



# *FF WEBPAGE*

## Anleitung

bis v1.3.x



Open Source Software  
entwickelt durch JK Effects  
von Julian Krauser  
22. April 2025

# Inhaltsverzeichnis

Inhaltsverzeichnis .....	II
Abkürzungsverzeichnis / Glossar .....	III
1 Einleitung .....	1
2 Installation .....	2
2.1 Docker .....	2
2.2 Docker-Compose .....	2
2.3 Git .....	5
2.4 Konfiguration .....	6
2.5 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 .....	13
3.2.4 Seiten .....	15
3.2.5 Artikel .....	17
3.2.6 Einsätze .....	18
3.2.7 Fahrzeuge .....	19
3.2.8 Termine: .....	20
3.2.9 User .....	20
3.3 Verwendung .....	21
3.3.1 Dynamic Zone .....	21
3.3.2 Shared .....	23

# 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.

### 2.2 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
```



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

## Anleitung zu FF Webpage bis v1.3.x – Installation

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=<random string>
12    - ADMIN_JWT_SECRET=<random string>
13    - APP_KEYS=<random string> # oder<random string>,<random string>, ...
14    - API_TOKEN_SALT=<random string>
15    - TRANSFER_TOKEN_SALT=<random string>
16   volumes:
17     - <volume|local path>:/app/public/uploads
18   ports:
19     - "1337:1337"
```

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 v1.3.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.

## 2.3 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.

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 v1.3.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

## 2.4 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.

## 2.5 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 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.

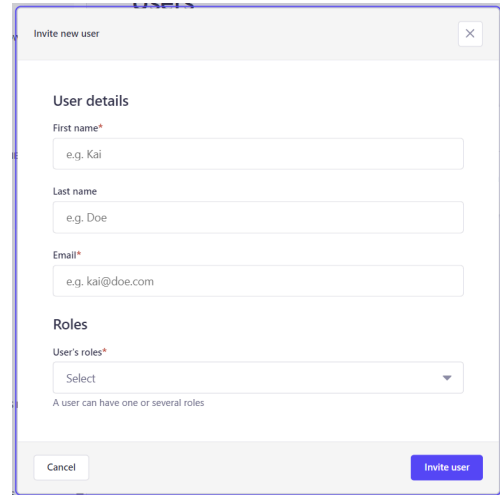
Wichtig ist, dass ein paar Einstellungen vor der ersten Verwendung vorgenommen werden müssen. Dazu zählen:

- 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 und Vehicle 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'.

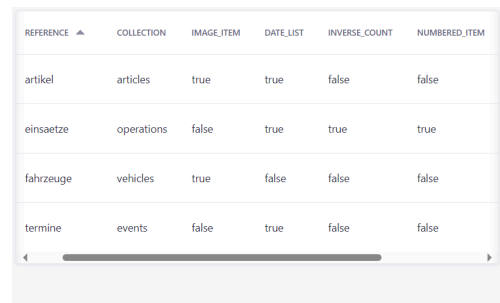
übernehmen und nur die Berechtigungen anzupassen. Diese können jedoch beliebig erweitert werden. Dieser Punkt betrifft nur die Erstellung von Inhalten in der Administrationsoberfläche.

