



FF WEBPAGE

Anleitung

bis v1.2.x



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

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Abkürzungsverzeichnis / Glossar	III
1 Einleitung	1
2 Installation	2
2.1 Docker	2
2.1.1 Docker-Compose	2
2.2 Git	5
2.3 Konfiguration	6
2.4 Update der Version	7
3 Strapi	8
3.1 Einrichtung	8
3.2 Modelle	10
3.2.1 Global	10
3.2.2 Startseite	13
3.2.3 Sammlungs Referenz	14
3.2.4 Seiten	15
3.2.5 Artikel	16
3.2.6 Einsätze	17
3.2.7 Fahrzeuge	18
3.2.8 Termine:	19
3.2.9 User	19
3.3 Verwendung	20
3.3.1 Dynamic Zone	20
3.3.2 Shared	20

Abkürzungsverzeichnis / Glossar

Collection Type: Speicher einer Auflistung an Einträgen eines bestimmten Typs.

Single Type: Speicher eines einzelnen Eintrags eines bestimmten Typs.

Slug: Identifikation eines Eintrags über den Wert eines Feldes, welcher einzigartig ist. Slugs können automatisiert erstellt werden.

1 Einleitung

FF Webpage - Der flexible Webseitenbaukasten für Feuerwehren und Vereine

FF Webpage ist ein modularer Webseitenbaukasten, bestehend aus einem Nuxt-Frontend und Strapi als CMS. Struktur und Navigation der Seite können über das Strapi CMS individuell konfiguriert werden, was eine flexible Gestaltung ermöglicht.

Speziell entwickelte Module erleichtern die Darstellung von Listen, wobei vordefinierte Listen für Einsätze, Termine und Artikel zur Verfügung stehen. Ein besonderes Feature ist die direkte Übernahme von Terminen aus FF Admin in die Website, wodurch eine nahtlose Integration in das FF-Ökosystem gewährleistet ist.

2 Installation

FF Webpage 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 Webpage 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.

2.1.1 Docker-Compose

Webseite

```
1 ff-landingpage:
2   image: docker.registry.jk-effects.cloud/ehrenamt/ff-webpage/frontend:<version>
3   container_name: ff-landingpage
4   restart: unless-stopped
5   environment:
6     - NUXT_STRAPI_URL=<cms_url>
7     - NUXT_PUBLIC_STRAPI_URL=<cms_url>
8     - NUXT_PUBLIC_APP_TITLE=<website_title>
9   ports:
10    - "3000:3000"
11   volumes:
12    - <volume|local_path>/favicon.png:/app/.output/public/favicon.png
```

YAML

Die Verwendung der Werte des Typs Environment werden unter dem Punkt Konfiguration erklärt.

Das Volume dient zur erweiterten Personalisierung der Webseite mit eigenem Logo der Feuerwehr oder des Vereins. Hiervon betroffen ist nur Icon im Browser-Tab.

Die Konfiguration der Volumes ist optional, falls Sie die Standard-Logos verwenden wollen. Die Dateien müssen exakt gleich geschrieben sein. Achten Sie deshalb auf Schreibfehler und Groß-/Kleinschreibung.

CMS - Inhaltsverwaltung

```
1  ff-cms:
2    image: docker.registry.jk-effects.cloud/ehrenamt/ff-webpage/cms:<version>
3    container_name: ff_cms
4    restart: unless-stopped
5    environment:
6      - DATABASE_CLIENT=mysql
7      - DATABASE_HOST=ff-db
8      - DATABASE_NAME=ffcms
9      - DATABASE_USERNAME=cms
10     - DATABASE_PASSWORD=<dbuserpasswd>
11     - JWT_SECRET=<tobemodified>
12     - ADMIN_JWT_SECRET=<tobemodified>
13     - APP_KEYS=<tobemodified>,<tobemodified>
14     - API_TOKEN_SALT=<tobemodified>
15     - TRANSFER_TOKEN_SALT=<tobemodified>
16    volumes:
17      - <volume|local path>:/app/public/uploads
18    ports:
19      - "1337:1337"
```

YAML

Die Verwendung der Werte des Typs Environment werden unter dem Punkt Konfiguration 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 Uploads 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
```

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
```

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

Hinweis Wenn eine Docker-Compose Datei verwendet wird, kann zusätzlich ein Netzwerk angelegt 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:

Anleitung zu FF Webpage bis v1.2.x – Installation

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.

2.2 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.

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-Dateien müssen vor der Ausführung von `npm run start` angelegt sein. Bei einer Änderung der Einträge müssen die Anwendungen lediglich neu gestartet werden.

