgenden Ansicht zur Überprüfung angezeigt.

In dieser Übersicht werden alle Daten nochmal geprüft, bevor mit „importieren“ endgültig gespeichert wird.

Das System erkennt automatisch verschiedene Datums- und Zeitformate:

- Excel-Datums-Format, Unix Timestamp
- Deutsche Formate: 01.02.2026 oder 01.02.2026 14:30
- ISO-Formate: 2026-02-01, 2026-02-01T14:30:00Z oder 2025-12-05 14:30:00
- US-Formate: MM/DD/YYYY, M/D/YYYY, DD-Mon-YYYY oder Month DD, YYYY
- Zeit-Formate: 19:30, 19:30 Uhr oder 7:30 PM

## 4.12 Anwendungsspezifische Einstellungen



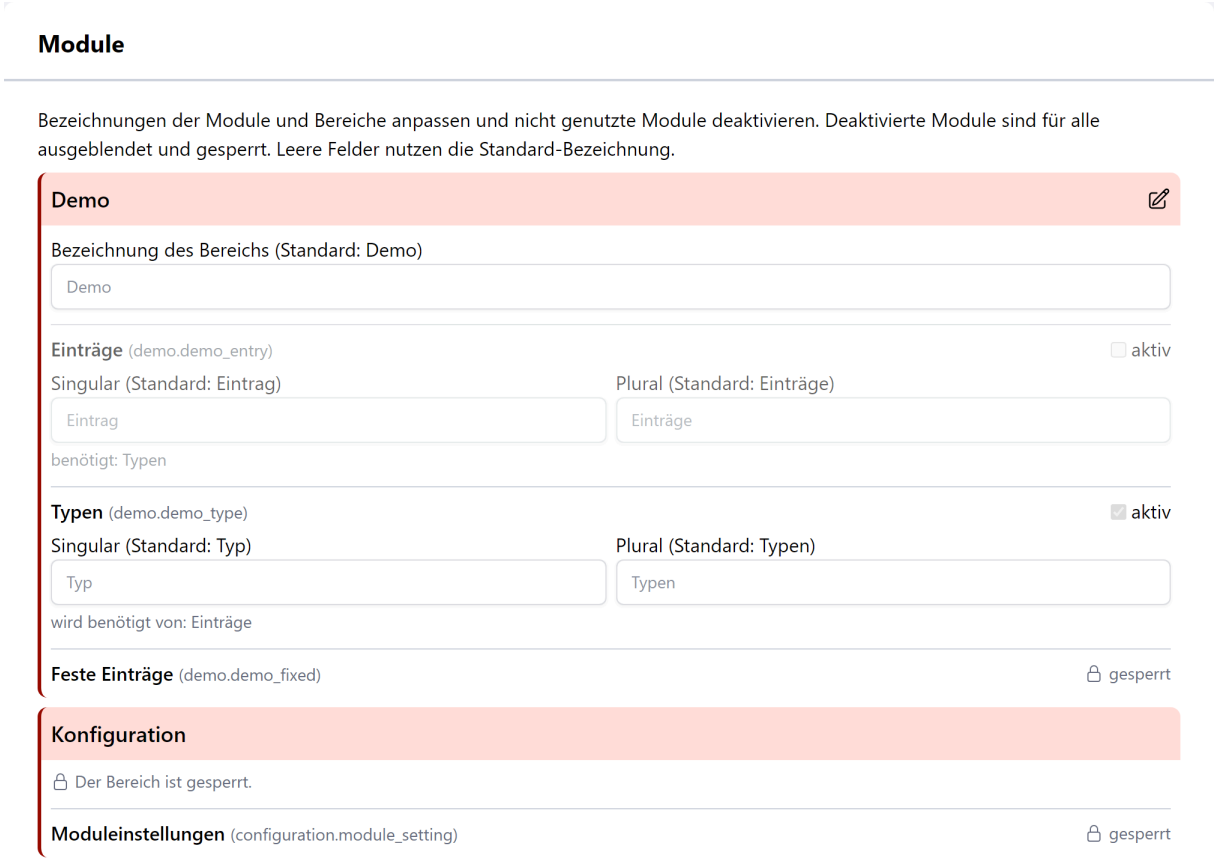

Das Modul Einstellungen ermöglicht eine flexible globale Konfiguration der Anwendung - ohne Neustart des Systems. Alle Anpassungen können direkt zur Laufzeit vorgenommen werden. Dadurch lassen sich beispielsweise Vereinslogos, Icons oder Texte im Login-Bereich individuell anpassen, ohne in den laufenden Betrieb einzugreifen.

Neben dem optischen Vereinsauftritt können auch grundlegende Informationen wie Links zum Impressum, zur Datenschutzerklärung oder zur Vereinswebseite hinterlegt werden. Diese Daten fließen dann automatisiert an verschiedenen Stellen im System ein.

Darüber hinaus ersetzt das Einstellungsmodul Teile der früheren, fest codierten Umgebungsvariablen (ENV). Werte wie etwa das Backup-Intervall oder E-Mail-Konten für den Versand können direkt in der Oberfläche geändert werden. Ein gespeichertes Passwort des E-Mail-Kontos wird dabei nicht angezeigt.

Insgesamt erlaubt das Modul eine Anpassung des Systems ohne Zugriff auf die Serverkonfiguration und Neustarts. Jedoch erfordern manche Änderungen das Neuladen der Webseite / WebApp.

## 4.13 Modulbezeichnungen




Unter `Verwaltung > Modulbezeichnungen` lassen sich die Bezeichnungen der Module und Bereiche an den eigenen Sprachgebrauch anpassen und nicht genutzte Module deaktivieren. Die Ansicht ist nach Bereichen gegliedert, jeder Bereich wird über das Stift-Symbol bearbeitet und gemeinsam gespeichert.

Für die Ansicht wird die Leseberechtigung des Moduls `Modulbezeichnungen` im Abschnitt Verwaltung benötigt, für Änderungen die Berechtigung zum Bearbeiten. Da ein deaktiviertes Modul für alle Nutzer gesperrt ist, handelt es sich um ein privilegiertes Modul: Die Berechtigung kann nur von Administratoren vergeben werden.

### 4.13.1 Bezeichnungen

Jedes Modul hat eine Standard-Bezeichnung im Singular und im Plural, jeder Bereich eine Standard-Bezeichnung. Sie stehen in Klammern über den Eingabefeldern. Ein ausgefülltes Feld ersetzt die Standard-Bezeichnung, ein leeres Feld nutzt wieder den Standard.

- **Plural:** Navigation, Seitentitel der Listen, Berechtigungen und Benachrichtigungseinstellungen, z.B. `Mitglieder`.
- **Singular:** Erstellen und Bearbeiten einzelner Einträge, z.B. `Mitglied erstellen`.

Die Bezeichnung wird ohne Artikel in die Texte eingesetzt. Bei einer Umbenennung von `Mitglied` in `Person` heißt die Schaltfläche danach `Person erstellen`.

Änderungen erscheinen sofort in allen geöffneten Sitzungen, ein Neuladen ist nicht nötig. Der Browser merkt sich die zuletzt geladenen Bezeichnungen, damit sie beim Start der Anwendung auch bei langsamer Verbindung direkt angezeigt werden.

### 4.13.2 Module deaktivieren

Über `aktiv` wird ein Modul ein- oder ausgeschaltet. Ein deaktiviertes Modul ist für alle Nutzer ausgeblendet, auch für Administratoren und den Besitzer:

- Es erscheint nicht mehr in der Navigation, in den Berechtigungen und in den Benachrichtigungseinstellungen.
- Aufrufe über einen gespeicherten Link führen zur Startseite zurück.
- Der Server lehnt alle Anfragen an das Modul ab, auch über WebApi-Token. Benachrichtigungen des Moduls werden nicht versendet.

Sind alle Module eines Bereichs deaktiviert, wird auch der Bereich ausgeblendet.

Die Daten und vergebenen Berechtigungen des Moduls bleiben erhalten. Nach dem erneuten Aktivieren steht das Modul wieder mit allen Einträgen zur Verfügung.

### 4.13.3 Abhängigkeiten

Manche Module setzen andere voraus, weil deren Einträge zum Erstellen benötigt werden, z.B. Mitglieder eine Anrede oder Fahrzeuge einen Fahrzeugtyp. Diese Beziehungen werden unter dem Modul angezeigt:

- **benötigt:** Module, die aktiv sein müssen, damit dieses Modul aktiviert werden kann.
- **wird benötigt von:** Module, die zuerst deaktiviert werden müssen, bevor dieses Modul deaktiviert werden kann.

Ist eine Änderung deshalb nicht möglich, ist der Schalter `aktiv` gesperrt, der Grund erscheint beim Darüberfahren mit der Maus. Module eines Bereichs, die voneinander abhängen, können in

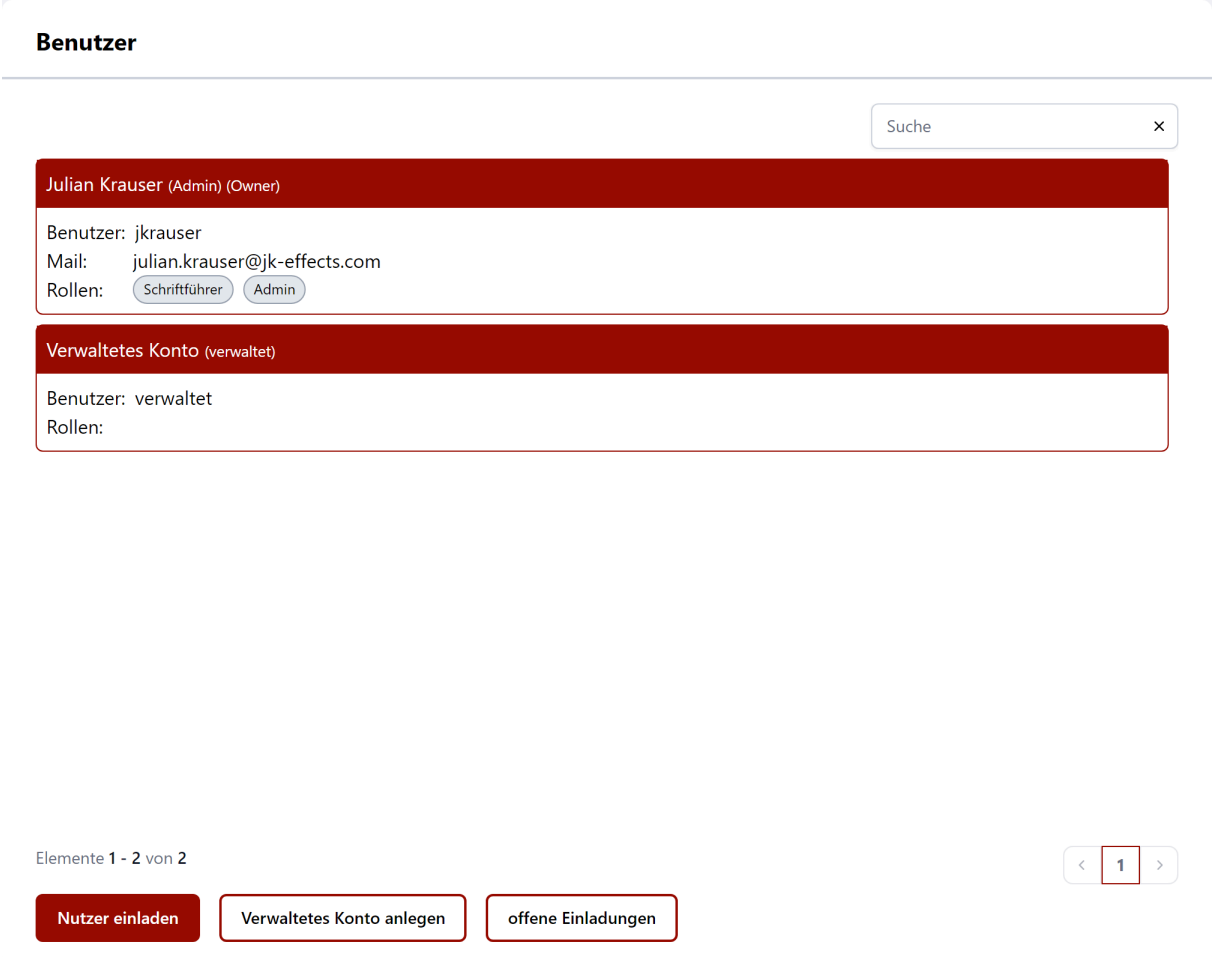
einem Schritt gemeinsam aktiviert oder deaktiviert werden. Liegt das benötigte Modul in einem anderen Bereich, muss zuerst dieser Bereich gespeichert werden.

Setzt ein aktives Modul ein deaktiviertes voraus, z.B. nach dem Einspielen eines Backups, meldet der Serverzustand (Abschnitt 2.6) einen Fehler.

### 4.13.4 Gesperrte Module

Mit einem Schloss gekennzeichnete Module und Bereiche können weder umbenannt noch deaktiviert werden. Dazu gehören die Bereiche Konfiguration und Verwaltung mit den Modulen dieser Anleitung, da sie für den Betrieb der Anwendung notwendig sind. Weitere Module können von der jeweiligen Anwendung gesperrt sein.

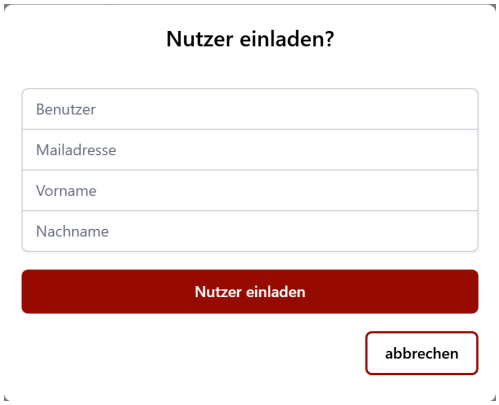
## 4.14 Benutzerverwaltung



Daten können nur von angemeldeten Benutzern geändert werden. Diese Benutzer melden sich mit einem TOTP oder Passwort an. Ein TOTP (Time-based One-time Password) ist ein zeitbasiertes Passwort, das sich jede Minute ändert. Nutzer können außerdem Passkeys (Abschnitt 5.5) für einen schnelleren Login erstellen. Jeder Benutzer hat Lese-, Bearbeitungs- und Löschrechte. Es können auch Rollen erstellt und Benutzern zugewiesen werden.

Benutzer mit dem Status `Owner` haben unabhängig der gesetzten Berechtigungen Vollzugriff auf die gesamte Anwendung.

### 4.14.1 Benutzer einladen




Benutzer mit eigener E-Mail-Adresse werden per Einladung hinzugefügt. Konten ohne E-Mail-Adresse lassen sich direkt anlegen (siehe „Verwaltete Konten“).
Einladungen sind nur solange gültig, bis sie angenommen oder zurückgezogen werden.
Bei einer Einladung wird ein personalisierter Link an die angegebene E-Mail-Adresse gesendet, über den sich der Benutzer registrieren kann. Nach der Registrierung hat der Benutzer noch keine Rollen oder Berechtigungen. Diese müssen nach der Registrierung zugewiesen werden.
Ein Benutzername oder eine E-Mail-Adresse darf nur einmal existieren.




Wenn ein Benutzer keine Berechtigungen hat oder diese ihm entzogen wurden, wird die Meldung auf der linken Seite angezeigt.
Nach der Registrierung wird diese Meldung standardmäßig angezeigt.
Mit einem Klick auf die Schaltfläche „Jetzt Berechtigungen neuladen“ werden die aktuellen Berechtigungen vom Server abgefragt. Bleibt die Ansicht unverändert, wurden dem Benutzer noch keine Berechtigungen zugewiesen. Andernfalls erfolgt eine Weiterleitung zur Admin-Oberfläche.