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

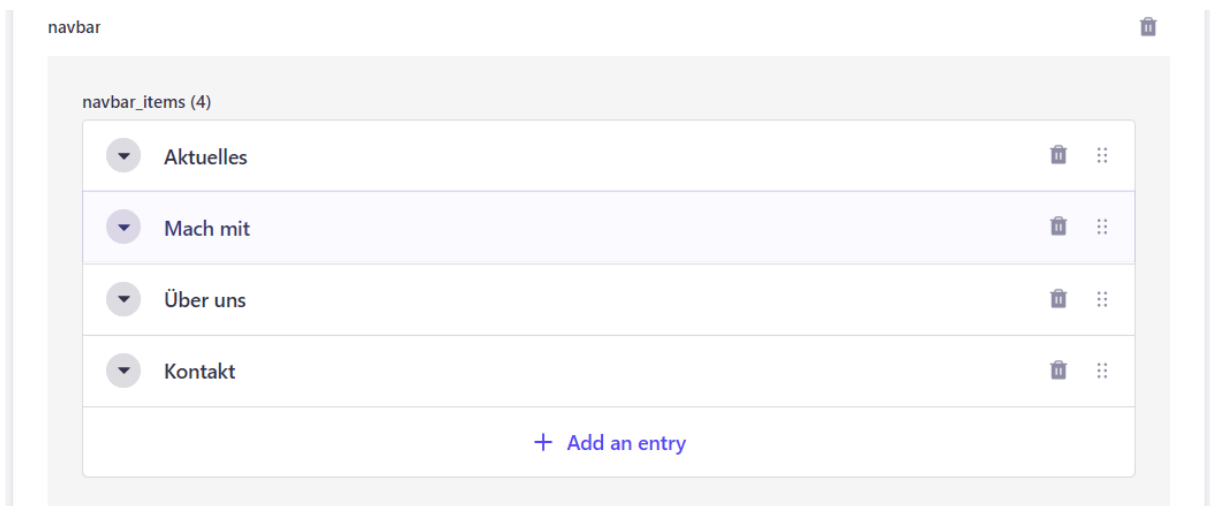
Im **Single Type Global** können Werte hinterlegt werden, die im Kontext der gesamten Website zur Verfügung stehen sollen. Diese Einträge umfassen die Navigationsleiste, die Fußzeile und SEO, welche die Anzeige in einer Suchmaschine steuern. Diese Werte müssen definiert werden, damit die Webseite funktionsfähig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

#### Logo



Das Logo wird global definiert und ersetzt dadurch das Fallback Logo der Webseite. Wird irgendwo ein Bild angezeigt, das nicht angegeben ist, so wird dort auch das Logo als Ersatz angezeigt.

#### 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 Bezeich-

nung 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.

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.

The image shows the Strapi admin interface for configuring 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 menu is set to 'Add relation'. Below the dropdown is a button 'Mach Mit' with a 'Published' status and a close button. At the bottom, there is a message 'No entry yet. Click to add one.' with a plus icon.The image shows the Strapi admin interface for configuring 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 menu is set to 'Add relation'. Below the dropdown is a button 'Einsätze' with a 'Published' status and a close button. At the bottom, there is a button '+ Add an entry'.

## Fußleiste

The screenshot shows the 'footer' configuration section in Strapi. It includes two text input fields: 'copyright' with the value 'JK Effects' and 'maintained\_by' with the value 'Julian Krauser'. Below these is a 'links (4)' section containing a list of four links: 'Impressum', 'Datenschutz', 'Codebase', and 'Entwickler'. Each link has a dropdown arrow on the left and delete and edit icons on the right. At the bottom of the links list is a '+ Add an entry' button.

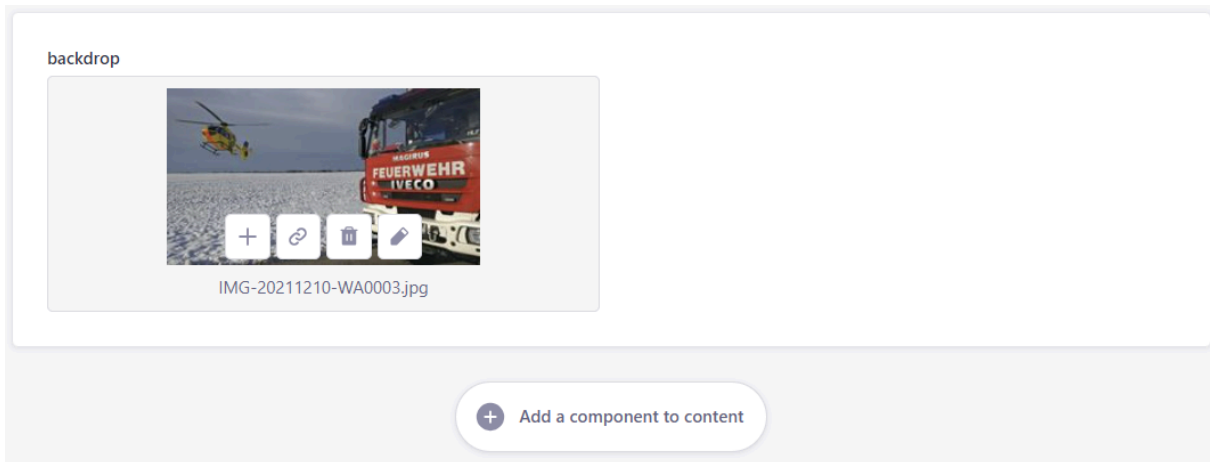
Die Fußzeile kann mit Verweisen auf den Betreiber und eigenen Links definiert werden. Die Links können entweder auf eine in den Seiten definierte URL ([Abschnitt 3.2.4](#)) oder auf externe Webseiten verweisen. Um auf eine definierte Seite zu verweisen, muss nur der Pfad angegeben werden. Ein Pfad kann 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' configuration section in Strapi. It features three input fields: 'metaTitle\*' with the value 'FF Webpage Demo' (with a note 'max. 60 characters' below it), 'metaDescription' with the value 'Demo-Seite der FF Webpage mit Beispielen für Inhal', and 'keywords' which is an empty text area.

