



FF OPERATION

Anleitung

bis v1.0.x



Open Source Software
entwickelt durch JK Effects
von Julian Krauser
16. Februar 2025

Inhaltsverzeichnis

| | |
|--|----------|
| Inhaltsverzeichnis | II |
| 1 Einleitung | 1 |
| 2 Installation | 2 |
| 2.1 Docker | 2 |
| 2.2 Docker-Compose | 2 |
| 2.3 Git | 6 |
| 2.4 Konfiguration | 7 |
| 2.5 Update der Version | 10 |
| 2.6 Wechsel des Datenbanksystems | 11 |
| 2.7 WebApp | 11 |
| 2.8 Einrichtung | 11 |

1 Einleitung

FF Operation - Flexibles Einsatzmanagement für Feuerwehren und Vereine

FF Operation ist eine leistungsfähige Einsatzverwaltungssoftware und ein zentraler Bestandteil des FF-Ökosystems. Neben Feuerwehreinsätzen können auch Arbeitseinsätze von Vereinen erfasst werden. Durch den modularen Aufbau und frei definierbare Stammdaten kann die Software flexibel an unterschiedliche Organisationen angepasst werden.

FF Operation ermöglicht darüber hinaus die kollaborative Erstellung von Einsatzberichten. Daten können offline ohne Login erfasst und später nahtlos in ein Online-Formular übertragen werden, was eine flexible und effiziente Dokumentation gewährleistet.

Wird FF Admin eingesetzt, kann es als zentrale Datenquelle für Basisinformationen dienen und ein zentrales Login-Management übernehmen.

2 Installation

FF Operation 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 Operation 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. Der <tag> des Images kann entweder `latest` für die neueste Version oder `vx.y.z` für eine bestimmte Version sein. Die Versionen können auch in den Releases der Repositories der Anwendungen nachgeschlagen werden. Dort lassen sich auch Informationen zu neuen Funktionen, Änderungen oder Fehlerbehebungen der jeweiligen Funktion finden.

2.2 Docker-Compose

App

```
1  ff-operation-app:
2    image: docker.registry.jk-effects.cloud/ehrenamt/ff-operation/app:<version>
3    container_name: ff_operation
4    restart: unless-stopped
5    ports:
6      - "80:80"
7    environment:
8      - SERVERADDRESS=<backend_url>
9      - APPNAMEOVERWRITE=<appname>
10     - IMPRINTLINK=<imprint link>
11     - PRIVACYLINK=<privacy link>
12     - CUSTOMLOGINMESSAGE=<betrieben von xy>
13    volumes:
14      - <volume|local path>/favicon.ico:/usr/share/nginx/html/favicon.ico
15      - <volume|local path>/favicon.png:/usr/share/nginx/html/favicon.png
16      - <volume|local path>/Logo.png:/usr/share/nginx/html/Logo.png
```



Anleitung zu FF Operation bis v1.0.x – Installation

Die Verwendung der Werte des Typs Environment werden unter dem Punkt Konfiguration ([Abschnitt 2.4](#)) erklärt.

Alle Environment Werte sind Optional und haben Standard-Werte.

Ist ein Wert optional und hat keinen Fallback, so wird in der Anwendung nichts angezeigt.

Die Volumes dienen zur erweiterten Personalisierung der App mit eigenem Logo der Feuerwehr oder des Vereins. Hiervon betroffen ist das Icon im Browser-Tab, jede Anzeige des FF Operation Logos innerhalb der App und das Icon, wenn die WebApp auf einem Gerät installiert wird.

Die Konfiguration der Volumes ist optional, falls Sie die Standard-Logos verwenden wollen.

Ein Teil der Logos haben eine Anforderung an die Auflösung:

| Icon | Auflösung | Anzeigeort |
|-------------|------------|------------------------------|
| favicon.ico | 48x48 px | Browser-Tab Icon |
| favicon.png | 512x512 px | WebApp Icon zur Installation |
| Logo.png | beliebig | Innerhalb der Anwendung |

Die Dateien müssen exakt gleich geschrieben sein. Achten Sie deshalb auf Schreibfehler und Groß-/Kleinschreibung.