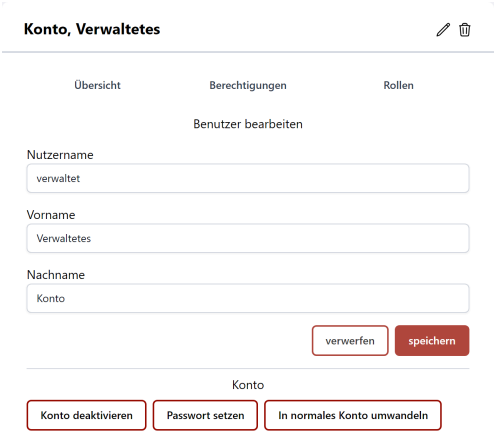
### 4.14.2 Verwaltete Konten

Verwaltete Konten haben keine E-Mail-Adresse. Sie sind für Personen gedacht, denen du kein Passwort weitergeben möchtest: Du legst das Konto unter `Verwaltung > Benutzer` mit Benutzername, Name und Passwort an, meldest es auf dem Gerät an und richtest dort einen Passkey ein. In der Benutzerliste ist es mit `(verwaltet)` gekennzeichnet.




- Ein Zurücksetzen per E-Mail und TOTP sind nicht möglich, Benachrichtigungen kommen nur in der App oder als Push.
- Benutzername und Passwort lassen sich mit dem aktuellen Passwort ändern. Passkeys lassen sich nur mit dem Passwort einrichten oder löschen.
- Wer Benutzer bearbeiten darf, kann das Passwort neu setzen. Dabei enden alle Sitzungen und alle Passkeys des Kontos werden gelöscht. Bei Konten mit Administratorrechten ist das nicht möglich, bei Konten mit privilegierten Modulen nur für Administratoren.
- Administratoren können ein verwaltetes Konto in ein normales umwandeln und umgekehrt, außer bei Konten mit Administratorrecht. In beiden Fällen enden alle Sitzungen und Passkeys des Kontos.
  - **In ein normales Konto:** Das Konto erhält eine E-Mail-Adresse, das bisherige Passwort bleibt gültig. An die Adresse geht ein Link, mit dem der Benutzer sein Passwort ersetzen kann.
  - **In ein verwaltetes Konto:** Die E-Mail-Adresse wird entfernt, die Anmeldung läuft danach nur noch über das dabei vergebene Passwort.

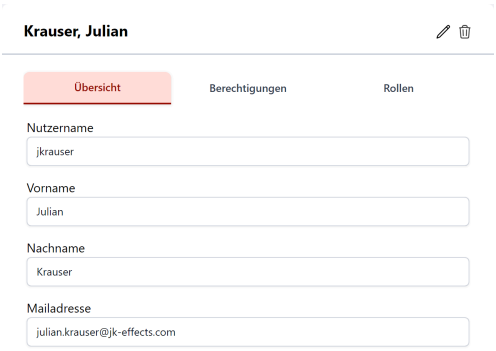
### 4.14.3 Konten deaktivieren




Wer Benutzer bearbeiten darf, kann ein Konto über das Stift-Symbol in der Detailansicht deaktivieren. Alle Sitzungen enden sofort, eine Anmeldung ist bis zur erneuten Aktivierung nicht mehr möglich. In der Benutzerliste ist das Konto mit `(deaktiviert)` gekennzeichnet. Das eigene Konto und das des Owners lassen sich nicht deaktivieren, Konten mit Administratorrechten nur durch Administratoren.

Im selben Bereich liegen auch das Setzen des Passworts und das Umwandeln verwalteter Konten.

### 4.14.4 Details

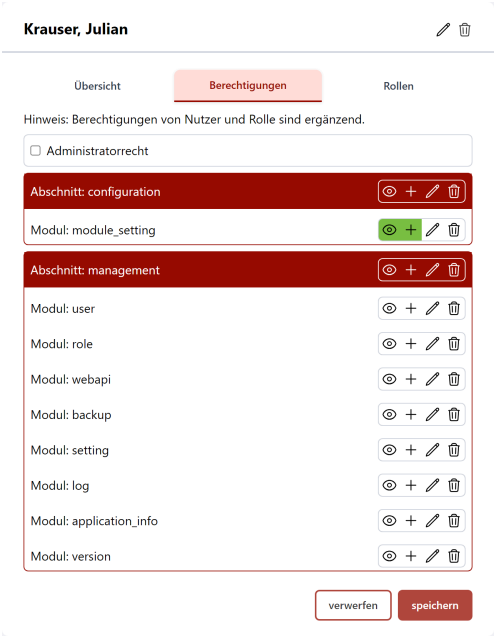



Jeder Benutzer verfügt über einen Benutzernamen, eine E-Mail-Adresse, Vor- und Nachname sowie zugewiesene Berechtigungen und Rollen.

In der Detailansicht können diese Daten eingesehen und bearbeitet werden. Benutzername und E-Mail-Adresse anderer Nutzer können dort nur Administratoren ändern. Das eigene Konto kann jeder Nutzer selbst ändern (siehe „Zugang und Anmeldedaten“).

**Wichtig:** Bei einer Änderung des Benutzernamens ist anschließend eine Anmeldung per Passwort oder TOTP nicht mehr möglich, sofern der neue Nutzernamen nicht weitergegeben wird. Der Zugang bleibt jedoch über Passkeys erhalten. Noch offene Links zum Zurücksetzen des Zugangs werden bei einer Änderung von Benutzername oder E-Mail-Adresse ungültig.

## 4.14.5 Berechtigungen




**Berechtigungen werden immer addiert.**
Das heißt: Wenn ein Benutzer direkt zugewiesene Berechtigungen und Rollen erhält, werden die Berechtigungen addiert und der Benutzer kann alles, was diesem direkt zugewiesen wurde und was die Rolle kann.
Das Berechtigungsmodell unterscheidet zwischen Lese-, Erstellungs-, Änderungs- und Löschberechtigungen. Die Berechtigungen sind abgestuft. Das heißt, wer anlegen darf, darf auch lesen. Wer bearbeiten darf, darf auch anlegen und dementsprechend auch lesen usw.
Die Berechtigungen sind entsprechend der Bereiche in der Navigationsleiste gruppiert. Wenn die Berechtigung für einen Abschnitts auf Lesen gesetzt ist, kann der Benutzer mit dieser Berechtigung Daten in allen Modulen des Abschnitts lesen.
Erweiterte Berechtigungen können pro Modul gesetzt werden. Dabei gelten die Berechtigungen des Abschnitts immer mehr als die eines Moduls innerhalb des Abschnitts.
Wer kein Administrator ist, sieht das Administratorrecht und privilegierte Module ausgegraut und kann sie nicht ändern.

## 4.14.6 Rollen

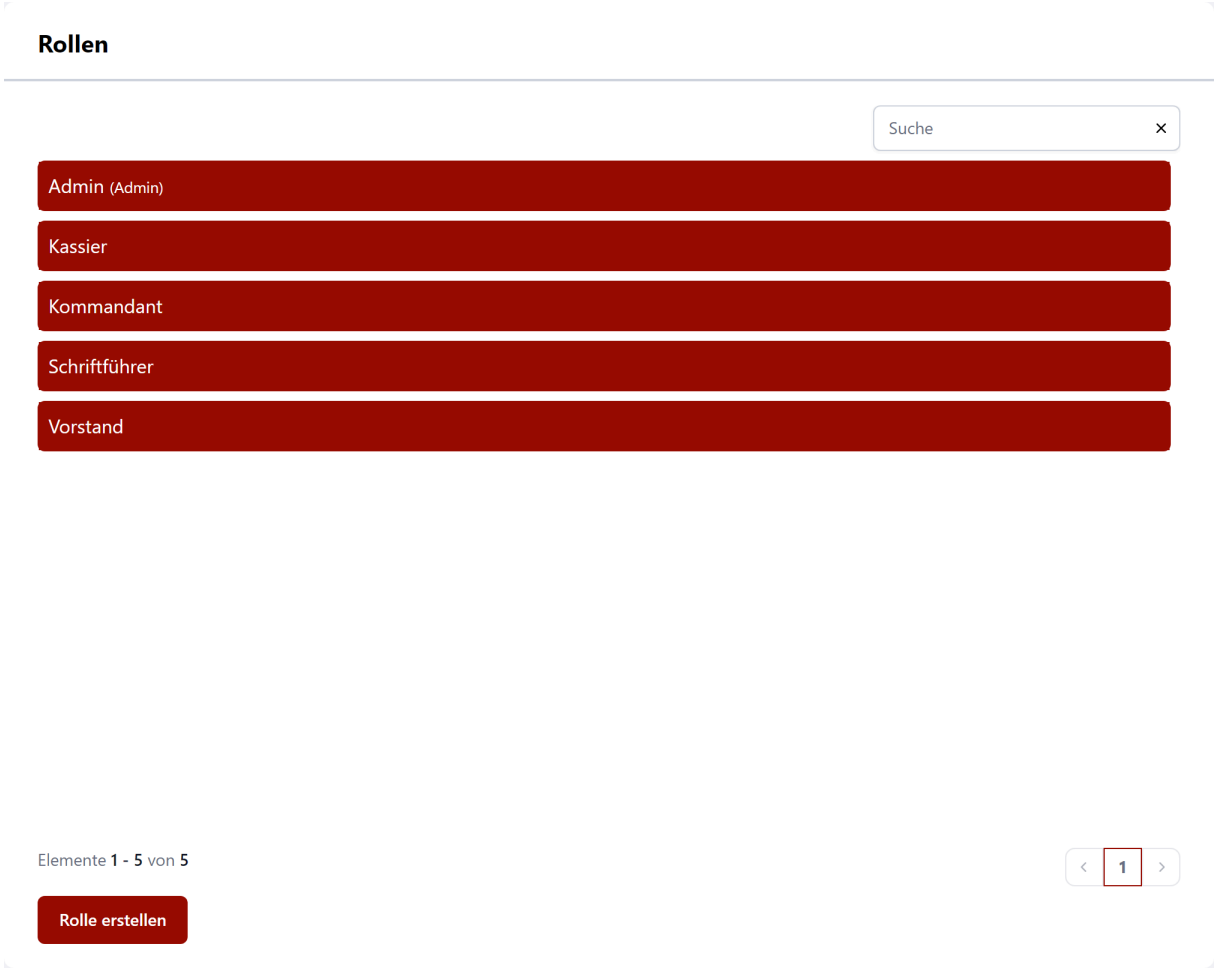



Rollen und Berechtigungen vergibt, wer Benutzer bzw. Rollen bearbeiten darf. Das Administratorrecht und privilegierte Module (zum Beispiel Backup, Webapi, Log und Einstellungen) vergeben nur Administratoren. Das gilt auch für Rollen, die solche Rechte enthalten: Nur Administratoren dürfen sie bearbeiten, löschen, zuweisen oder entziehen.
Rollen fassen Berechtigungen zusammen und ergänzen die einem Benutzer direkt zugewiesenen Berechtigungen.
Rollen können sich nicht gegenseitig ausschließen.
Einem Benutzer können mehrere Rollen zugewiesen werden. Muss aber keine Rollen haben.

**Hinweis:** Änderungen an Berechtigungen oder Rollen wirken sich unmittelbar auf die Benutzer

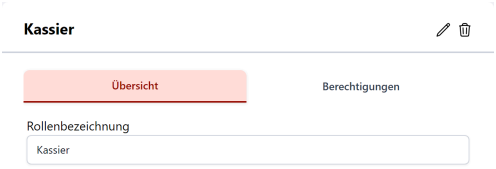
aus. Sie werden sofort an die Anwendung übertragen und dort angezeigt, sofern diese beim betroffenen Benutzer geöffnet ist. Ist der Nutzer offline, werden die Änderungen beim nächsten Öffnen der Anwendung übernommen.

## 4.15 Rollenverwaltung




Rollen dienen der Abstraktion von Berechtigungen. Dadurch können spezifische Berechtigungsschemata direkt einem oder mehreren Benutzern zugewiesen werden, ohne dass diese mehrfach manuell direkt zugewiesen werden müssen.

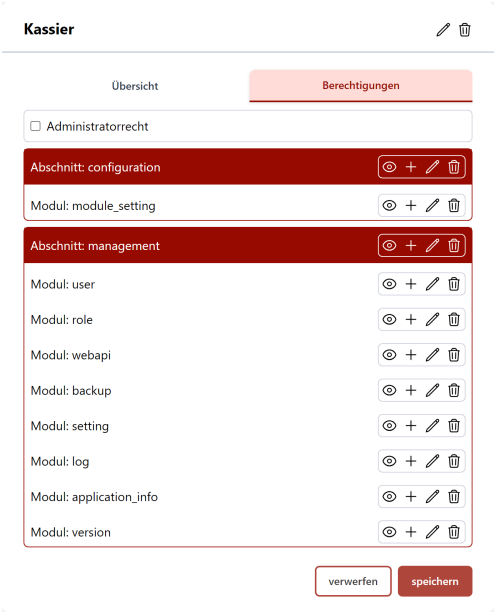
### 4.15.1 Details




Jede Rolle verfügt über einen Bezeichnung sowie zugewiesene Berechtigungen.
In der Detailansicht können diese Daten eingesehen und bearbeitet werden.
Rollen bearbeitet, wer das Recht zum Bearbeiten von Rollen hat. Rollen mit Administratorrecht oder privilegierten Modulen dürfen nur Administratoren bearbeiten oder löschen.