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 optional aus einem Bild, das über der Navigationsleiste angezeigt werden kann. Die Funktionalität, dass etwas über der Navigationsleiste angezeigt wird, wird anhand der Füllung des Feldes `backdrop` entschieden. Ist kein Bild angegeben, wird überhalb der Navigationsleiste nichts angezeigt.

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



Der **Collection Type Sammlungs Referenz** definiert global, wie die 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.

- `show_image`: Soll das Element in der Liste ein Bild anzeigen.
- `show_date`: Soll das Element in der Liste ein Datum anzeigen.
- `enable_detail`: Soll das Element auf eine Detailseite weiterleiten können. Zusätzlich wird angegeben, ob eine Detailseite überhaupt verfügbar ist.
- `list_with_date`: Dieses Auswahlfeld beinhaltet mehrere Varianten.
  - `none`: Soll die Liste ohne Gruppierung nach Zeit angezeigt werden.
  - `by-year`: Sollen die Listenelemente nach Jahren gruppiert angezeigt und navigiert werden.
  - `by-month`: Sollen die Elemente durch einen Text mit dem Monat getrennt werden.
- `items_with_number`: Dieses Auswahlfeld beinhaltet mehrere Varianten.
  - `none`: Sollen die Liste keine Nummerierung besitzen.
  - `numbered`: Sollen die Listenelemente mit einer Nummer angezeigt werden.
  - `inverted`: Sollen die Listenelemente mit einer Nummer in umgekehrter Zählweise angezeigt werden.

### Beispiele:

Anzeige von z.B. Artikeln als Liste mit Bild und ``list_with_date`` mit dem Werte ``by-year``. ``items_with_number`` hat den Wert ``none``.



Anzeige von z.B. Einsätzen als Liste ohne Bild und ``list_with_date`` mit dem Werte ``by-month``. Invertiert nummeriert wird über ``items_with_number`` = ``inverted``



Anzeige von z.B. Terminen als Liste ohne Bild und `list\_with\_date` mit dem Werte `by-year`. `items\_with\_number` hat den Wert `none`.



Anzeige von z.B. Fahrzeugen als Liste ohne Datum und Nummer:

`list\_with\_date` = `inverted`

`items\_with\_number` = `inverted`



### 3.2.4 Seiten

The screenshot shows the Strapi form for creating a new 'Seiten' (Pages) entry. The form includes the following fields and controls:

- identifier\***: A text input field containing 'Einsätze'.
- slug\***: A text input field containing 'einsaetze' with a refresh icon.
- ref\_only\_access**: A toggle switch set to 'FALSE'.
- hero\***: A section for the hero image, labeled 'banner'. It contains a preview of a fire scene image (IMG-20240328-WA0011.jpg) and icons for adding, linking, deleting, and editing the image.
- title**: A text input field containing 'vergangene Einsätze'.

Der **Collection Type Seiten** speichert alle Seiten, die erreicht werden können. Jede Seite ist über den URL-Pfad, welche im **Slug** definiert ist, erreichbar, sofern `ref_only_access` auf `false` steht und eine Navigationseinstellung den **Slug** nicht überlagert. Standardmäßig ist die Erreichbarkeit auf `<meine Domain>/<slug>` gesetzt.

Der **Slug** kann auch vollständige Pfade beinhalten. Dabei wird ein Pfad wie `einsaetze` zu `<meine Domain>/einsaetze` oder `mach-mit-formular` zu `<meine Domain>/mach-mit/formular`. Ein `_` stellt dabei ein `/` dar.

Ist bei einer Seite `ref_only_access` auf `true` gesetzt, so ist eine Seite nur noch per Verlinkung über die Navigation erreichbar.

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.