Webseite

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

Shell

Anleitung zu FF Webpage bis v1.2.x – Installation

`npm run start` nutzt das eingebaute Hosting von Nuxt, welches SSR ermöglicht, um die App zu betreiben.

Strapi

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

Shell

2.3 Konfiguration

Folgende Werte können zu einem Container konfiguriert werden:

Variable	Zweck	Fallback	optional
App-Variablen			
NUXT_STRAPI_URL	URL, unter der das CMS erreichbar ist.		✖
NUXT_PUBLIC_STRAPI_URL	URL, unter der das CMS erreichbar ist.		✖
NUXT_PUBLIC_APP_TITLE	Text im Browser-Tab	FF Webpage	✓
Server-Variablen			
DB_TYPE	Folgende Datenbanktypen sind verfügbar: mysql, postgres		✖
DB_HOST	URL zur Datenbank		✖
DB_PORT	Port der Datenbank		✖
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			✖
ADMIN_JWT_SECRET			✓
APP_KEYS			✓
API_TOKEN_SALT			✓
TRANSFER_TOKEN_SALT			✓
Database-Variablen			
MYSQL_DATABASE	Name der Datenbank, die bei Erstellung direkt angelegt wird.		✖

Variable	Zweck	Fallback	optional
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 nicht automatisch die Daten in die neue Datenbank.

2.4 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 im Account des Eigentümers 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.

3 Strapi

Disclaimer: Fehler, die auf der Webseite angezeigt werden, können auf eine fehlerhafte Konfiguration in Strapi hinweisen.

3.1 Einrichtung

Um die Webseite nutzen zu können, müssen die Datenspeicher Global mit der Navigation sowie die Startseite ausgefüllt sein.

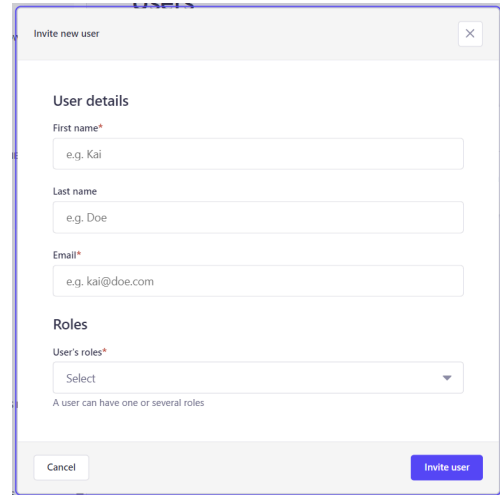
Wichtig ist, dass ein paar Einstellungen vor der ersten Verwendung vorgenommen werden müssen. Dazu zählen:

- Admin Benutzer erstellen:** Erstellen Sie einen Admin-Benutzer, um auf das CMS zugreifen zu können. Dazu muss die URL zum Content Management System geöffnet werden. Dort kann direkt der erste Benutzer als Admin angelegt werden.
- Einstellungen anpassen:** In den Einstellungen müssen unter dem Punkt Users & Permissions Plugin die Rollen und Berechtigungen angepasst werden. Unter Roles > Public müssen die Berechtigungen find, findOne und falls vorhanden distinctYears und findByYear für bestimmte Vorlagen aktiviert werden. Dadurch kann die Website später ohne weitere Einstellungen lesend auf die veröffentlichten Inhalte zugreifen. Die Leseberechtigungen müssen für Article, Collection-lookup, Event, Global, Homepage, Operation, Page, Vehicle und Upload aktiviert werden.
- Medienverwaltung:** Es wird empfohlen, unter Global Settings > Media Library alle Optionen auf true zu setzen. Dadurch wird z.B. auch die automatische Bilddrehung aktiviert. Dadurch werden die Bilder automatisch in die richtige Ausrichtung gedreht und in der Größe optimiert.
- Rollen und Berechtigungen:** Unter Administration Panel > Roles können die Rollen und Berechtigungen für die Benutzer angepasst werden. Es wird empfohlen, die Standardrollen zu

The screenshot shows the Strapi configuration interface. It includes three plugin configuration panels: 'Article' (with checkboxes for create, delete, distinctYears, find, findByYear, findOne, and update), 'Collection-lookup', and 'Event'. Below these is the 'Asset Management' section with toggle switches for 'Responsive friendly upload', 'Size optimization', and 'Auto orientation', all of which are currently set to 'TRUE'.