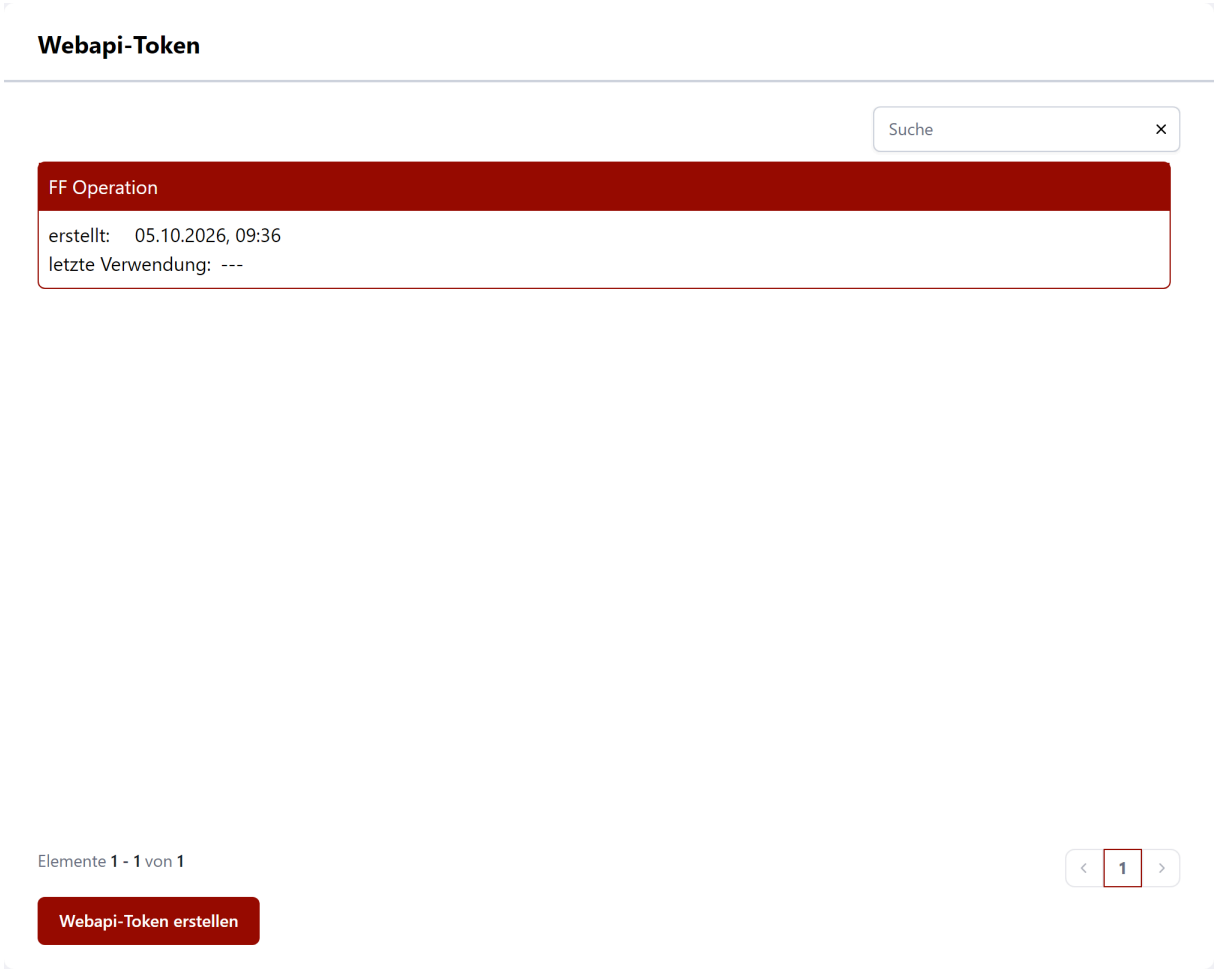
### 4.15.2 Berechtigungen

**Berechtigungen werden immer addiert.**
Das heißt: Wenn ein Benutzer mehrere Rollen zugewiesen bekommt, werden die Berechtigungen addiert und der Benutzer kann alles, was die Rollen ermöglichen. Zusätzlich werden auch die nutzerspezifischen Berechtigungen addiert.
Das Berechtigungsmodell unterscheidet zwischen Lese-, Erstellungs-, Änderungs- und Löschberechtigungen. Die Berechtigungen sind abgestuft. Das heißt, wer anlegen darf, darf auch lesen. Wer bearbeiten darf, darf auch anlegen und dementsprechend auch lesen usw.
Die Berechtigungen sind entsprechend der Bereiche in der Navigationsleiste gruppiert. Wenn die Berechtigung für einen Abschnitts auf Lesen gesetzt ist, kann der Benutzer mit dieser Berechtigung Daten in allen Modulen des Abschnitts lesen.

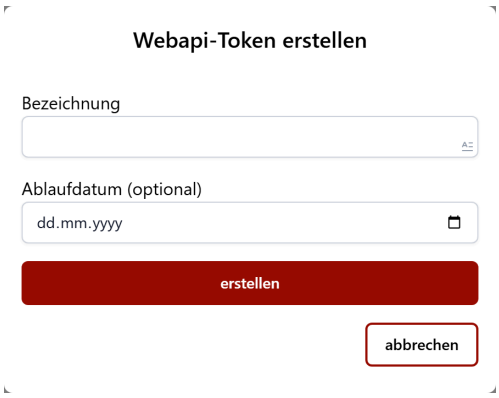



Erweiterte Berechtigungen können pro Modul gesetzt werden. Dabei gelten die Berechtigungen des Abschnitts immer mehr als die eines Moduls innerhalb des Abschnitts.
Manche Module bieten zusätzlich einzelne **weitere Berechtigungen** an, die über das Lese-/Erstellungs-/Änderungs-/Löschschema hinausgehen, z.B. um einzelnen Nutzern gezielt eine sonst nur Admins vorbehaltene Aktion zu erlauben oder um Werte wie Anzeigelimits einzuschränken. Diese werden am Ende des Berechtigungseditors gesondert aufgeführt und sind je nach Art entweder als Häkchen oder als Eingabewert gesetzt.

## 4.16 WebApi




WebApi-Tokens ermöglichen externen Clients oder anderen Anwendungen des FF Ökosystems oder eigenen Anwendungen die Kommunikation mit der Anwendung. Dabei haben WebApis Zugriff auf fast alle Daten der Anwendung, sofern die Berechtigungen gesetzt sind.
Kategorisch ausgeschlossen ist der Zugriff auf die Anwendungsmodule Backup, WebApi und Logs sowie Serverinformationen wie z.B. die Version. Das gilt auch für WebApis mit Administratorrecht. Diese Module werden im Berechtigungseditor der WebApi nicht angezeigt.
Auch Konten deaktivieren, verwaltete Konten anlegen und umwandeln sowie Passwörter setzen ist über eine WebApi nicht möglich. Abgerufene Benutzer können deaktiviert sein, verwaltete Konten haben keine E-Mail-Adresse.
WebApis mit Administratorrecht können nur Administratoren bearbeiten. Haben sie kein Ablaufdatum, weist der Serverzustand (Abschnitt 2.6) darauf hin.



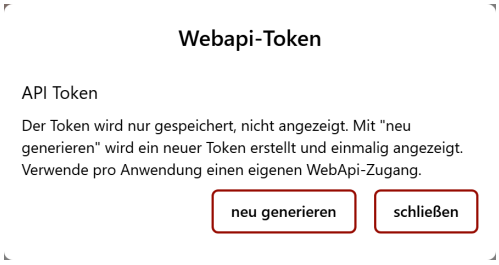

Eine WebApi kann mit einem Verfallsdatum erstellt werden. Dieses Ablaufdatum wird fest eingestellt, wenn eine WebApi Zugriff anfordert.
Zusätzlich wird für jede WebApi ein Token erstellt, das zur Authentifizierung der WebApi verwendet werden kann.

### 4.16.1 Details



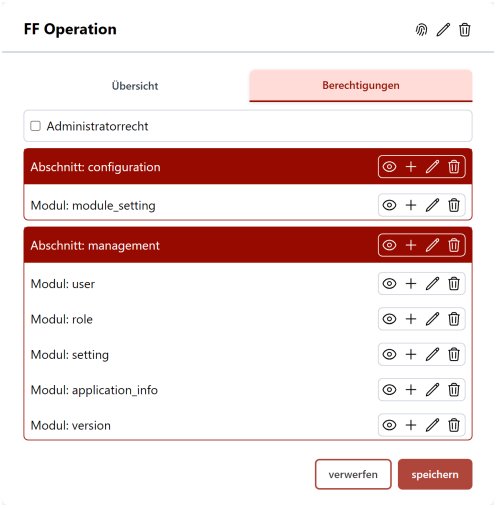

Jede WebApi verfügt über eine Bezeichnung und ein Verfallsdatum.
In der Detailansicht können diese Daten eingesehen und bearbeitet werden.
**Wichtig:**

Bei einer Änderung der Berechtigungen werden die Berechtigungen erst übernommen, wenn das Zugangs-Token (siehe unten Abschnitt 4.16.3) neu ausgestellt wird. Das Intervall, in dem das Token erneuert wird, ist die `JWT-Gültigkeitsdauer`, welche in den Einstellungen (Abschnitt 4.12) gesetzt werden kann.




Unter dem Fingerabdruck-Symbol kann das Token der WebApi erneuert werden.
Ein Token wird nur einmalig direkt nach dem Erzeugen angezeigt und muss dann kopiert werden. Danach lässt es sich nicht mehr abrufen. Geht es verloren, muss es mit `neu generieren` ersetzt werden. Das bisherige Token wird dabei sofort ungültig, alle Systeme, die es verwenden, müssen umgestellt werden. Pro Anwendung sollte deshalb ein eigener WebApi-Zugang verwendet werden.
Diese Token sind JWTS und können daher ohne Kenntnis der Serverkonfiguration nicht gefälscht werden.
Zur schnelleren Konfiguration in anderen Anwendungen zeigt der Dialog mit dem Token zusätzlich die Domain des Server der Anwendung an.

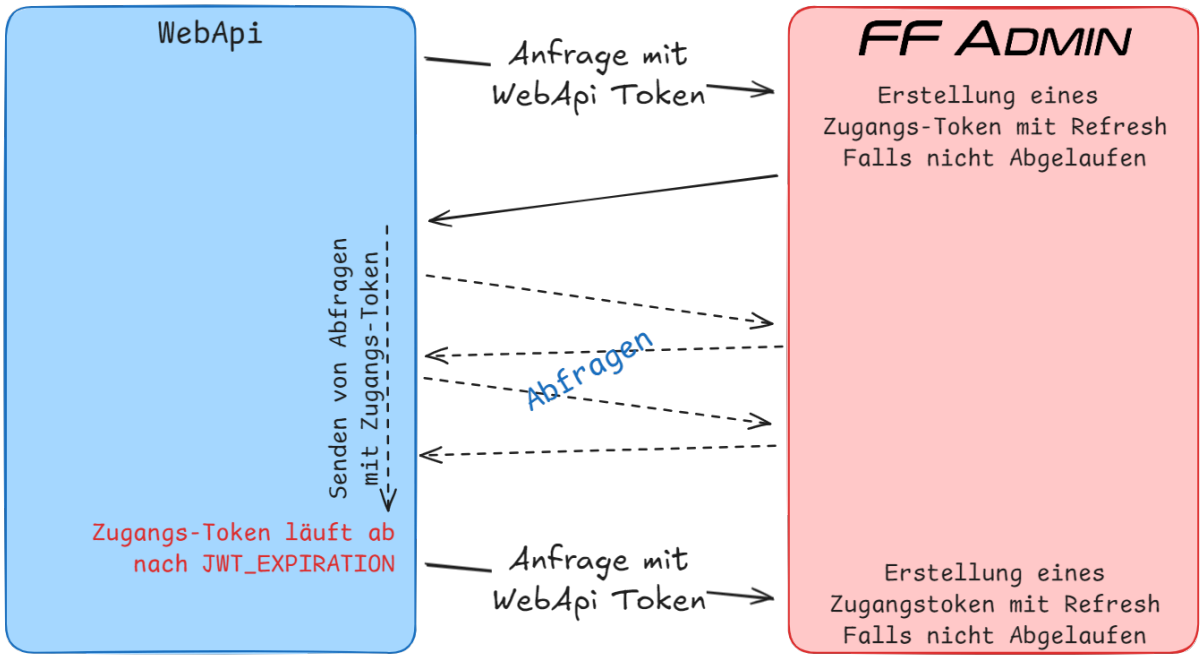
## 4.16.2 Berechtigungen




Einer WebApi können im Vergleich zu Benutzern nur direkte Berechtigungen zugewiesen werden (analog zu den Rollen).
Das Berechtigungsmodell unterscheidet zwischen Lese-, Erstellungs-, Änderungs- und Löschberechtigungen. Die Berechtigungen sind abgestuft. Das heißt, wer anlegen darf, darf auch lesen. Wer bearbeiten darf, darf auch anlegen und dementsprechend auch lesen usw.
Die Berechtigungen sind entsprechend der Bereiche in der Navigationsleiste gruppiert. Wenn die Berechtigung für einen Abschnitts auf Lesen gesetzt ist, kann der Benutzer mit dieser Berechtigung Daten in allen Modulen des Abschnitts lesen.
Erweiterte Berechtigungen können pro Modul gesetzt werden. Dabei gelten die Berechtigungen des Abschnitts immer mehr als die eines Moduls innerhalb des Abschnitts.

## 4.16.3 WebApi Zugriffsablauf

Die WebApi bezieht den Zugang wie folgt:



### 4.16.4 API-Clients

Zur einfacheren Verwendung der WebApi in eigenen Anwendungen gibt es fertige API-Clients für TypeScript, Python und Go. Sie werden aus der OpenAPI-Spezifikation des jeweiligen Servers erzeugt und decken damit alle Routen ab, auch die des Core.

| App | TypeScript (npm) | Python | Go |
|---|---|---|---|
| FF Admin | `@ff-admin/ff-admin-api` | `ff-admin-api` | `code.jk-effects.cloud/ff-admin/ff-admin-api` |
| FF Operation | `@ff-admin/ff-operation-api` | `ff-operation-api` | `code.jk-effects.cloud/ff-admin/ff-operation-api` |

Die Version eines Pakets folgt der Linie der Anwendung: `1.25.x` passt z.B. zu FF Admin 1.25. Für jede unterstützte Linie (Abschnitt 2.12.1) gibt es ein passendes Paket. Alle Pakete liegen in der Registry unter https://code.jk-effects.cloud/ff-admin/ff-admin-api-clients.

Alle Clients bieten dasselbe:

- `WebApiClient` für die Anmeldung mit einem WebApi-Token und `UserClient` für die Anmeldung als Benutzer (Passwort oder TOTP).
- Automatische Erneuerung des Zugangs-Tokens, wenn dieses abgelaufen ist.
- Einen Fehlertyp `ApiError` mit Status und den Fehlerdetails des Servers.
- Alle Routen der Spezifikation. Routen, die für WebApis gesperrt sind (siehe oben), schlagen mit einem WebApi-Token fehl und sind nur mit einem `UserClient` nutzbar.

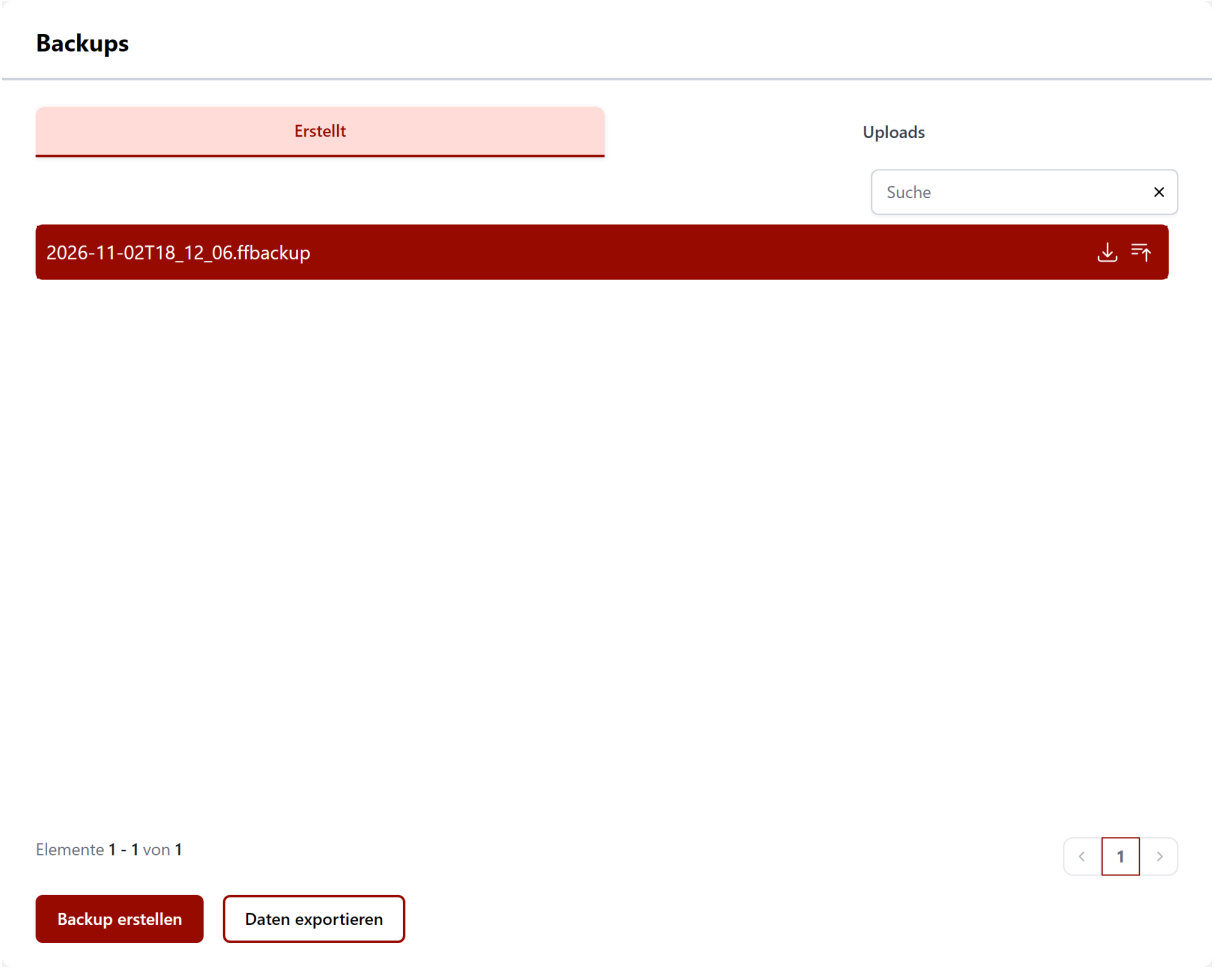
Beispiel in TypeScript:

```typescript
import { WebApiClient } from "@ff-admin/ff-admin-api";

const client = new WebApiClient({
  server: "https://admin.ff-beispiel.de",
  webapiToken: "<webapi-token>",
});
await client.initialize();

const page = await client.api.admin.member.getAllMembers({ query: { offset: 0, count: 25 } });
```

Anleitungen zur Installation und Nutzung in allen Sprachen stehen in der README des Repositories. Welche Routen und Felder es gibt, zeigt die API-Dokumentation des Servers, sofern sie aktiviert ist.