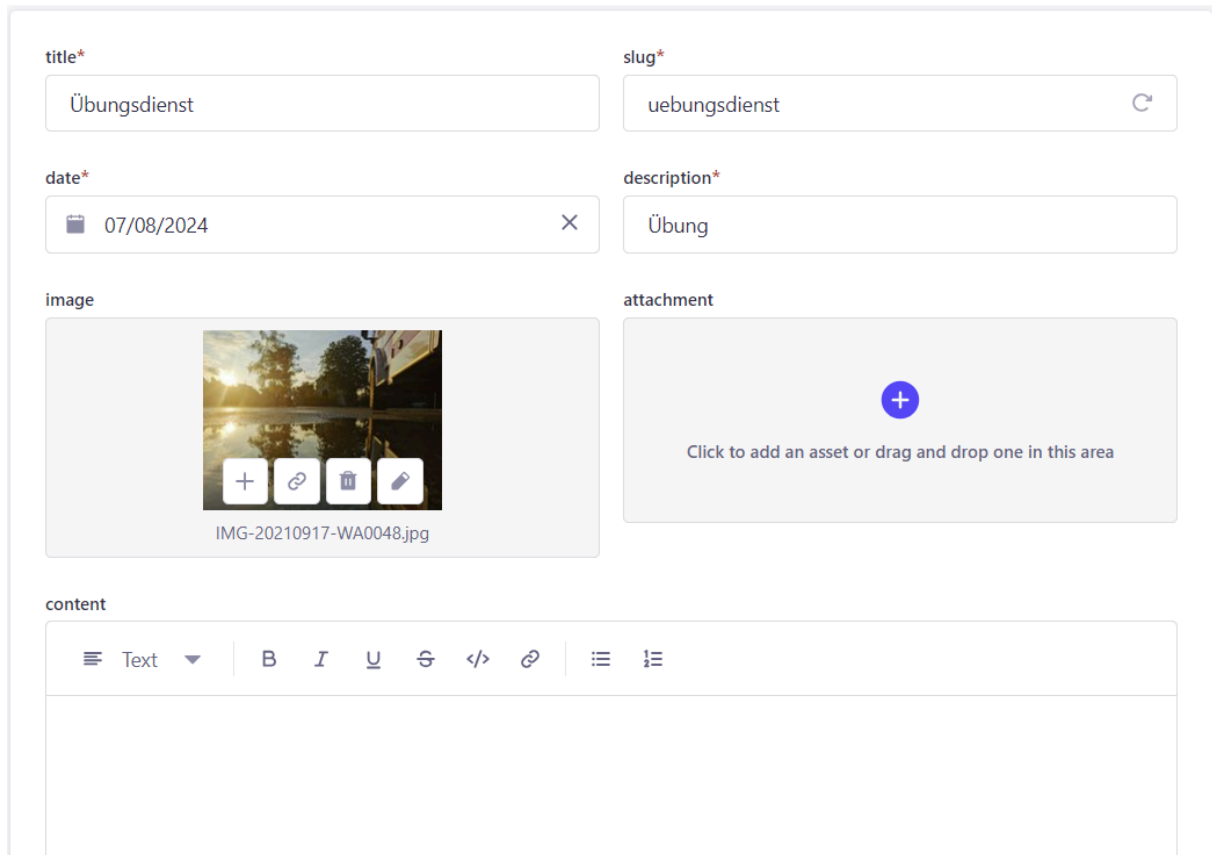
Eine Seite kann ein Banner (Hero) haben. Dieses Banner wird direkt unter der Navigation angezeigt. Der Hero kann mit einem Unterüberschrift versehen werden.

Ist das Banner oder die Unterüberschrift nicht gesetzt, so wird der nicht erstellte Inhalt nicht 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 sunset image with a filename "IMG-20210917-WA0048.jpg" and icons for adding, linking, deleting, and editing.
- attachment**: A large grey box 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 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 die Detailseite eines Artikels geöffnet werden kann, wird in der Sammlungsreferenz (Abschnitt 3.2.3) anhand des Wertes `enable_detail` eingestellt.

### 3.2.6 Einsätze

The screenshot shows the Strapi CMS form for the 'Einsätze' collection type. 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.
- date\***: A date picker showing '04/10/2024' and a time picker showing '02:00'.
- description\***: A text input field containing 'Feuer'.
- image**: 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'.
- attachment (1 / 1)**: A section showing a single image attachment. The image is a photograph of a fire with thick black smoke. 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 die Detailseite eines Einsatzes geöffnet werden kann, wird in der Sammlungsreferenz ([Abschnitt 3.2.3](#)) anhand des Wertes `enable_detail` eingestellt.