Anleitung zu FF Operation bis v1.0.x – Installation

Server

```
1  ff-operation-server:
2      image: docker.registry.jk-effects.cloud/ehrenamt/ff-operation/server:<version>
3      container_name: ff_operation_server
4      restart: unless-stopped
5      ports:
6          - "5000:5000"
7      environment:
8          - DB_TYPE=<database type>
9          - DB_HOST=<database host>
10         - DB_PORT=<database port>
11         - DB_NAME=<database name>
12         - DB_USERNAME=<database username>
13         - DB_PASSWORD=<database password>
14         - JWT_SECRET=<jwt secret>
15         - JWT_EXPIRATION=<jwt expiration>
16         - REFRESH_EXPIRATION=<refresh expiration>
17         - PWA_REFRESH_EXPIRATION=<pwa refresh expiration>
18         - MAIL_USERNAME=<mailadress|username>
19         - MAIL_PASSWORD=<mail password>
20         - MAIL_HOST=<mail server url>
21         - MAIL_PORT=<port>
22         - MAIL_SECURE=<boolean>
23         - CLUB_NAME=<club name>
24         - CLUB_WEBSITE=<club website>
25         - BACKUP_INTERVAL=<backup interval>
26         - BACKUP_COPIES=<backup parallel copies>
27         - BACKUP_AUTO_RESTORE=<boolean>
28         - USE_SECURITY_STRICT_LIMIT=<boolean>
29         - SECURITY_STRICT_LIMIT_WINDOW=<time window>
30         - SECURITY_STRICT_LIMIT_REQUEST_COUNT=<strict_request_count>
31         - USE_SECURITY_LIMIT=<boolean>
32         - SECURITY_LIMIT_WINDOW=<time window>
33         - SECURITY_LIMIT_REQUEST_COUNT=<request_count>
34         - TRUST_PROXY=<proxy config>
35      volumes:
36          - <volume|local path>:/app/files
```

YAML

Die Verwendung der Werte des Typs Environment werden unter dem Punkt Konfiguration (Abschnitt 2.4) 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 der Anwendung.

Innerhalb dem Ordner, der dem Volume zugeordnet ist, werden Backups und Ausdrucke der

Anleitung zu FF Operation bis v1.0.x – Installation

geschriebenen Protokolle und Newsletter abgelegt.

Datenbank

Als Datenbank können MySQL, Postgres und SQLite verwendet werden. Postgres wird für den Produktiven Einsatz empfohlen.

Konfiguration von MySQL:

```
1 ff-db:
2   image: mariadb:<version (bsp 11.2)>
3   container_name: ff_db
4   restart: unless-stopped
5   ports:
6     - "3306:3306"
7   environment:
8     - MYSQL_DATABASE=<database name>
9     - MYSQL_USER=<username>
10    - MYSQL_PASSWORD=<user password>
11    - MYSQL_ROOT_PASSWORD=<root password>
12   volumes:
13     - <volume|local path>:/var/lib/mysql
```

YAML

MYSQL_USER und MYSQL_PASSWORD sind optional. Werden diese nicht gesetzt, kann der Server entweder mit dem Nutzer root und dem gesetzten MYSQL_ROOT_PASSWORD Zugang zur Datenbank erhalten, oder es wird im nachhinein ein Nutzerzugang erstellt, der Zugriff auf die erstellte Datenbank hat. MYSQL_DATABASE erstellt direkt eine Datenbank, die durch einen angelegten MYSQL_USER verfügbar ist.

Konfiguration von Postgres:

```
1 ff-db:
2   image: postgres:<version (bsp 16)>
3   container_name: ff_db
4   restart: unless-stopped
5   ports:
6     - "5432:5432"
7   environment:
8     - POSTGRES_DB=<database name>
9     - POSTGRES_USER=<username>
10    - POSTGRES_PASSWORD=<user password>
11   volumes:
12     - <volume|local path>:/var/lib/postgresql/data
```

YAML

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 ange-

Anleitung zu FF Operation bis v1.0.x – Installation

legt werden. Dadurch ist das Veröffentlichen der Datenbank-Port-Exposes nicht mehr notwendig. Das Entfernen der port-Exposes verhindert den direkten Zugriff auf die Ports von außerhalb. Ergänzt muss hierfür das `network` und die Teilhabe des Backend-Containers am Netzwerk:

1. Ergänzung zu Server und Datenbank:

```
1 networks:
2   - ff_internal
```

2. Ergänzung zur finalen Compose:

```
1 networks:
2   ff_internal:
```

3. Optionale Ergänzung zum Server:

```
1 depends_on:
2   - ff-db
```

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

2.3 Git

Eine Alternative zu Docker ist die direkte Ausführung der Anwendungen auf dem Server oder Desktop Gerät.