Das bisherige Paket `@ff-admin/webapi-client` (https://code.jk-effects.cloud/FF-Admin/ff-admin-webapi-client) bleibt verfügbar und enthält zusätzlich die Anbindung an das CMS von FF Webpage. Für neue Anbindungen an FF Admin und FF Operation sollten die oben genannten Clients verwendet werden.

## 4.17 Backups

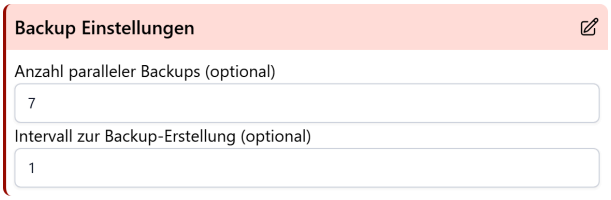



Backups sind automatisch gesicherte Datenstände der Anwendung, die jederzeit wiederhergestellt werden können. Die Dateien sind verschlüsselt und werden beim Einspielen auf Veränderungen geprüft.

Ein Backup lässt sich nur auf einem Server derselben Anwendung einspielen, der dasselbe `APPLICATION_SECRET` und `APPLICATION_SALT` verwendet. Um Daten in eine andere Instanz zu übertragen, gibt es den Export (siehe „Daten exportieren“).

**Wichtig:** Backups und Exporte enthalten nur die Datenbank. Dateien wie Dokumente oder Fotos sind nicht enthalten, der Ordner `/app/files` muss deshalb zusätzlich gesichert werden.

Backups herunterladen, wiederherstellen und Daten exportieren dürfen nur Administratoren.

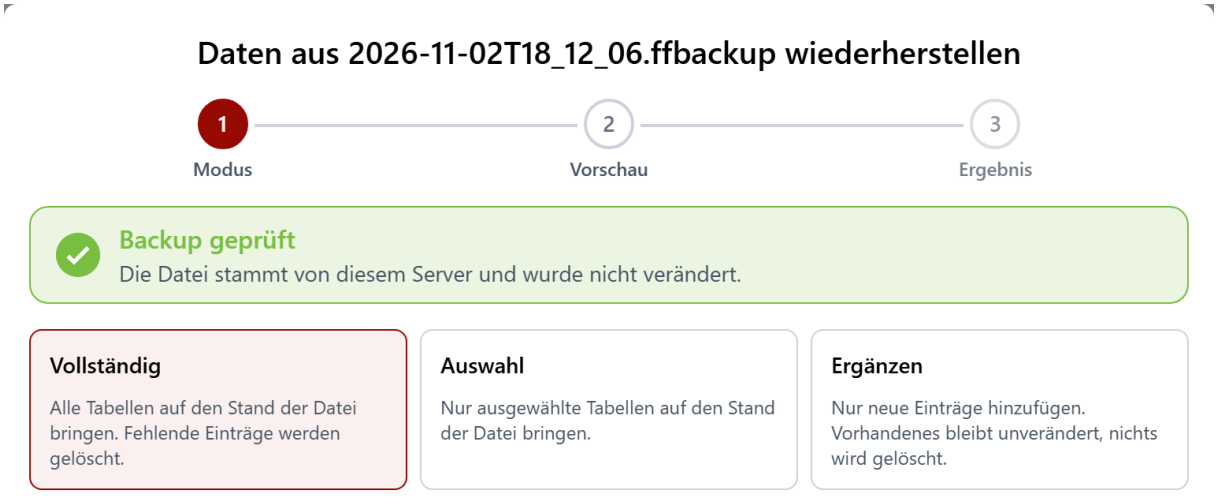



Backups werden in einem festgelegten Intervall erstellt. Das kleinste Intervall beträgt täglich. Zusätzlich kann konfiguriert werden, wie viele Backups parallel zur Verfügung stehen sollen. Standardmäßig ist eine Anzahl von 7 eingestellt, d.h. die gespeicherten Backups reichen immer eine Woche in die Vergangenheit. Die Anzahl bezieht sich dabei nicht auf die Anzahl an Backups im System, sondern auf die Anzahl Tage, an denen ein Backup gemacht wurde. Heißt, dass 2 Backups des selben Tages als eines gewertet wird.
Das Intervall und die Anzahl and Backups, welche parallel gespeichert sein sollen, können in den Einstellungen Abschnitt 4.12 gesetzt werden.

Eine Funktion, die mit den Backups eingeführt wurde, ist `AUTO RESTORE`.
`AUTO RESTORE` ermöglicht die automatische Wiederherstellung des letzten Backups - sofern vorhanden - wenn die Datenbank beim Start des Servers leer ist. Verwendet wird das neueste Backup, das sich prüfen lässt. Exporte und veränderte Dateien werden übersprungen.

### 4.17.1 Backup wiederherstellen

Das Symbol neben einem Backup öffnet einen Assistenten. Er prüft zuerst die Datei und zeigt das Ergebnis an:



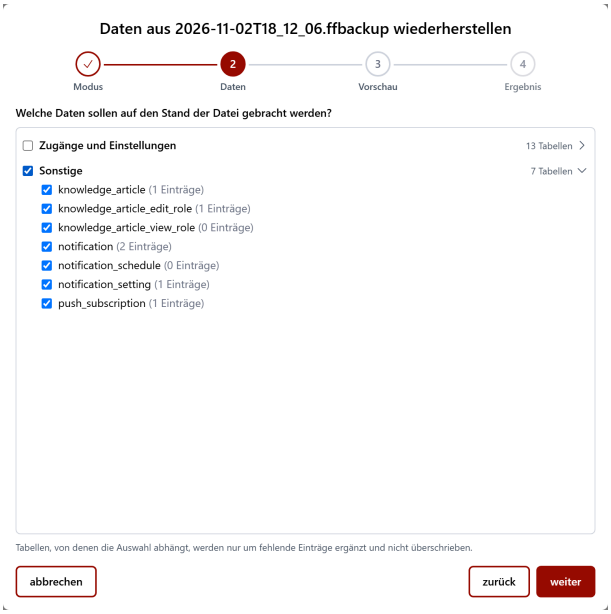
| Anzeige | Bedeutung | Einspielen durch |
|---|---|---|
| Backup geprüft | Die Datei stammt von diesem Server und wurde nicht verändert. | Administratoren |
| Export geöffnet | Das Passwort eines Exports passt. Das sagt nichts darüber aus, wer die Datei erstellt hat. | Administratoren |
| Passwort nötig | Die Datei ist ein Export und muss erst mit ihrem Passwort geöffnet werden. | niemand |
| Anderer Schlüssel | Die Datei wurde auf einem Server mit anderem Secret oder Salt erstellt. | niemand |
| Datei verändert | Die Datei wurde nach dem Erstellen verändert. | niemand |
| Altes Format, nicht prüfbar | Die Herkunft einer Datei aus einer früheren Version lässt sich nicht zuverlässig prüfen. | nur Owner, nach Bestätigung |

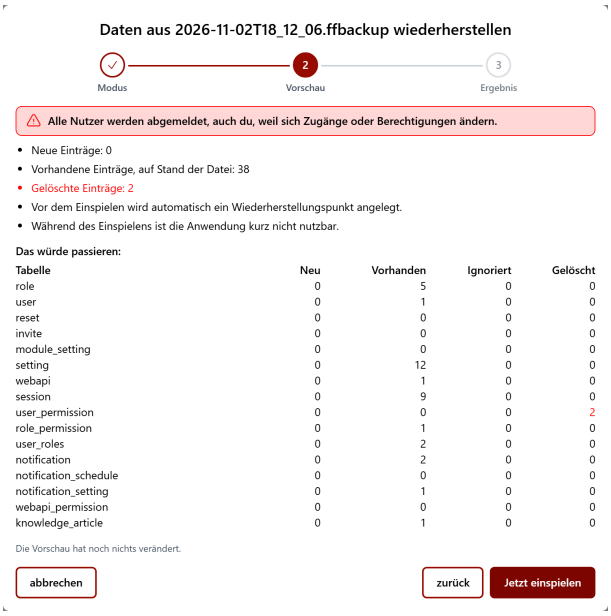
Backups liegen je nach Alter in verschiedenen Formaten vor. Aktuelle Backups haben die Endung .ffbackup. Ältere .json-Dateien lassen sich weiterhin einspielen, sofern sie bereits eine Signatur enthalten (Format 2). Noch ältere .json-Dateien (Format 1) werden abgelehnt.
Danach wird festgelegt, wie die Daten übernommen werden:

- **Vollständig:** Alle Tabellen werden auf den Stand der Datei gebracht. Einträge, die in der Datei fehlen, werden gelöscht.
- **Auswahl:** Nur die ausgewählten Tabellen werden auf den Stand der Datei gebracht. Tabellen, von denen die Auswahl abhängt, werden nur um fehlende Einträge ergänzt.
- **Ergänzen:** Es werden nur neue Einträge hinzugefügt. Vorhandenes bleibt unverändert, gelöscht wird nichts.

Eine Auswahl, die Zugänge und Einstellungen enthält, darf nur der Owner einspielen. Bei Exporten steht „Vollständig“ nicht zur Verfügung.

Bei **Auswahl** und **Ergänzen** folgt die Auswahl der Daten. Die Tabellen sind nach Bereichen gruppiert und lassen sich aufklappen, um einzelne Tabellen zu wählen.




Vor dem Einspielen zeigt eine **Vorschau** je Tabelle, was passieren würde. Die Vorschau selbst verändert nichts. Sie zeigt auch, ob alle Nutzer abgemeldet werden. Das passiert nur, wenn sich Nutzer, Rollen, Berechtigungen, Passkeys oder Sitzungen ändern. Reine Einstellungen oder Stammdaten melden niemanden ab.

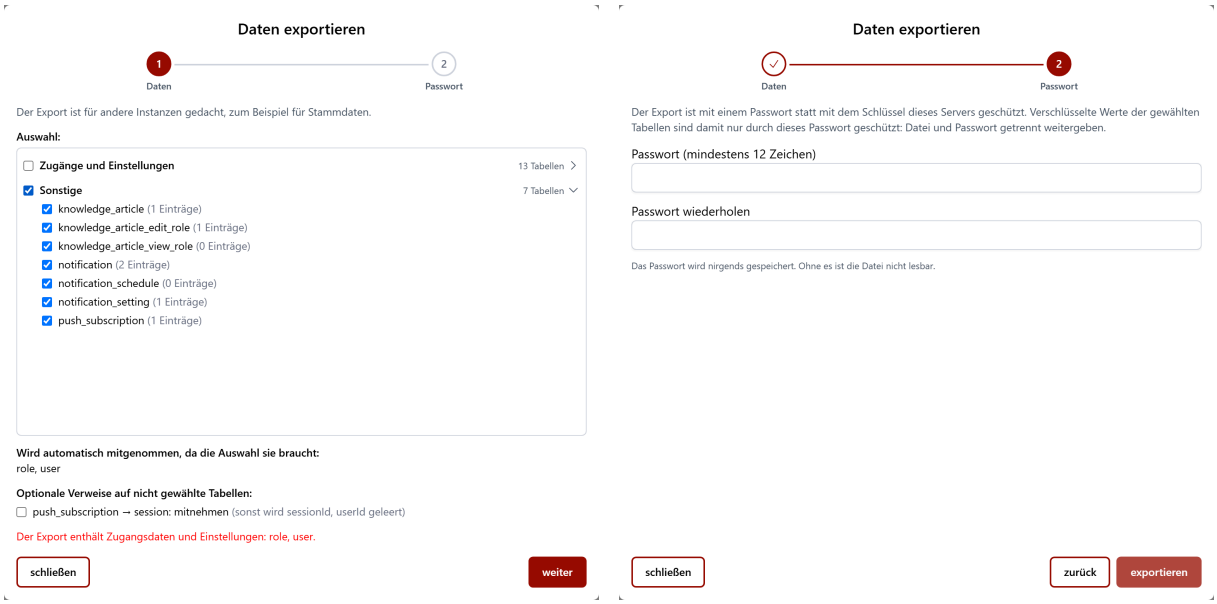
Mit `Jetzt einspielen` wird automatisch ein Wiederherstellungspunkt angelegt, mit dem sich der Vorgang zurücknehmen lässt. Während des Einspielens ist die Anwendung kurz nicht nutzbar. Das Ergebnis wird anschließend in derselben Form angezeigt.

**Das sollte man beachten:**

- **Zugangsdaten bleiben erhalten:** Benutzername, Passwort bzw. TOTP, E-Mail-Adresse und Owner-Status vorhandener Nutzer ändert eine Wiederherstellung nie. Solange es bereits Nutzer gibt, wird auch kein neuer Nutzer zum Owner.
- **Neuere Konten verlieren Rechte:** Ein Nutzer, der nach dem Backup angelegt wurde, bleibt bei „Vollständig“ bestehen, verliert aber Rollen, Berechtigungen und Passkeys. Das gilt auch für einen neuen Administrator, der die Wiederherstellung selbst ausführt. Die Vorschau zeigt das als gelöschte Einträge der Tabellen `user_roles`, `user_permission` und `passkey`.

- **Ignorierte Einträge:** Ist ein eindeutiger Wert schon vergeben, etwa der Name einer neu angelegten Rolle, bleibt der vorhandene Eintrag bestehen und der aus der Datei wird ignoriert. Vorschau und Ergebnis zeigen die Anzahl je Tabelle.
- **Nicht lesbare Werte:** Lässt sich ein verschlüsselter Wert der Datei nicht lesen, wird er gemeldet. Vorhandene Einträge behalten dann ihren Wert, neue bleiben an dieser Stelle leer.