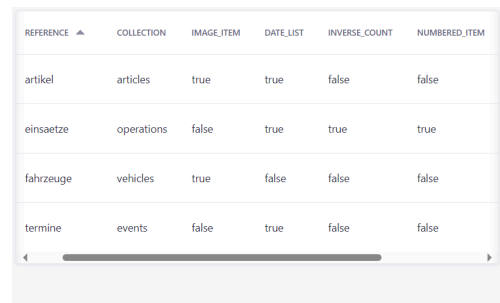
übernehmen und nur die Berechtigungen anzupassen. Diese können jedoch beliebig erweitert werden. Dieser Punkt betrifft nur die Erstellung von Inhalten in der Administrationsoberfläche.

5. **Nutzer einladen:** Unter Administration Panel > Users können weitere Benutzer eingeladen werden. Diese erhalten dann eine E-Mail mit einem Link, über den sie ein Passwort festlegen können. Die eingeladenen Benutzer erhalten dann eine E-Mail, über die sie ihr Passwort setzen können. Rollen können hier direkt bei der Einladung vergeben werden, so dass nach der Registrierung direkt mit der Bearbeitung begonnen werden kann. Theoretisch können Benutzer auch eingeladen und direkt editiert werden, um ein Passwort zu setzen. Dies kann verwendet werden, wenn die E-Mail-Adresse nicht existiert.



6. **Erstellung von Inhalten:** Nachdem die Einstellungen vorgenommen wurden, können die Inhalte erstellt werden. Dazu können die Inhalte über die Schnittstelle zu den bestehenden Modellen erstellt werden.

7. **Einrichtung Sammlungsreferenz:** Die Sammlungsreferenz ist ein eigenes Modell, das die Darstellung von Artikeln, Einsätzen, Fahrzeugen und Terminen regelt. Die Sammlungswerte sind die Verweise auf die verfügbaren Listen. Es wird empfohlen, diese Werte direkt zu definieren, damit sie in anderen Modellen als Referenz verwendet werden können.



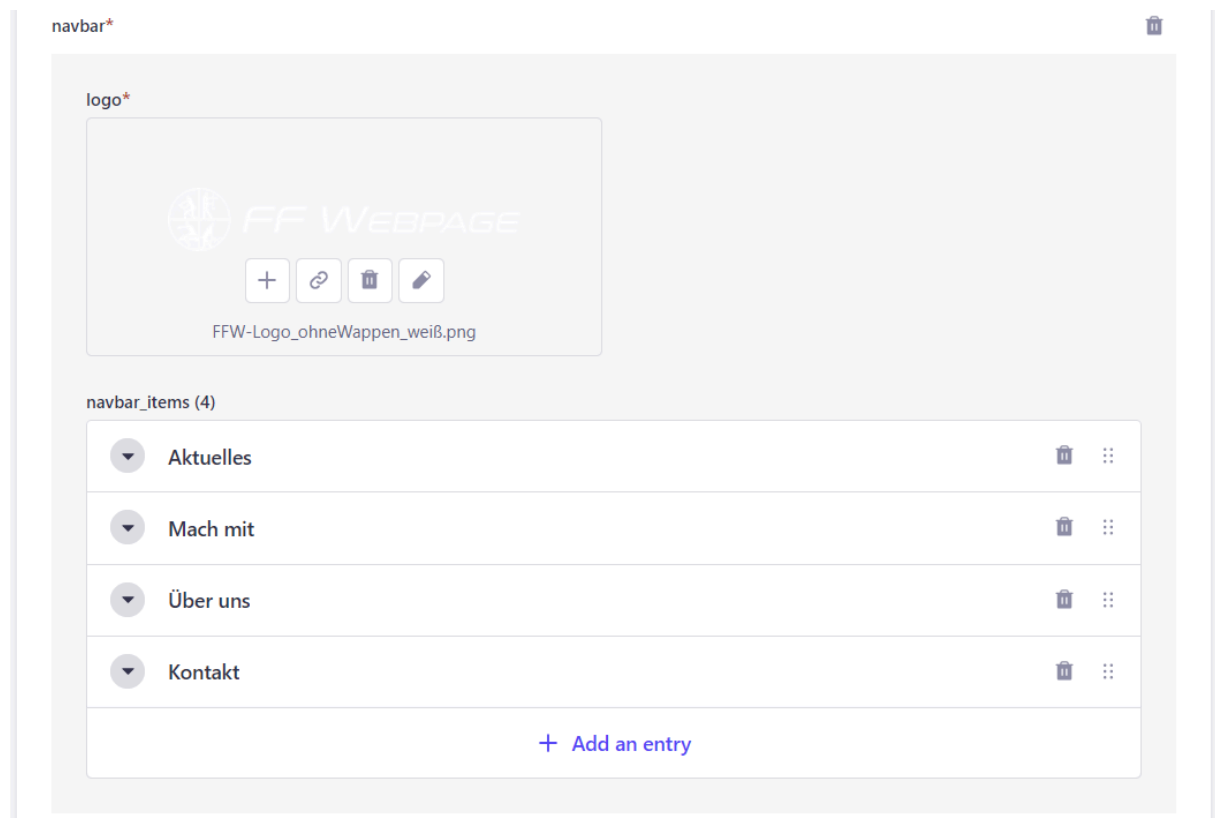
REFERENCE	COLLECTION	IMAGE_ITEM	DATE_LIST	INVERSE_COUNT	NUMBERED_ITEM
artikel	articles	true	true	false	false
einsaetze	operations	false	true	true	true
fahrzeuge	vehicles	true	false	false	false
termine	events	false	true	false	false