Hierzu müssen die App und der Server als Quellcode auf das System geladen und dort direkt verwendet werden.

Die Veröffentlichung der App und des Servers, damit diese aus dem Internet erreichbar sind, muss gesondert eingerichtet werden.

Das System muss NodeJs und die bevorzugte Datenbank installiert haben.

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

Die NodeJs Prozesse können auch durch Tools wie pm2 verwaltet werden.

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.

Anleitung zu FF Operation bis v1.0.x – Installation

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.

App

```
1 git clone https://forgejo.jk-effects.cloud/Ehrenamt/ff-operation.git
2 cd ff-operation
3 npm install
4 npm run build
```

Shell

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

Server

```
1 git clone https://forgejo.jk-effects.cloud/Ehrenamt/ff-operation-server.git
2 cd ff-operation-server
3 npm install
4 npm run build
5 npm run start
```



















Shell











2.4 Konfiguration









Folgende Werte können zu einem Container konfiguriert werden:

| Variable | Zweck | Fallback | optional |
|--------------------|---|--------------|----------|
| 📌 App-Variablen | | | |
| SERVERADDRESS | URL, über welche das Backend erreicht werden kann. Die URL muss mit <code>http://</code> oder <code>https://</code> starten und darf keinen Pfad beinhalten. Wenn das Backend auf der gleichen URL wie die App läuft, kann diese Variable weggelassen werden. | | ✓ |
| APPNAMEOVERWRITE | Anzeige eines anderen Namens als FF Operation. | FF Operation | ✓ |
| IMPRINTLINK | Link zum Impressum des Betreibers. | | ✓ |
| PRIVACYLINK | Link zur Datenschutzerklärung des Betreibers. | | ✓ |
| CUSTOMLOGINMESSAGE | Nachricht auf der Login-Seite. (Bsp.: betrieben von xy) | | ✓ |
| | | | |

Anleitung zu FF Operation bis v1.0.x – Installation

| Variable | Zweck | Fallback | optional |
|--|---|--------------|---|
|  Server-Variablen | | | |
| DB_TYPE | Folgende Datenbanktypen sind verfügbar: mysql, sqlite, postgres | mysql |  |
| DB_HOST | URL zur Datenbank oder Dateipfad zur SQLite-Datenbank | |  |
| DB_PORT | Port der Datenbank | 3306 |  |
| DB_NAME | Name der Datenbank in welcher die Tabellen erstellt werden. | |  |
| DB_USERNAME | Nutzername für Zugang zu Datenbank | |  |
| DB_PASSWORD | Passwort zum Zugang zur Datenbank | |  |
| JWT_SECRET | Zufällige Zeichenkette zur Validierung der Session-Tokens. | |  |
| JWT_EXPIRATION | Gültigkeitsdauer eines Session-Tokens. Format: [0-9]*(y d h m s) | 15m |  |
| REFRESH_EXPIRATION | Gültigkeitsdauer eines Logins nach letzter Nutzung der App im Browser Format: [0-9]*(y d h m s) | 1d |  |
| PWA_REFRESH_EXPIRATION | Gültigkeitsdauer eines Logins nach letzter Nutzung der installierten App Format: [0-9]*(y d h m s) | 5d |  |
| MAIL_USERNAME | Nutzername oder Mailadresse | |  |
| MAIL_PASSWORD | Passwort zum Nutzernamen oder der Mailadresse | |  |
| MAIL_HOST | URL des Mailservers | |  |
| MAIL_PORT | Port des Mailservers für Versand (SMTP). Ports sind 25, 465, 587 | 587 |  |
| MAIL_SECURE | Soll eine Secure Verbindung aufgebaut werden. Muss true sein bei Port 465. | false |  |
| CLUB_NAME | Wird für TOTP Titel verwendet. | FF Operation |  |
| CLUB_WEBSITE | | |  |