### 4.17.2 Daten exportieren

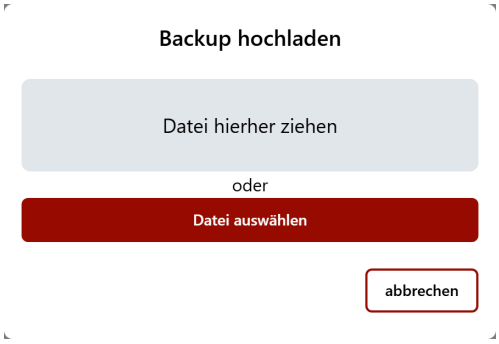



Ein Export überträgt Daten in eine andere Instanz, zum Beispiel vorbereitete Stammdaten als Ausgangspunkt für eine neue Installation. Über `Daten exportieren` werden die gewünschten Daten ausgewählt und ein Passwort mit mindestens 12 Zeichen vergeben. Die Datei wird direkt heruntergeladen und bleibt nicht auf dem Server.

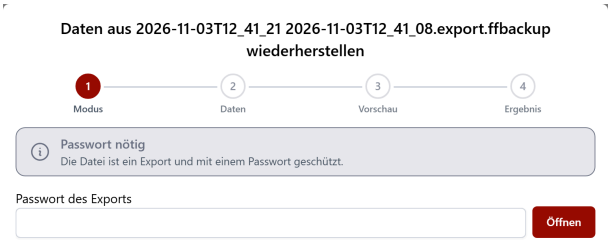
- Tabellen, die die Auswahl zwingend braucht, werden automatisch mitgenommen. Optionale Verweise auf nicht gewählte Tabellen können mitgenommen werden, sonst werden sie geleert.
- Exporte, die Zugänge und Einstellungen enthalten, darf nur der Owner erstellen und einspielen.

**Achtung:** Ein Export ist nur durch sein Passwort geschützt, auch nachdem Secret oder Salt des Servers gewechselt wurden. Datei und Passwort sollten getrennt weitergegeben und wie Zugangsdaten behandelt werden. Das Passwort wird nirgends gespeichert, ohne es ist die Datei nicht lesbar.

### 4.17.3 Backup hochladen

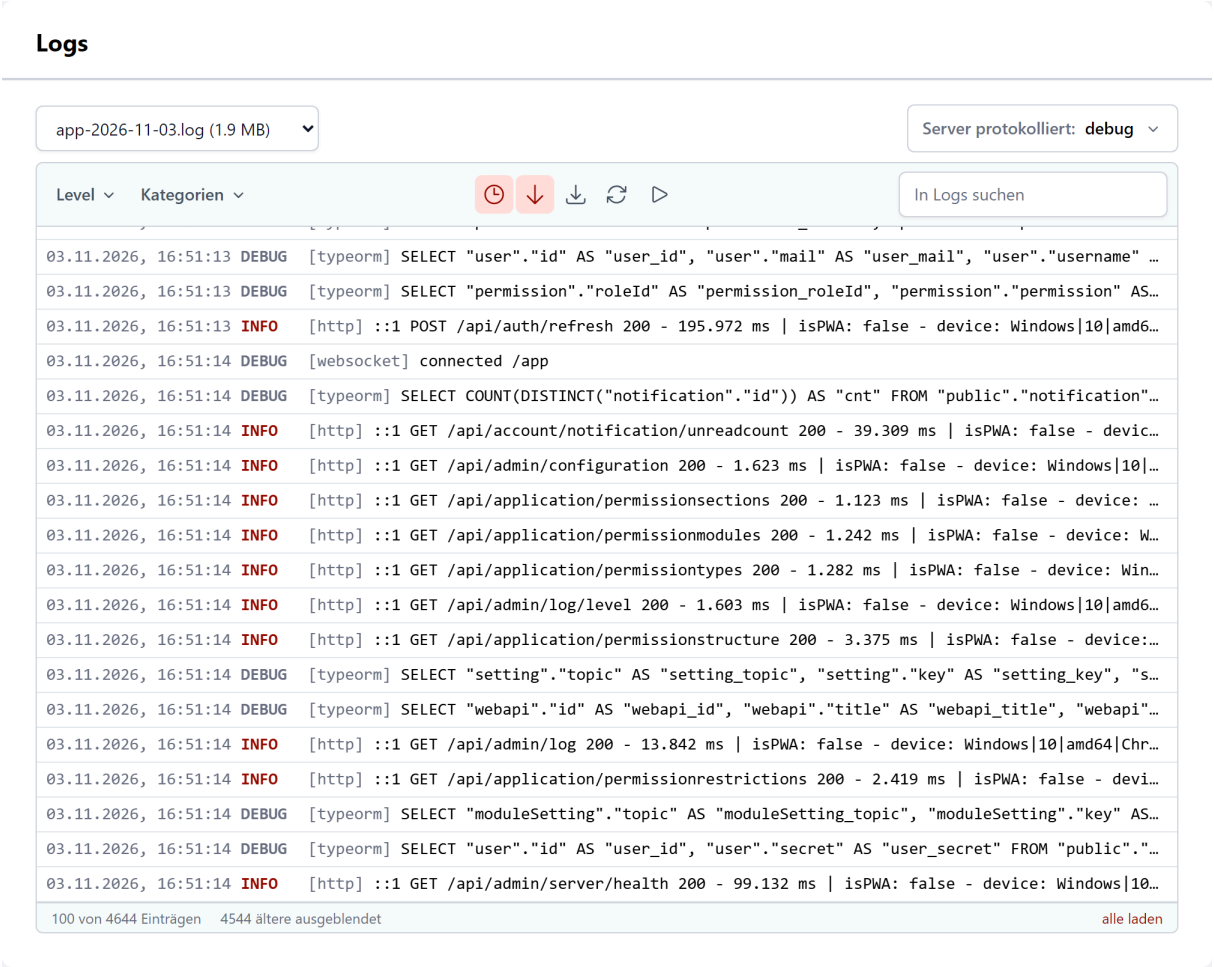



Heruntergeladene Backups und Exporte können jederzeit hochgeladen und im Admin gespeichert werden. Im Gegensatz zu den automatisch erstellten Backups sind die Slots nicht begrenzt. Es können also beliebig viele Dateien hochgeladen und gespeichert werden.
Eine Datei darf standardmäßig bis zu 200 MB groß sein, die Grenze lässt sich mit `BACKUP_UPLOAD_MAX_SIZE` anpassen. Dateien anderer Anwendungen werden abgelehnt.




**Export einspielen:** Die Datei hochladen, das Symbol zum Wiederherstellen wählen und mit dem Passwort öffnen. Anschließend werden die Daten üblicherweise mit „Ergänzen“ übernommen.
**Instanz umziehen:** Für einen vollständigen Umzug werden `APPLICATION_SECRET` und `APPLICATION_SALT` auf dem neuen Server unverändert übernommen und ein Backup zusammen mit dem Ordner `/app/files` übertragen.

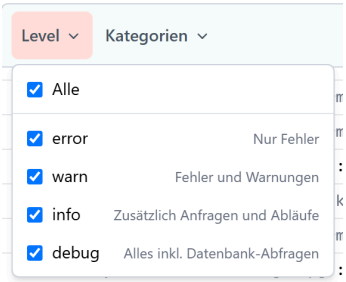
# 4.18 Logs




Unter `Verwaltung > Logs` können die Log-Dateien des Servers direkt in der Anwendung eingesehen werden, ohne Zugriff auf den Server oder den Docker-Container zu benötigen. Wie die Log-Dateien geschrieben und aufbewahrt werden, ist unter Logging (Abschnitt 2.8) beschrieben.

Für die Ansicht wird die Leseberechtigung des Moduls `Logs` im Abschnitt Verwaltung benötigt. Da die Logs technische Details wie IP-Adressen und Gerätenamen der Anfragen enthalten, sollte diese Berechtigung nur an Nutzer vergeben werden, die den Betrieb der Anwendung betreuen.

### 4.18.1 Ansicht




Über die Auswahl oben links wird die anzuzeigende Log-Datei gewählt. Neben dem Dateinamen ist die Größe der Datei angegeben. Angezeigt werden zunächst die neuesten Einträge der Datei, ältere Einträge können über `alle laden` nachgeladen werden. Ein Klick auf einen Eintrag zeigt den vollständigen Text sowie zusätzliche Details an.

Die Einträge lassen sich wie folgt eingrenzen und darstellen:

- **Level:** Anzeige nur bestimmter Stufen, z.B. ausschließlich Fehler und Warnungen.
- **Kategorien:** Anzeige nur bestimmter Bereiche des Servers, z.B. API-Anfragen (`http`) oder Datenbank (`typeorm`).
- **Suche:** Volltextsuche innerhalb der Einträge.
- **Zeitstempel:** Ein- und Ausblenden des Zeitpunkts der Einträge.
- **Neuesten Einträgen folgen:** Springt an das Ende der Datei.
- **Herunterladen:** Lädt die ausgewählte Datei vollständig herunter, z.B. um sie bei einer Fehlermeldung weiterzugeben.
- **Neu laden:** Lädt die Datei erneut vom Server.
- **Live:** Neue Einträge werden direkt am Ende angezeigt, ohne die Datei neu laden zu müssen.

### 4.18.2 Protokollierung ändern



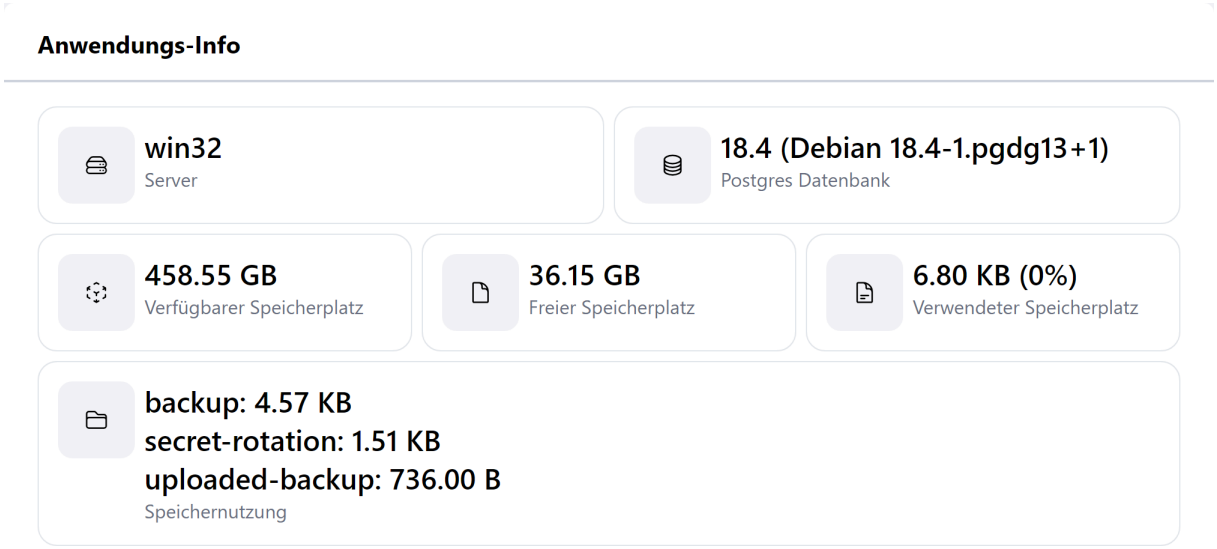

Oben rechts wird angezeigt, mit welchem Detailgrad der Server aktuell protokolliert. Mit der Berechtigung zum Bearbeiten des Moduls `Logs` kann dieser Wert geändert werden, z.B. kurzzeitig auf `debug`, um einen Fehler nachvollziehen zu können.

Die Änderung gilt sofort für alle neuen Einträge, aber nur bis zum nächsten Neustart des Servers. Solange ein abweichender Wert gesetzt ist, wird dieser als `temporär` gekennzeichnet und kann über das Menü auf den konfigurierten Wert (`LOG_LEVEL`) zurückgesetzt werden. Ein dauerhafter Wechsel erfolgt ausschließlich über die Umgebungsvariable `LOG_LEVEL` (siehe Abschnitt 2.5).

## 4.19 Anwendungsinformationen und Versionsmanagement

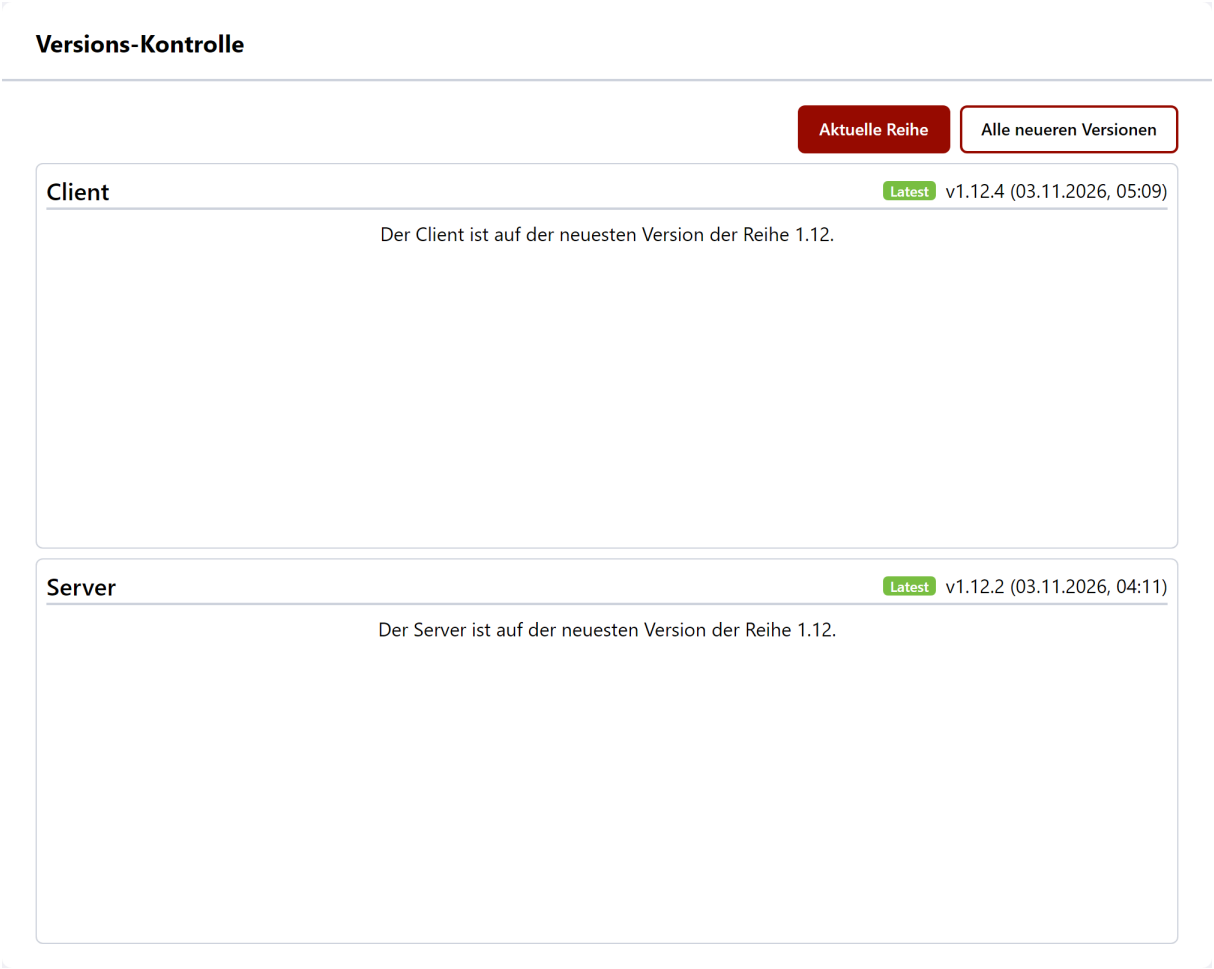
Die Anwendung stellt in der Oberfläche Informationen zum Server, der Datenbank und dem Speicherbedarf sowie der aktuellen Version und möglichen verfügbaren Updates zur Verfügung.