3.2 Modelle

3.2.1 Global

Im **Single Type Global** können Werte hinterlegt werden, die im Kontext der gesamten Website zur Verfügung stehen sollen. Diese Einträge umfassen die Navigationsleiste, die Fußzeile und SEO, welche die Anzeige in einer Suchmaschine steuern. Diese Werte müssen definiert werden, damit die Webseite funktionsfähig angezeigt werden kann.

Navigation



Die Navigationsleiste kann mit einem Logo und eigenen Links definiert werden. Die Links können entweder eigenständig oder mit Sublinks erstellt werden. Außerdem kann ein Top-Level-Link direkt auf eine Seite oder auf einen Sub-Link verweisen. Die Links erhalten immer eine Bezeichnung und einen Text, wie der Link in der URL aussehen soll. Jedem Link, der etwas anzeigen soll, muss eine Seite zugeordnet werden. Das Anlegen von Seiten wird in [Abschnitt 3.2.4](#) erklärt.

Anleitung zu FF Webpage bis v1.2.x – Strapi

Links in der Navigation ohne untergeordneten Links:



Top-Level-Links ohne Unterlinks werden in der Hauptnavigationsleiste angezeigt. Diese Links verweisen auf eine zugeordnete Seite. Die angelegten Seiten können über das angezeigte Dropdown-Menü ausgewählt werden.

Links in der Navigation mit untergeordneten Links:



Top-Level-Links mit Unterlinks werden in der Hauptnavigationsleiste angezeigt. Wenn ein Link ausgewählt wird, werden die erzeugten Sublinks angezeigt. Je nach Konfiguration des Links wird dann entweder direkt eine Seite oder eine Seite des Sublinks aufgerufen. Beim Verweis auf Sublinks muss der Verweis

default_active_child mit dem Wert der URL des Sublinks übereinstimmen.

This screenshot shows the configuration page for a link named 'Mach mit'. The 'name' field is 'Mach mit' and the 'URL' field is 'mach-mit'. The 'default_active_child' field is empty. The 'page (1)' dropdown is set to 'Add relation'. A 'Published' button is visible. The 'nav_bar_sub_items (0)' section is empty, with a message 'No entry yet. Click to add one.' and a plus icon.This screenshot shows the configuration page for a link named 'Aktuelles'. The 'name' field is 'Aktuelles' and the 'URL' field is 'aktuelles'. The 'default_active_child' field is 'einsaetze'. The 'page (1)' dropdown is set to 'Add relation'. A 'Published' button is visible. The 'nav_bar_sub_items (1)' section shows a sublink named 'Einsätze' with its own configuration fields and a 'Published' button. A '+ Add an entry' button is at the bottom.

Fußleiste

This screenshot shows the configuration page for the footer. It has two input fields: 'copyright' with the value 'JK Effects' and 'designed_developed_by' with the value 'Julian Krauser'. Below these is a section titled 'links (4)' containing a list of four links: 'Impressum', 'Datenschutz', 'Codebase', and 'Entwickler'. Each link has a dropdown arrow on the left and a trash icon with a three-dot menu on the right. At the bottom of the list is a '+ Add an entry' button.

Die Fußzeile kann mit Verweisen auf den Betreiber und eigenen Links definiert werden. Die Links können entweder auf eine in den Seiten definierte URL ([Abschnitt 3.2.4](#)) oder auf externe Webseiten verweisen. Um auf eine definierte Seite zu verweisen, muss nur der Pfad angegeben werden. Ein Pfad kann mit einem / beginnen oder ein ganzer Pfad sein (aktuelles/einsaetze). Soll ein Link auf eine externe Seite verweisen, muss dem Link `https://` oder `http://` vorangestellt werden.