| Variable | Zweck | Fallback | optional |
|-------------------------------------|--|----------|---|
| BACKUP_INTERVAL | Wie viele Tage Abstand sollen zwischen Backups liegen? (min. 1) | 1 |  |
| BACKUP_COPIES | Wie viele parallele Kopien von Backups sollen parallel Verfügbar sein? (min. 1) | 7 |  |
| BACKUP_AUTO_RESTORE | Soll das neueste Backup bei Server-Start automatisch geladen werden, wenn die Datenbank als leer erkannt wird? | true |  |
| USE_SECURITY_STRICT_LIMIT | Soll ein Anfrage-Limit für Login, Reset und Co gesetzt werden? In diesem Fall ist der Nutzer nicht angemeldet, sondern versucht es. | true |  |
| SECURITY_STRICT_LIMIT_WINDOW | Über welches Zeitfenster soll das Limit angewandt werden? 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? In diesem Fall ist der Nutzer angemeldet. | true |  |
| SECURITY_LIMIT_WINDOW | Über welches Zeitfenster soll das Limit angewandt werden? 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? Ist dieser Wert nicht gesetzt, wird davon ausgegangen, dass kein Proxy verwendet wird. Folgende Werte können gesetzt | |  |

| Variable | Zweck | Fallback | optional |
|--|---|----------|---|
| | werden: true / false Anzahl der Proxies: [0-9]* IP-Adresse des Proxy: ip IP-Adressen der Proxy: ip1,ip2,... | | |
|  Database-Variablen | | | |
| MYSQL_DATABASE | Name der Datenbank, die bei Erstellung direkt angelegt wird. | |  |
| MYSQL_USER | Benutzername des Users, der bei Erstellung direkt angelegt wird. | |  |
| MYSQL_PASSWORD | Passwort zum User, das bei Erstellung gesetzt wird. | |  |
| MYSQL_ROOT_PASSWORD | Passwort für den User root, das bei Erstellung gesetzt wird. | |  |
| | | | |
| 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!

: Bei Verwendung von SQLite sind diese Variablen nicht notwendig!

Hinweis: Eine fehlerhafte Konfiguration der optionalen oder geforderten Variable verhindert das Starten der Anwendung.

Hinweis: Eine Änderung der Datenbank übernimmt die Daten nur automatisch in die neue Datenbank, wenn BACKUP_AUTO_RESTORE aktiviert ist und ein Backup angelegt ist. Es werden dann die Daten des gefundenen Backups in die neue Datenbank eingefügt.

2.5 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 `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 `Benutzer > 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.

2.6 Wechsel des Datenbanksystems

`AUTO RESTORE` ist standardmäßig aktiviert und ermöglicht die automatische Wiederherstellung des letzten Backups - sofern vorhanden - wenn die Datenbank beim Start des Servers leer ist.

Dies vereinfacht auch den Wechsel zwischen Datenbanken, da nur eine neue Verbindung zu einer anderen Datenbank aufgebaut werden muss und die Daten nach dem Neustart des Servers automatisch in die neue leere Datenbank übertragen werden.

2.7 WebApp

FF Operation ist als WebApp verfügbar. Dadurch lässt sich die Anwendung auf einem Smartphone oder Desktop über den Browser installieren.

2.8 Einrichtung

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

1. **Admin Benutzer erstellen:** Erstellen Sie einen Admin Benutzer unter dem Pfad `/setup`, um auf die Mitgliederverwaltung Zugriff zu erhalten. Nach der Erstellung des ersten Benutzers wird der Pfad automatisch geblockt.
2. **Rollen und Berechtigungen:** Unter `Benutzer > Rollen` können die Rollen und Berechtigungen für die Benutzer erstellt und angepasst werden.
3. **Nutzer einladen:** Unter `Benutzer > Benutzer` können weitere Nutzer eingeladen werden. Diese erhalten dann eine E-Mail mit einem Link, um ein TOTP zu erhalten.