### 4.19.1 Anwendungsinformationen




Die Anwendungsinformationen geben einen aktuellen Status der Anwendung und dem Server aus. Angezeigt werden das Betriebssystem des Servers und die aktuelle Version der Datenbank. Die Version der Datenbank bestimmt unter anderem die Generierung von Ids.
Weiterhin wird der aktuelle Speicherbedarf und der verfügbare oder der zugewiesene Speicherplatz angezeigt. Die Speicherinformationen werden auch mit der Nutzung der einzelnen Module im Dateisystem angegeben.

### 4.19.2 Versionsmanagement




Die Oberfläche stellt außerdem Informationen zur aktuell installierten Version auf dem Server bereit. Die lokal installierte PWA-Version kann sich von der Version auf dem Server unterscheiden. Die Anwendung selbst stellt Versionsinformationen unter `/_clientinfo.json` bereit. Diese Version wird auch für die Prüfung verfügbarer Updates verwendet.

Die Versions-Kontrolle zeigt für Client und Server jeweils die installierte Version mit Veröffentlichungsdatum sowie die verfügbaren neueren Versionen an. Die Informationen zu den neuen Versionen stammen aus den Git-Releases der Anwendungen.

Neben der Version zeigt eine Markierung den Support-Status der installierten Reihe (Abschnitt 2.12.1):

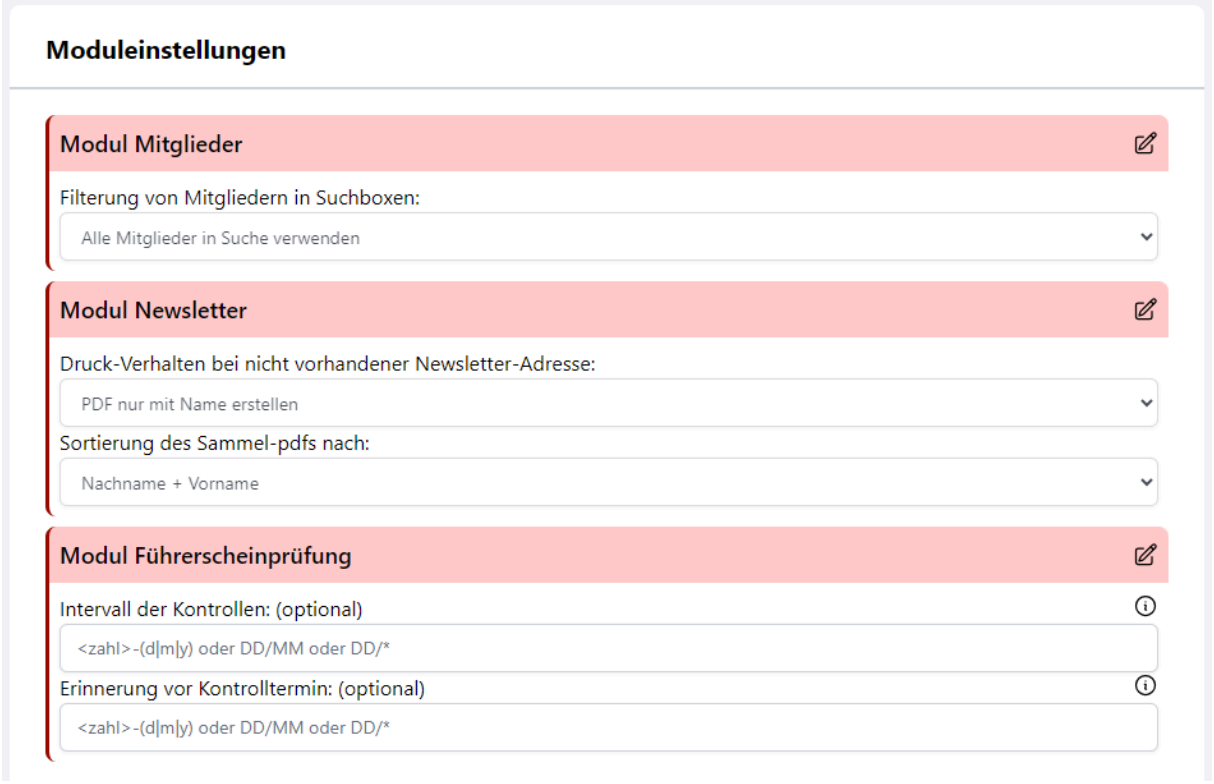
- **Latest:** Die installierte Version gehört zur neuesten Reihe.
- **Supported:** Die Reihe wird weiterhin mit Fehler- und Sicherheitskorrekturen versorgt. Zusätzlich werden die neueste Version der Reihe und die neueste Version insgesamt angegeben.
- **EOL:** Die Reihe wird nicht mehr unterstützt und erhält keine Sicherheitskorrekturen mehr. Ein rot hinterlegter Hinweis empfiehlt ein Update auf eine unterstützte Version.

Der Support-Status wird aus der zentralen `support.json` bezogen. Ist diese nicht erreichbar, entfällt die Markierung.
Über die Auswahl oberhalb der Liste lässt sich festlegen, welche neueren Versionen angezeigt werden:

- **Aktuelle Reihe** (Standard): nur neuere Patch-Versionen der installierten Reihe, also Updates ohne neue Funktionen.
- **Alle neueren Versionen:** auch Versionen neuerer Reihen.

Die Versionen werden nach Versionsnummer absteigend sortiert, die neueste Version steht oben.

## 4.20 Modulspezifische Einstellungen




Das Modul Moduleinstellungen erweitert das bestehende Einstellungssystem um spezifische Optionen, die ausschließlich einzelne Module betreffen.

Die Einstellungen zu den Modulen sind nur sichtbar, wenn ein Nutzer sowohl die Berechtigung für die Moduleinstellungen als auch für das jeweilige Modul besitzt. Dadurch wird sichergestellt, dass nur autorisierte Personen modulbezogene Anpassungen vornehmen können.

Mit den Moduleinstellungen lassen sich Funktionen und Darstellungen einzelner Module gezielt

steuern, ohne das Gesamtsystem zu beeinflussen. Folgende Moduleinstellungen stehen aktuell zur Verfügung:

**Mitglieder:** Legt fest, welche Mitglieder in Suchfeldern anderer Module zur Auswahl angeboten werden - entweder alle Mitglieder oder nur solche mit aktuell laufender Mitgliedschaft. Zusätzlich kann einzeln aktiviert werden, ob Aufnahmeantrag, Urkunden und Aus-/Fortbildungsnachweise an den pdf-Ausdruck eines Mitglieds angehängt werden (Abschnitt 4.1.1). Standardmäßig ist keine dieser Optionen aktiv.

**Newsletter:** Steuert das Druckverhalten (Abschnitt 4.4) - u.a. ob und wie Termine im Ausdruck sortiert werden.

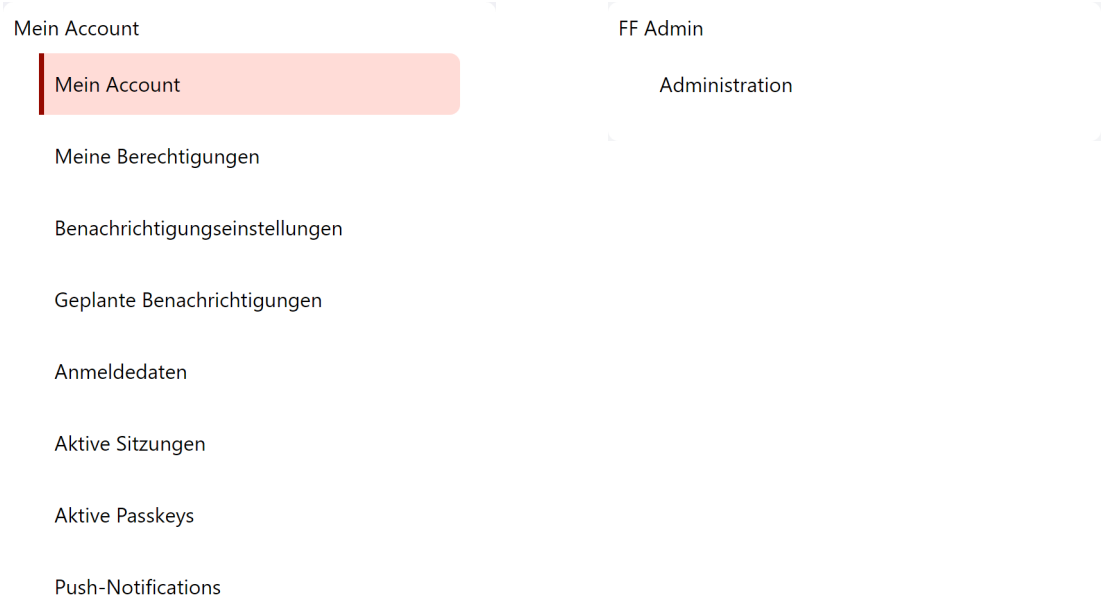
**Führerscheinkontrolle:** Legt das Prüfintervall sowie die Erinnerungsfrist vor der nächsten fälligen Kontrolle fest (Abschnitt 4.5).

**Kalender:** Aktiviert und konfiguriert die Synchronisierung der Kalendereinträge mit FF Webpage, inklusive Auswahl der zu synchronisierenden Terminarten (Abschnitt 4.2).

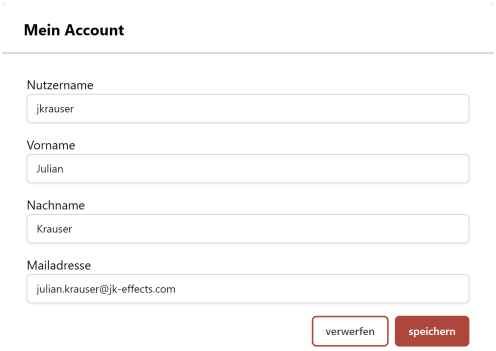
**Webpage WebAPI Connect:** Hinterlegt die Zugangsdaten (URL und Token), über die FF Admin mit FF Webpage kommuniziert - Voraussetzung für die Kalender-Synchronisierung.

# 5 Benutzerbereich und Accounteigenschaften

Der Accountbereich ist für jeden Benutzer durchgehend zugänglich und gliedert sich in allgemeine Kontoinformationen sowie einen Administrations- bzw. Eigentümerbereich.




## 5.1 Zugang und Anmeldedaten




Die Kontoinformationen sind in zwei separate Bereiche unterteilt: Der Bereich **Mein Account** zeigt allgemeine Accountdaten wie Benutzername und E-Mail-Adresse an. Jeder Nutzer kann seinen Benutzernamen und seine E-Mail-Adresse nach einer erneuten Anmeldung selbst ändern, Administratoren können diese Daten auch für andere Nutzer bearbeiten. Im Bereich **Anmeldedaten** können die Einstellungen für die Authentifizierung verwaltet werden - entweder das zeitbasierte Einmalpasswort (TOTP) oder das klassische Passwort.

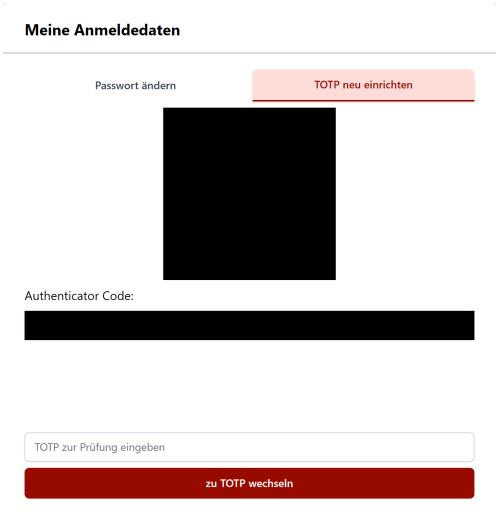
### 5.1.1 Anmeldeoptionen

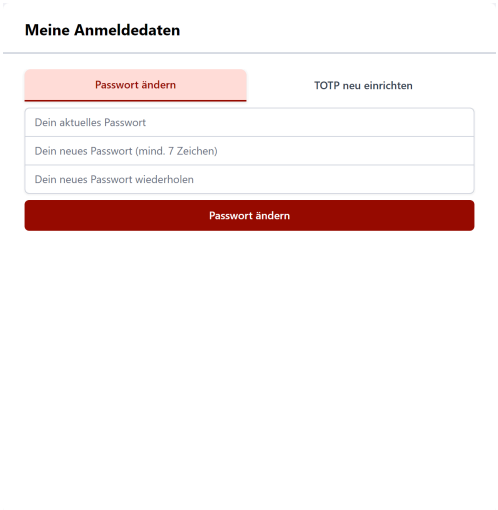
Nutzer können sich entweder mit einem zeitbasierten Einmalpasswort (TOTP) oder einem klassischen Passwort anmelden. Je nach gewählter Methode stehen folgende Optionen zur Verfügung:

- **TOTP-Anmeldung:** Der TOTP-QR-Code kann erneut angezeigt werden, um die Authentifizierung auf einem weiteren Gerät einzurichten. Alternativ kann zu einer Passwort-basierten Anmeldung gewechselt werden.
- **Passwort-Anmeldung:** Das Passwort kann geändert oder zu einer TOTP-basierten Anmeldung gewechselt werden.
- **Passkey-Anmeldung:** Passkeys können als alternative Loginmethode erstellt werden und erleichtern den Login. Passkeys ersetzen allerdings nicht das gesetzte Passwort bzw. das eingerichtete TOTP. (Abschnitt 5.5)
- **QRCode-Anmeldung:** Auch kann sich auf anderen Geräten via QR-Code angemeldet werden. Dazu muss über den Knopf `Login mit QRCode` in der Loginmaske der QRCode angezeigt werden. In der App selbst kann der Code über das User-Menü gescannt werden.

### 5.1.2 Änderung der Anmeldedaten

Unter dem Menüpunkt Anmeldedaten kann der QR-Code für das aktuelle TOTP abgerufen werden. Auch können unter diesem Menüpunkt die Anmeldedaten geändert oder zwischen Anmeldung mittels TOTP bzw. Passwort gewechselt werden.
Um unberechtigten Zugriff zu verhindern wird bei Aufruf der Seite nochmals eine Authentifizierung angefordert.
Werden Passwort oder Anmeldeart geändert, bleibt nur die aktuelle Sitzung bestehen, alle anderen Geräte werden abgemeldet. Nach dem Zurücksetzen des Zugangs per E-Mail enden alle Sitzungen.
Verwalteten Konten (siehe „Verwaltete Konten“) stehen TOTP und das Zurücksetzen per E-Mail nicht zur Verfügung.