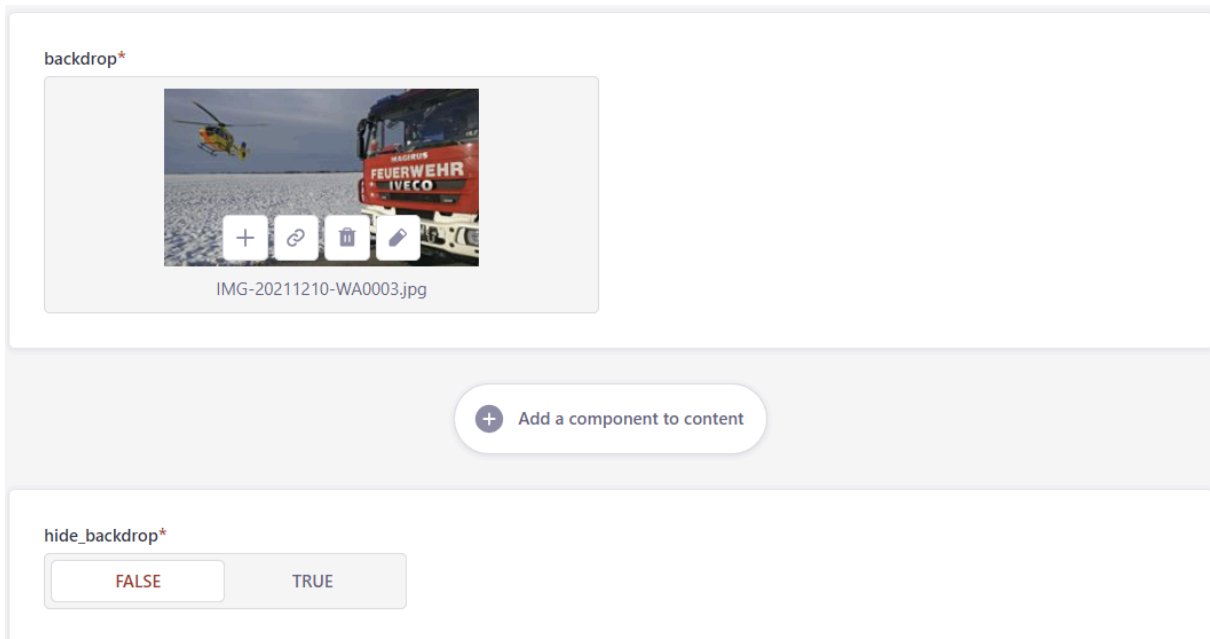
SEO (Search Engine Optimization)



The screenshot shows the 'SEO' section of the Strapi interface. It contains three input fields: 'metaTitle*' with the value 'FF Webpage Demo' and a character limit of 'max. 60 characters'; 'metaDescription*' with the value 'Demo-Seite der FF Webpage mit Beispielen für Inhal:' and a character limit of 'min. 50 characters'; and 'keywords' which is currently empty.

Der Abschnitt SEO ist optional und kann verwendet werden, um den Titel der Registerkarte und die Beschreibung der Seite festzulegen. Dies optimiert die Ergebnisse in Suchmaschinen, da u.a. die Beschreibung direkt angezeigt wird.

3.2.2 Startseite



Der **Single Type Startseite** definiert die Startseite. Diese ist immer fest und muss konfiguriert werden, damit die Website funktionsfähig ist.

Die Startseite besteht aus einem Bild, das über der Navigationsleiste angezeigt werden kann. Die Funktionalität, dass etwas über der Navigationsleiste angezeigt wird, kann auch über `hide_backdrop` deaktiviert werden.

Das Logo, das auch auf der Startseite verwendet wird, ist das gleiche, das unter Global (Abschnitt 3.2.1) angegeben wurde.

Die Erstellung von Inhalten mit Hilfe von Komponenten wird unter Verwendung in Abschnitt 3.3 erläutert. Die Startseite ist eine vollwertige Seite wie die Seiten (Abschnitt 3.2.4), jedoch mit erweiterten Darstellungsmöglichkeiten, um die Startseite hervorzuheben.