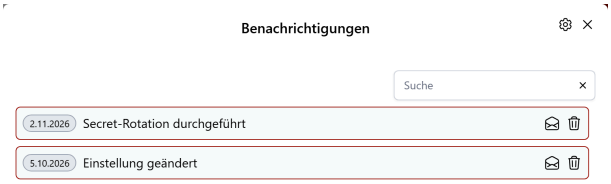
## 5.2 Benachrichtigungen

Benutzer können Benachrichtigungen zu Vorgängen der Anwendung einstellen und erhalten. Benachrichtigungen werden immer gespeichert und können über die Glocke in der Navigationsleiste angezeigt werden.
Für die Module, welche Benachrichtigungen versenden, kann der Versandweg eingestellt werden. Zur Verfügung stehen aktuell:

- **In-App**: Nachrichten werden direkt in der Anwendung angezeigt, wenn diese geöffnet ist
- **E-Mail**: Benachrichtigungen werden per E-Mail versendet
- **Web-Push**: Benachrichtigungen werden direkt auf das Gerät gesendet (siehe Abschnitt 5.3)

Einstellungen zu den Benachrichtigungen können nur für Module vorgenommen werden, auf die der Account Zugriff hat.

### 5.2.1 Benachrichtigungsanzeige




In der Benachrichtigungsanzeige werden alle Benachrichtigungen zentral aufgelistet. Für jede Benachrichtigung sind folgende Informationen ersichtlich:

- Der Nachrichteninhalt
- Das sendende Modul
- Die verwendeten Zustellwege (In-App, E-Mail, Web-Push)

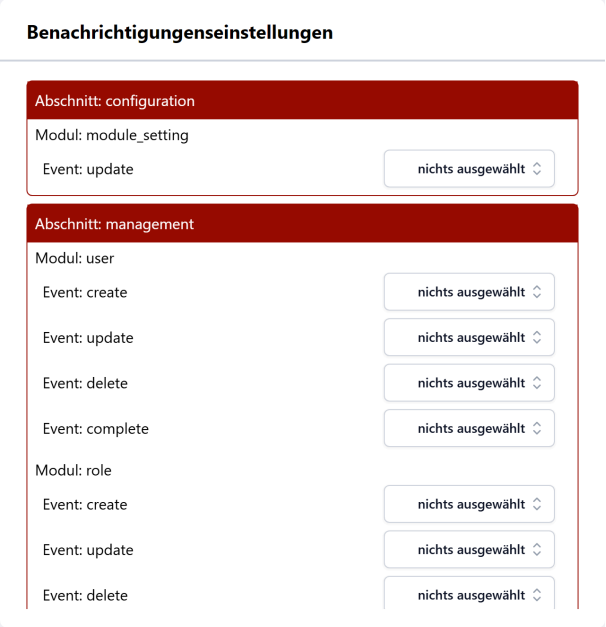
Hier können versendete Benachrichtigungen jederzeit nachgelesen werden.

### 5.2.2 Benachrichtigungseinstellungen

Die Benachrichtigungseinstellungen sind über das Zahnrad in der Nachrichtenanzeige oder im Usermenü unter „Mein Account“ erreichbar.
Für jedes Modul und Event können die Versandwege individuell konfiguriert werden:

- **In-App**: Direkte Anzeige in der Anwendung
- **E-Mail**: Versand per E-Mail
- **Web-Push**: Versand als Push-Benachrichtigung auf das Gerät (siehe Abschnitt 5.3)

Es können mehrere Versandarten gleichzeitig aktiviert werden.




Benachrichtigungen werden für folgende Ereignisse versendet:

- **create**: Neues Element wurde erstellt
- **update**: Element wurde aktualisiert
- **delete**: Element wurde gelöscht
- **complete**: Vorgang wurde abgeschlossen (Bsp.: Newsletter wurde fertig versandt)
- **remind**: Erinnerung an anstehende Aufgabe (Bsp.: Prüfung ist fällig)

### 5.2.3 Geplante Benachrichtigungen

Neben den ereignisbasierten Benachrichtigungen können Benutzer geplante Benachrichtigungen konfigurieren. Dabei handelt es sich um regelmäßig versendete Zusammenfassungen oder Erinnerungen, deren Inhalt dynamisch berechnet wird.
Die verfügbaren Benachrichtigungstypen werden von der Anwendung bereitgestellt und können je nach Instanz variieren (z.B. Erinnerungen an fällige Geräteprüfungen oder Führerscheinkontrollen).

**Konfiguration**

Für jeden verfügbaren Benachrichtigungstyp kann der Benutzer folgende Einstellungen vornehmen:

- **Aktiv/Inaktiv**: Einzelne Benachrichtigungen können ein- oder ausgeschaltet werden
- **Wochentage**: Auswahl der Tage, an denen die Benachrichtigung versendet werden soll (Sonntag bis Samstag)
- **Uhrzeit**: Festlegung der Sendezeit (Format HH:MM)
- **Kanäle**: Auswahl der Zustellwege — es können mehrere gleichzeitig aktiviert werden:
  - In-App: Anzeige in der Anwendung
  - E-Mail: Versand per E-Mail
  - Web-Push: Push-Benachrichtigung auf das Gerät (siehe Abschnitt 5.3)

Einige Benachrichtigungstypen sind als *täglich* markiert. Bei diesen Typen entfällt die Auswahl der Wochentage, da der Inhalt tagesaktuell ist und bei Auslassung eines Tages nicht nachgeholt werden kann (z.B. „heute fällige Prüfungen“). Für diese Benachrichtigungen kann nur die Sendezeit und die Kanäle konfiguriert werden.

**Versandlogik**
Das System prüft alle 5 Minuten, ob geplante Benachrichtigungen fällig sind. Eine Benachrichtigung gilt als fällig, wenn:
- sie aktiv ist
- der aktuelle Wochentag in der Konfiguration enthalten ist
- die konfigurierte Uhrzeit erreicht oder überschritten ist
- sie am heutigen Tag noch nicht versendet wurde

Ist eine Benachrichtigung fällig, wird der zugehörige Inhalt berechnet. Ergibt die Berechnung, dass es nichts zu melden gibt (z.B. keine fälligen Prüfungen), wird keine Benachrichtigung versendet. Andernfalls wird die Nachricht über alle konfigurierten Kanäle zugestellt.
Pro Benachrichtigungstyp und Benutzer wird maximal eine Nachricht pro Tag versendet, unabhängig davon, wie oft das System prüft.

## 5.3 Push-Benachrichtigungen

Push-Benachrichtigungen ermöglichen es, Mitteilungen direkt auf das Gerät zu erhalten — auch wenn die Anwendung nicht geöffnet ist. Sie funktionieren unabhängig davon, ob die App als PWA installiert ist oder im Browser verwendet wird.

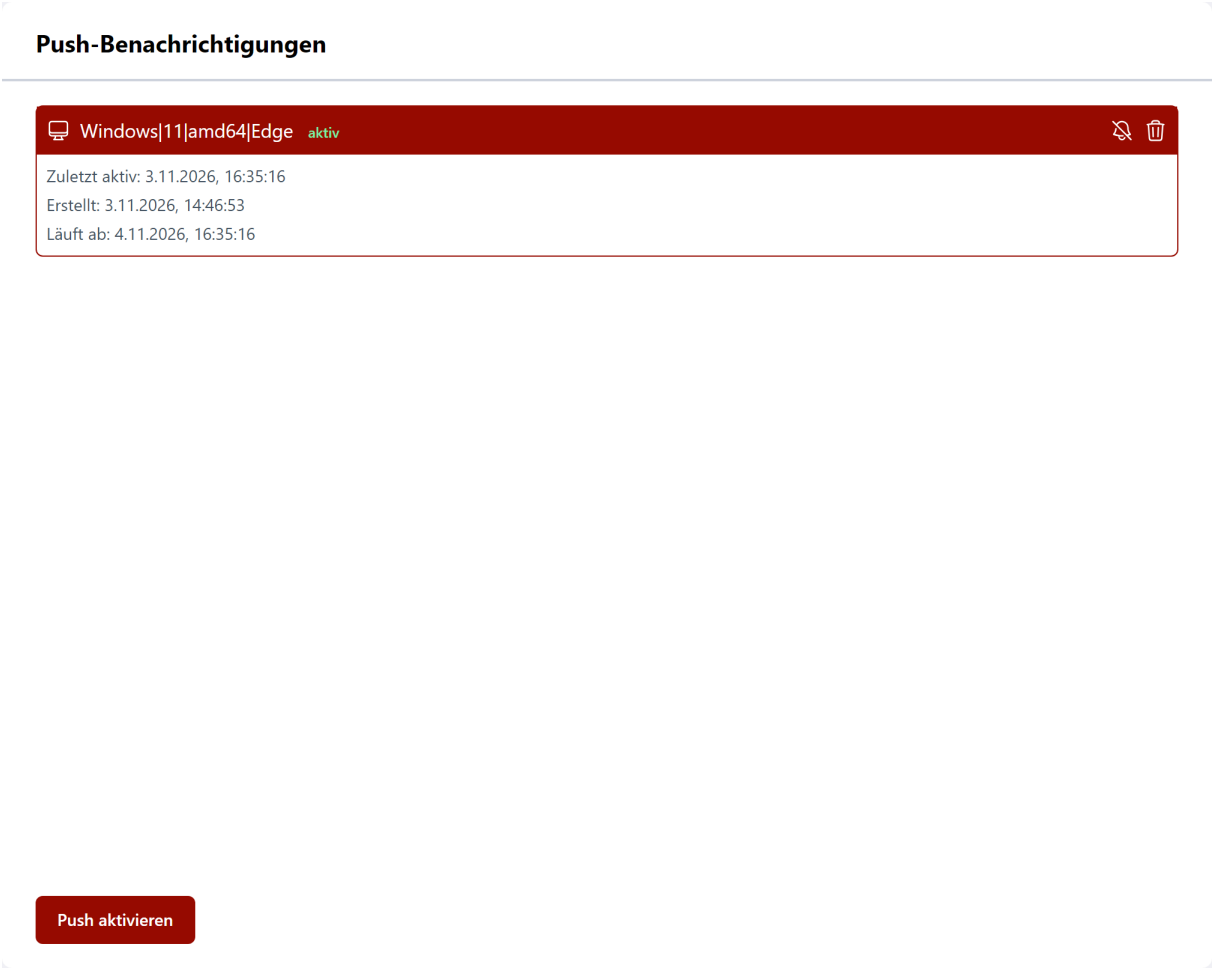
### 5.3.1 Funktionsweise

Push-Benachrichtigungen nutzen den Web-Push-Standard. Dabei wird eine verschlüsselte Verbindung zwischen dem Server und dem Gerät des Nutzers aufgebaut. Der Server kann darüber Nachrichten senden, die das Betriebssystem als Benachrichtigung anzeigt.
Voraussetzungen:
- Ein moderner Browser mit Push-Unterstützung (Chrome, Firefox, Edge, Safari 16+)
- Die Berechtigung für Benachrichtigungen muss im Browser erteilt sein
- Ein aktiver Service Worker muss registriert sein (wird automatisch verwaltet)

### 5.3.2 Push aktivieren

Die Aktivierung erfolgt über den Accountbereich unter `Push-Benachrichtigungen`. Nach Klick auf `Push aktivieren` fragt der Browser nach der Berechtigung, Benachrichtigungen anzeigen zu dürfen. Wird die Berechtigung erteilt, wird das Gerät automatisch für Push-Benachrichtigungen registriert. Der Eintrag erscheint in der Übersichtsliste.
**Hinweis:** Push wird auch automatisch bei jeder Anmeldung aktiviert, sofern die Berechtigung bereits erteilt wurde.

### 5.3.3 Übersicht und Verwaltung




In der Übersicht werden alle registrierten Geräte des Nutzers aufgelistet. Für jeden Eintrag sind folgende Informationen sichtbar:

- **Gerätetyp**: Mobilgerät, Tablet oder Desktop (mit passendem Icon)
- **Status**: aktiv oder inaktiv
- **Zuletzt aktiv**: Zeitpunkt der letzten Nutzung
- **Erstellt**: Zeitpunkt der Registrierung
- **Ablaufdatum**: Wann die Subscription automatisch ungültig wird

### 5.3.4 Deaktivieren und Löschen

Für jede registrierte Push-Subscription stehen zwei Aktionen zur Verfügung:
**Deaktivieren** (durchgestrichene Glocke):

- Die Subscription wird als inaktiv markiert
- Es werden keine Push-Nachrichten mehr an dieses Gerät gesendet

- Bei der nächsten Anmeldung auf dem betroffenen Gerät wird Push automatisch reaktiviert

**Löschen** (Mülleimer):

- Die Subscription wird dauerhaft aus der Datenbank entfernt
- Push wird auf dem betroffenen Gerät nicht automatisch reaktiviert
- Der Nutzer muss Push auf dem Gerät manuell erneut aktivieren

Beide Aktionen können remote ausgeführt werden — d.h. von einem anderen Gerät aus.

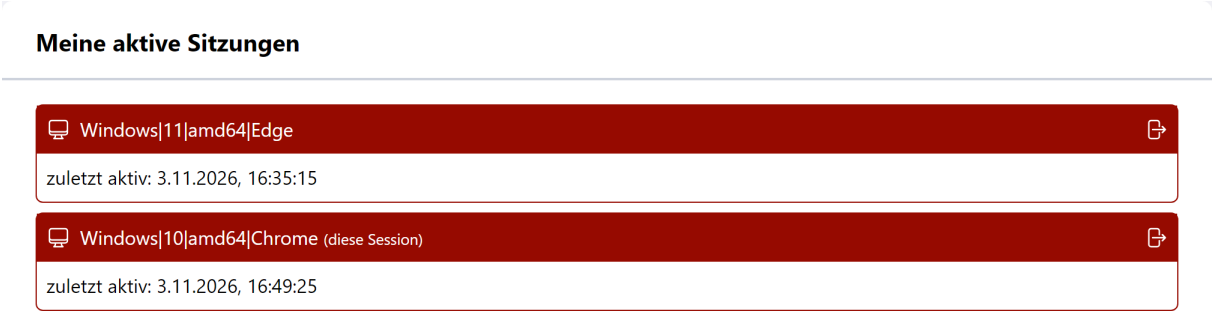
### 5.3.5 Lebenszyklus einer Push-Subscription

| Aktion | Auswirkung |
|---|---|
| Login | Push wird aktiviert (oder reaktiviert, falls deaktiviert) |
| Session-Refresh | Ablaufdatum wird verlängert |
| Logout | Push wird deaktiviert |
| Remote Deaktivieren | Push wird deaktiviert, nächster Login reaktiviert |
| Remote Löschen | Push wird entfernt, nur manuelle Neuaktivierung möglich |
| Ablaufdatum erreicht | Subscription wird beim nächsten Sendeversuch ignoriert |
| Browser-Berechtigung entzogen | Server erhält Fehlermeldung und deaktiviert automatisch |

### 5.3.6 Mehrere Benutzer auf einem Gerät

Push-Benachrichtigungen unterstützen die Nutzung eines Geräts durch mehrere Personen. Jeder Benutzer hat eine eigene Subscription — die Registrierung eines Nutzers beeinflusst die Push-Einstellungen anderer Nutzer auf demselben Gerät nicht.

## 5.4 Sessionverwaltung




Die Sessionverwaltung bietet Nutzern umfassende Kontrolle über ihre aktiven Anmeldungen. Mit dieser Funktion können Sie jederzeit einsehen, auf welchen Geräten Ihr Account derzeit aktiv ist und Details zu den entsprechenden Sitzungen abrufen.

### 5.4.1 Geräteerkennung und Datenschutz

Bei jeder Anmeldung erfasst das System automatisch Geräteinformationen wie Betriebssystem, Browser und Gerätetyp. Diese Daten werden direkt vom Gerät bereitgestellt und ermöglichen eine zuverlässige Geräteidentifikation – ohne dabei Fingerprinting-Techniken zur vollständigen Geräteprofilierung einzusetzen.

### 5.4.2 Verwaltung aktiver Sessions

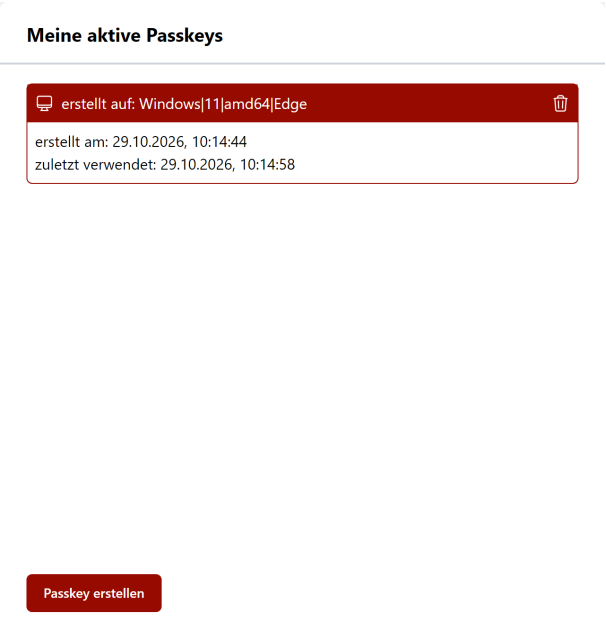
Sie haben die Möglichkeit, Sessions von anderen Geräten aus zu beenden:

- **Ist die App aktiv:** Das betroffene Gerät wird sofort abgemeldet.
- **App ist inaktiv:** Der Zugriff wird beim nächsten Öffnen der Anwendung blockiert - eine erneute Anmeldung ist erforderlich.

Diese Funktion erhöht die Sicherheit Ihres Accounts und gibt Ihnen vollständige Kontrolle über Ihre aktiven Sitzungen.

Sessions enden auch automatisch, zum Beispiel wenn das Passwort geändert oder der Zugang zurückgesetzt wird, das Konto deaktiviert wird oder eine Wiederherstellung die Zugänge ändert. Mit einer Session endet auch der Push-Empfang des jeweiligen Geräts.

## 5.5 Passkeys




Passkeys sind eine sichere Alternative zu Passwörtern. Sie basieren auf der FIDO2-Technologie und ermöglichen es einem Nutzer, sich mit biometrischen Daten (Fingerabdruck, Gesichtserkennung) oder einer PIN zu authentifizieren, ohne sich ein komplexes Passwort merken zu müssen.

Die Passkeys werden direkt auf dem Gerät gespeichert und ersetzen klassische Zugangsdaten komplett. Beim Login erkennt die Anwendung automatisch, dass ein Passkey vorhanden ist, und bietet diese Option direkt an - dadurch muss sich ein Nutzer nicht mehr an Passwörter erinnern.

**Vorteile von Passkeys**

- **Einfacherer Login**: Statt Passwort eingeben reicht Fingerabdruck oder Gesichtserkennung
- **Höhere Sicherheit**: Passkeys sind resistent gegen Phishing und Brute-Force-Attacken

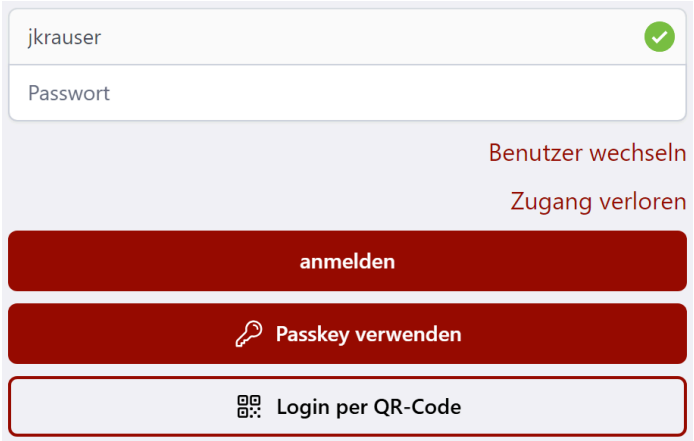
- **Gerätübergreifend**: Dein Passkey wird auf mehreren Geräten synchronisiert und kann auch auf anderen Geräten verwendet werden

### 5.5.1 Erstellung eines Passkeys

Ein Passkey kann nur erstellt werden, wenn der Nutzer bereits angemeldet ist. In den Kontoeinstellungen unter `Meine aktiven Passkeys` lässt sich ein neuer Passkey hinzufügen. Zum Erstellen und Löschen eines Passkeys ist eine erneute Anmeldung nötig.
Nach Klick auf `Passkey erstellen` führt das Gerät durch den Erstellungsprozess und fordert zur Bestätigung eine PIN oder biometrische Authentifizierung an.
Passkeys können auch über ein anderes Gerät erstellt werden. Nach Auswahl dieser Option wird ein QR-Code angezeigt, den man mit einem anderen Gerät scannen kann. Dadurch werden beide Geräte synchronisiert und der Passkey steht auf beiden zur Verfügung.

### 5.5.2 Login mit Passkeys

Passkeys können auf dem Gerät direkt nach der Erstellung verwendet werden. Ist der Nutzername schon Hinterlegt, wird ein Knopf unterhalb des Login-Formulars angezeigt:




Sollte noch kein Nutzername eingegeben sein, wird ein vorhandener Passkey zur Auswahl angezeigt, wenn ins Feld „Benutzername“ geklickt wird.
Bei Auswahl eines Passkeys erfolgt eine Bestätigung durch den Nutzer. Alternativ kann ein anderes Gerät zum Login verwendet werden: Nach Auswahl dieser Option wird ein QR-Code angezeigt, über den die Anmeldung auf einem anderen Gerät erfolgen kann.



# 5.6 Übertragung Administration

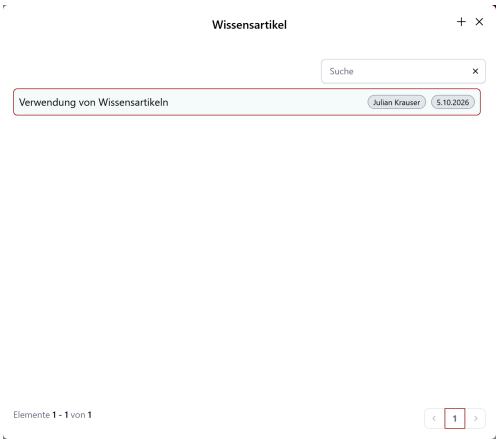


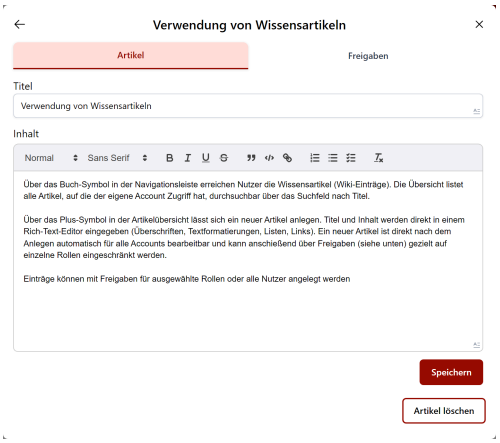

Bei der Übertragung der Administration wird ein anderer Nutzer zum Owner ernannt und erhält damit Vollzugriff auf die gesamte Anwendung, unabhängig von einzelnen Berechtigungen.
Der bisherige Administrator verfügt danach nur noch über die Berechtigungen, die ihm durch seine Rollen oder direkten Zuweisungen zuteil werden. Da der `Owner`-Status stets nur einem einzelnen Nutzer zugewiesen ist, verliert der bisherige Administrator diesen automatisch mit der Übertragung.
Der Menüpunkt **Administration** ist ausschließlich für den aktuellen Owner sichtbar und erreichbar - alle anderen Nutzer sehen diesen Punkt im Accountbereich nicht, unabhängig von ihren sonstigen Berechtigungen.

## 5.6.1 Durchführung

Zur Übertragung wird der gewünschte neue Owner über die Nutzersuche ausgewählt und die Auswahl mit **übertragen** bestätigt. Dafür ist eine erneute Anmeldung erforderlich (wie z.B. bei der Änderung der Anmeldedaten).
**Wichtig:** Die Übertragung wirkt sofort und kann nicht direkt rückgängig gemacht werden. Der bisherige Administrator kann die Administration nur zurückerhalten, wenn der neue Owner sie ihm erneut überträgt.

# 5.7 Wissensartikel




Über das Buch-Symbol in der Navigationsleiste erreichen Nutzer die Wissensartikel (Wiki-Einträge). Die Übersicht listet alle Artikel, auf die der eigene Account Zugriff hat, durchsuchbar über das Suchfeld nach Titel.

## 5.7.1 Anlegen und Bearbeiten

Über das Plus-Symbol in der Artikelübersicht lässt sich ein neuer Artikel anlegen. Titel und Inhalt werden direkt in einem Rich-Text-Editor eingegeben (Überschriften, Textformatierungen, Listen, Links). Ein neuer Artikel ist direkt nach dem Anlegen automatisch für alle Accounts bearbeitbar und kann anschließend über die Freigaben (siehe unten) gezielt auf einzelne Rollen eingeschränkt werden.
Ein bestehender Artikel öffnet sich mit zwei Reitern:

- **Artikel**: Titel und Inhalt, editierbar für jeden Account mit Bearbeitungsrecht
- **Freigaben**: Sichtbarkeit und Bearbeitbarkeit des Artikels

Ohne Bearbeitungsrecht wird der Inhalt nur lesend angezeigt, der Reiter „Freigaben“ entfällt in diesem Fall.

## 5.7.2 Freigaben

Sichtbarkeit und Bearbeitbarkeit werden für jeden Artikel unabhängig voneinander gesetzt:

- **Für alle sichtbar**: macht den Artikel für alle Accounts lesbar
- **Für alle bearbeitbar**: macht den Artikel für alle Accounts bearbeitbar
- **Rollen**: Rollen können für Sichtbarkeit bzw. Bearbeitbarkeit ausgewählt werden. Ein Account ohne Admin-Rechte kann dabei nur Rollen zuweisen, denen er selbst angehört; Admins sowie der Owner der Anwendung können beliebige Rollen zuweisen. Bereits am Artikel hinterlegte fremde Rollen bleiben davon unberührt und können weiterhin entfernt werden.

Jeder Account mit Bearbeitungsrecht kann die Freigaben jederzeit über den Reiter „Freigaben“ anpassen.
**Hinweis:** Ein Artikel muss immer für mindestens eine Rolle oder für alle bearbeitbar bleiben. Das Speichern der Freigaben wird verweigert, wenn dadurch niemand außer dem erstellenden Account mehr bearbeiten könnte. So geht das hinterlegte Wissen nicht verloren, wenn der ursprüngliche Account gelöscht wird oder ausscheidet.
Der Eigentümer (Owner) der Anwendung sieht und bearbeitet zusätzlich automatisch alle Artikel, die nicht ausschließlich für den erstellenden Account bestimmt sind - also alle Artikel mit Rollen- oder „für alle“-Freigabe.

### 5.7.3 Löschen

Wissensartikel lassen sich über den Button „Artikel löschen“ entfernen. Da dies unwiderruflich ist, muss die Löschung in einem zusätzlichen Dialog bestätigt werden.

# 6 Ökosystem FF Admin

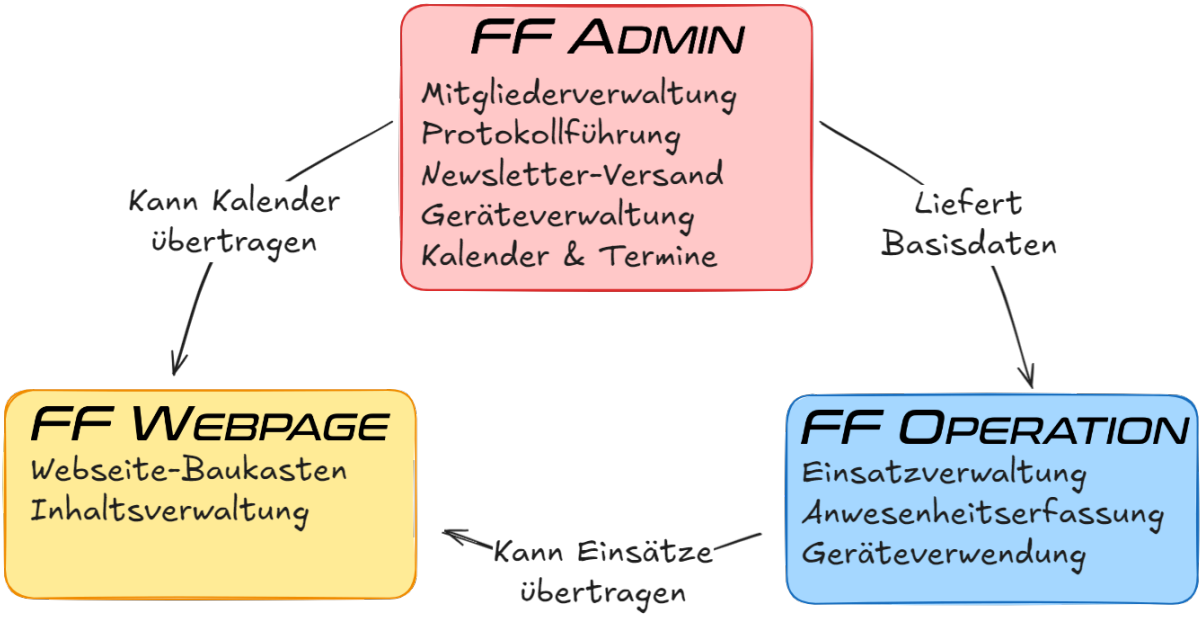



Abbildung 134: Übersicht zum Ökosystem

Das **FF Ökosystem** bietet eine zentrale Lösung für die Verwaltung von Mitgliedern, Einsätzen und Inhalten.

**FF Admin** ist das Herzstück mit der Mitglieder- und Geräteverwaltung sowie der Module zur Wehr- und Vereinsverwaltung. Es stellt Stammdaten bereit und ermöglicht mit allen seinen Modulen die effiziente Verwaltung einer Feuerwehr oder Vereins.

**FF-Webpage** erleichtert die Erstellung und Verwaltung von Webseiteninhalten. Kalender- bzw. Einsatzdaten können von FF Admin und FF Operation an die Webseite geschickt und dort veröffentlichen werden, um Mitglieder und die Öffentlichkeit auf dem Laufenden zu halten.

**FF Operation** unterstützt das Einsatzmanagement, die Anwesenheitserfassung und den Geräteeinsatz. Es kann die Mitgliederdaten, Gerätschaften und co aus FF Admin beziehen.

# 7 Roadmap

Folgende Funktionalitäten sind für FF Admin in Planung (Auszug):

- **Calendar Link Dictionary:** Speicherung von Kalender-Link-Configs mit Aliase wie eine Name.
- **Finanzen I:** Speicherung und Erstellung von Rechnungen als pdf mit XML-Einbettungen für E-Rechnungen.
- **Finanzen II:** Darstellung der Einnahmen und Ausgaben in einem Graphen.

Eine ausführliche Liste geplanter Funktionen kann im Git unter Issues gefunden werden:
https://code.jk-effects.cloud/FF-Admin/ff-admin/issues
https://code.jk-effects.cloud/FF-Admin/ff-admin-server/issues