3.2.3 Sammlungs Referenz

The screenshot shows the 'Collection Reference' configuration form in Strapi. It includes the following fields and options:

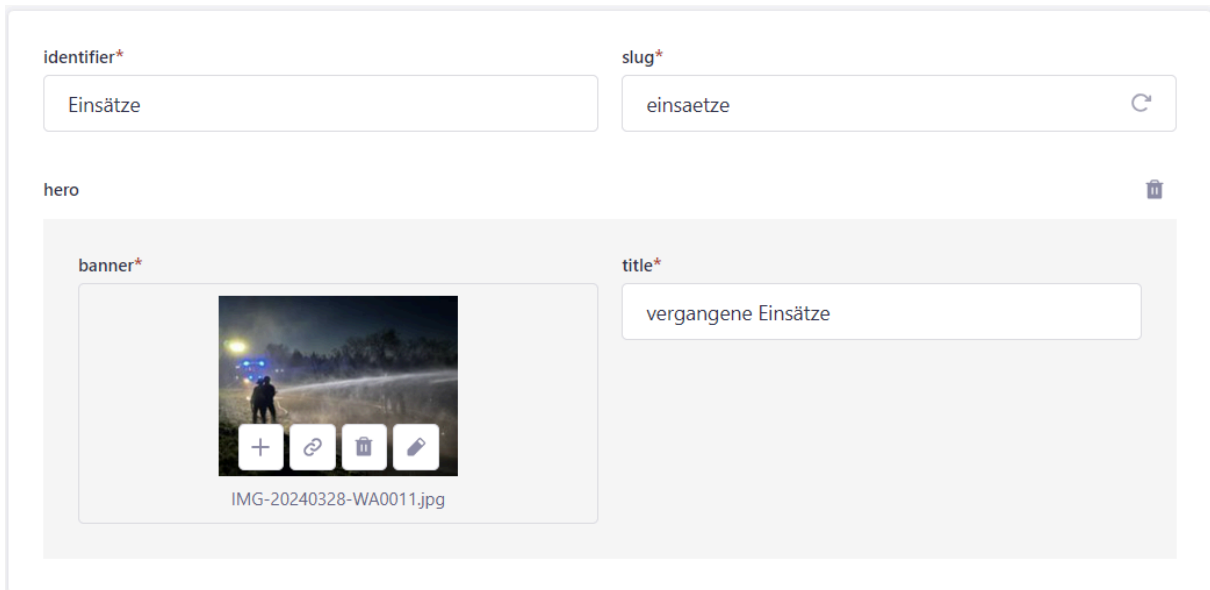
- reference***: A text input field containing 'artikel'.
- collection***: A dropdown menu showing 'articles'.
- image_item***: A toggle switch with 'FALSE' and 'TRUE' options; 'TRUE' is selected.
- date_list***: A toggle switch with 'FALSE' and 'TRUE' options; 'TRUE' is selected.
- numbered_item***: A toggle switch with 'FALSE' and 'TRUE' options; 'FALSE' is selected.
- inverse_count***: A toggle switch with 'FALSE' and 'TRUE' options; 'FALSE' is selected.

Der **Collection Type Collection Reference** definiert, wie die definierten Listen für Artikel, Einsätze, Termine und Fahrzeuge angezeigt werden sollen. Damit wird auch festgelegt, dass der Inhalt bei jeder Verwendung einheitlich ist.

Damit eine Liste wie Artikel, Einsätze, Termine oder Fahrzeuge auf den Seiten verwendet werden kann, muss eine Sammlungsreferenz erstellt worden sein.

- **image_item**: Wenn das Element in der Liste ein Bild anzeigen soll.
- **date_list**: Sollen die Listenelemente nach Jahren gruppiert angezeigt und navigiert werden. Die Elemente werden durch einen Text mit dem Monat getrennt.
- **numbered_item**: Sollen die Listenelemente ohne Bild anstelle des Tages als große Zahl durchnummeriert werden?
- **inverse_count**: Soll die Anzeige umgekehrt werden (neueste oder älteste zuerst). Bei der Verwendung von **numbered_item** wird dadurch die Nummerierung geändert.

3.2.4 Seiten



The screenshot shows the Strapi admin interface for creating a new page. The form is divided into two main sections: 'identifizier*' and 'hero'. The 'identifizier*' section contains two input fields: 'Einsätze' for 'identifizier*' and 'einsaetze' for 'slug*'. The 'hero' section contains a 'banner*' field with a preview image of firefighters at night and a 'title*' field with the text 'vergangene Einsätze'.

Der Collection Type **Seiten** speichert alle Seiten, die erreicht werden können. Derzeit ist jede Seite - ob in der Navigation angegeben oder nicht - über den in der URL definierten **Slug** erreichbar, sofern nicht eine Navigationseinstellung den **Slug** überlagert. Standardmäßig ist die Erreichbarkeit auf <meine Domain>/<slug> gesetzt.

Der Identifier dient nur zur lesbaren Identifizierung der Seite, wenn sie einem Navigationslink zugeordnet ist. Der **Slug** speichert den Pfad, der in der URL angezeigt wird.

Eine Seite kann ein Banner (Hero) haben. Dieses Banner wird direkt unter der Navigation angezeigt. Der Hero kann mit einem Untertitel versehen werden. Ist kein Banner gesetzt, wird direkt unter der Navigation der Kontext der Seite angezeigt.

Der Hauptteil einer Seite kann mit definierten Komponenten erstellt werden. Die Erstellung von Inhalten mit Hilfe von Komponenten wird unter Verwendung in [Abschnitt 3.3](#) erläutert.



3.2.5 Artikel

The screenshot shows the Strapi admin interface for creating a new article. The form is organized into several sections:

- title***: A text input field containing "Übungsdienst".
- slug***: A text input field containing "uebungsdienst" with a refresh icon on the right.
- date***: A date picker showing "07/08/2024" with a close icon on the right.
- description***: A text input field containing "Übung".
- image**: A media library preview showing a landscape photo with a blue sky and trees. Below the image are icons for adding, linking, deleting, and editing. The filename "IMG-20210917-WA0048.jpg" is visible at the bottom.
- attachment**: A large grey area with a blue plus icon and the text "Click to add an asset or drag and drop one in this area".
- content**: A rich text editor with a toolbar containing icons for text, bold, italic, underline, link, unlink, code, and list. Below the toolbar is a large empty text area.

Der **Collection Type Artikel** speichert Beiträge der Website ähnlich wie Artikel in einer Zeitung. Die Art und Weise, wie ein Artikel auf der Website angezeigt wird, wird in der Sammlungsreferenz (Abschnitt 3.2.3) festgelegt.

Jeder Artikel enthält einen Titel, einen URL-Pfad (**Slug**), ein Datum und eine Beschreibung.

Die restlichen Felder werden je nach Konfiguration der Sammlungsreferenz (Abschnitt 3.2.3) benötigt. Content wird z.B. nur benötigt, wenn ein Artikel geöffnet werden kann. Ob ein Artikel geöffnet werden kann, wird unter Verwendung (Abschnitt 3.3) erläutert.

3.2.6 Einsätze

The image shows a Strapi form for a collection type named 'Einsätze'. The form is divided into several sections:

- title***: A text input field containing 'Brand'.
- slug***: A text input field containing 'brand' with a refresh icon on the right.
- date***: A date and time picker showing '04/10/2024' and '02:00'.
- description***: A text input field containing 'Feuer'.
- image**: A large gray box with a blue plus icon and the text 'Click to add an asset or drag and drop one in this area'.
- attachment (1 / 1)**: A box showing a single image attachment. The image is a landscape with a large, dark, billowing cloud of smoke or fire. Below the image are icons for adding, linking, deleting, and editing. The filename 'IMG-20210813-WA0001.jpg' is displayed below the image.
- content**: A rich text editor with a toolbar containing icons for text, bold, italic, underline, link, unlink, code, and list. The editor area is currently empty.

Der **Collection Type Einsätze** speichert vergangene Einsätze ab. Die Art und Weise, wie ein Einsatz auf der Website angezeigt wird, wird in der Sammlungsreferenz ([Abschnitt 3.2.3](#)) festgelegt. Jeder Einsatz enthält einen Titel, einen URL-Pfad (**Slug**), ein Datum mit Uhrzeit und eine Beschreibung.

Die restlichen Felder werden je nach Konfiguration der Sammlungsreferenz ([Abschnitt 3.2.3](#)) benötigt. Content wird z.B. nur benötigt, wenn ein Einsatz geöffnet werden kann. Ob ein Einsatz geöffnet werden kann, wird unter Verwendung ([Abschnitt 3.3](#)) erläutert.

3.2.7 Fahrzeuge

The screenshot shows the Strapi content editor interface for a collection type named 'Fahrzeuge'. The form is organized into several sections:

- title***: A text input field containing 'Feuerwehrboot'.
- slug***: A text input field containing 'feuerwehrboot' with a refresh icon on the right.
- date**: A date picker showing a calendar icon and a time selection dropdown set to '--:--'.
- description***: A text input field containing 'Boot'.
- image**: A media library preview showing a photo of a fireboat on water. Below the image are icons for adding, linking, deleting, and editing. The filename 'IMG-20210719-WA0005.jpg' is visible.
- attachment**: A large gray area with a blue plus icon and the text 'Click to add an asset or drag and drop one in this area'.
- content**: A rich text editor with a toolbar containing icons for text, bold, italic, underline, link, unlink, code, and list. The editor area is currently empty.

Der **Collection Type Fahrzeuge** speichert Fahrzeuge des Fuhrparks ab. Die Art und Weise, wie ein Fahrzeug auf der Website angezeigt wird, wird in der Sammlungsreferenz ([Abschnitt 3.2.3](#)) festgelegt.

Jeder Einsatz enthält einen Titel, einen URL-Pfad (**Slug**) und eine Beschreibung.

Die restlichen Felder werden je nach Konfiguration der Sammlungsreferenz ([Abschnitt 3.2.3](#)) benötigt oder nicht angezeigt, wenn diese optional sind. Content wird z.B. nur benötigt, wenn ein Fahrzeug geöffnet werden kann. Ob ein Fahrzeug geöffnet werden kann, wird unter Verwendung ([Abschnitt 3.3](#)) erläutert.

3.2.8 Termine:

The screenshot shows the Strapi admin interface for the 'Termine' collection type. The form is divided into several sections:

- title***: A text input field containing 'Jahreshauptversammlung'.
- slug***: A text input field containing 'jahreshauptversammlung' with a refresh icon.
- date***: A date and time picker showing '01/06/2025' and '01:00'.
- description***: A text input field containing 'Jahreshauptversammlung der Feuerwehr'.
- image**: A placeholder area with a blue plus icon and the text 'Click to add an asset or drag and drop one in this area'.
- attachment**: A placeholder area with a blue plus icon and the text 'Click to add an asset or drag and drop one in this area'.
- content**: A rich text editor with a toolbar containing icons for text, bold, italic, underline, link, unlink, code, and list.

Der **Collection Type Termine** speichert vergangene oder anstehende Termine. Die Art und Weise, wie ein Termin auf der Website angezeigt wird, wird in der Sammlungsreferenz ([Abschnitt 3.2.3](#)) festgelegt.

Jeder Termin enthält einen Titel, einen URL-Pfad (**Slug**), ein Datum mit Uhrzeit und eine Beschreibung.

Die restlichen Felder werden je nach Konfiguration der Sammlungsreferenz ([Abschnitt 3.2.3](#)) benötigt. Content wird z.B. nur benötigt, wenn ein Termin geöffnet werden kann. Ob ein Termin geöffnet werden kann, wird unter Verwendung ([Abschnitt 3.3](#)) erläutert.

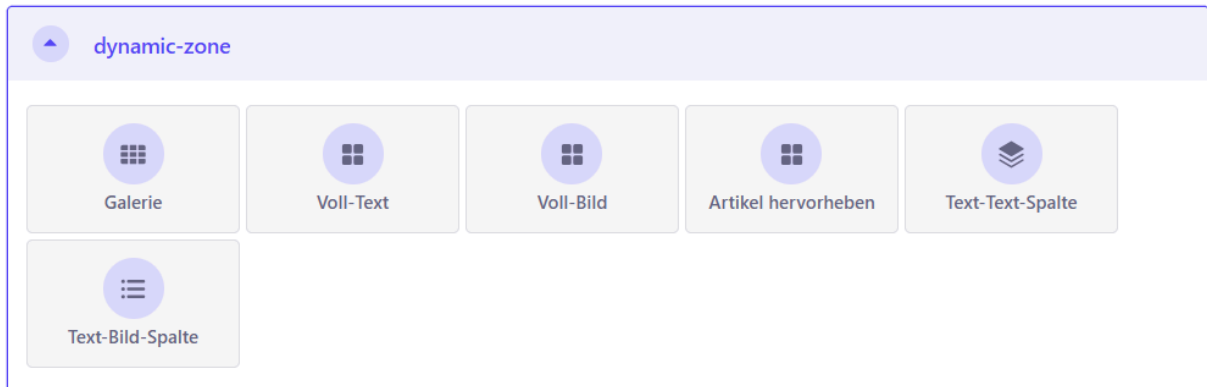
3.2.9 User

Die Verwendung des **Collection Type User** ist nicht erforderlich, da diese nur benötigt wird, wenn sich Besucher der Website auf der Website anmelden könnten.

3.3 Verwendung

Es gibt 2 Arten an Komponenten, welche zur Darstellung von Inhalten verwendet werden können.

Komponenten der `dynamic-zone` werden verwendet, um Inhalte direkt hinzuzufügen.



Komponenten aus `shared` werden verwendet, um Inhalte aus anderen Listen verwenden zu können.



3.3.1 Dynamic Zone

Galerie

Voll-Text

Voll-Bild

Text-Text-Spalte

Text-Bild-Spalte

3.3.2 Shared

Artikel hervorheben

Inhaltsauswahl