### 3.2.7 Fahrzeuge

The screenshot shows the Strapi admin interface for the 'Fahrzeuge' collection type. The form includes the following fields:

- title\***: A text input field containing 'HLF 20/10'.
- slug\***: A text input field containing 'hlf-20-10' with a refresh icon.
- date\***: A date picker field showing '02/14/2025' with a close icon.
- description\***: A text input field containing 'unser HLF'.
- image**: A field showing a preview of a vehicle at night with a file name '20240328\_193109.jpg' and action icons (add, link, delete, edit).
- attachment (1 / 2)**: A field showing a gallery of two vehicle images with a file name '20240328\_193109.jpg' and action icons.
- content**: A rich text editor with a toolbar containing icons for text, bold, italic, underline, link, unlink, code, and list.

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.

Jedes Fahrzeug 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 die Detailseite eines Fahrzeugs geöffnet werden kann, wird in der Sammlungsreferenz ([Abschnitt 3.2.3](#)) anhand des Wertes `enable_detail` eingestellt.

### 3.2.8 Termine:

The screenshot shows the Strapi admin interface for the 'Termine' collection type. The form is organized 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 die Detailseite eines Termins geöffnet werden kann, wird in der Sammlungsreferenz ([Abschnitt 3.2.3](#)) anhand des Wertes `enable_detail` eingestellt.

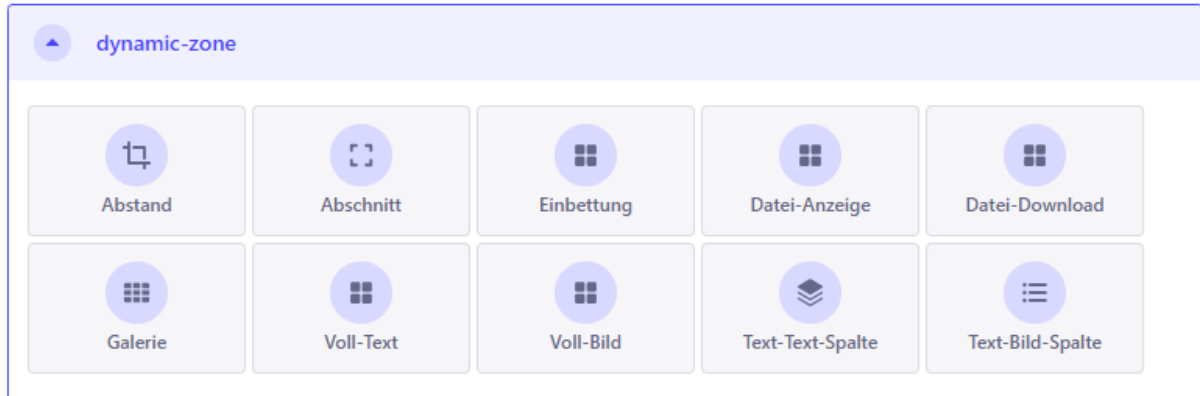
### 3.2.9 User

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.



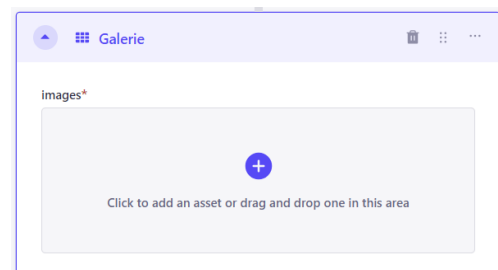
Komponenten aus `shared` werden verwendet, um Inhalte aus anderen Listen verwenden zu können.



### 3.3.1 Dynamic Zone

#### Galerie

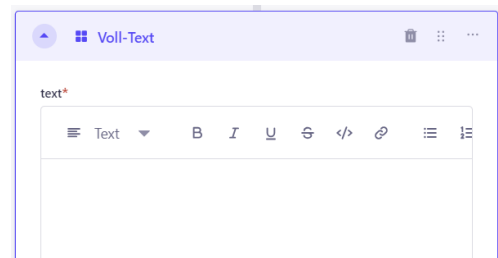
Die Komponente Galerie ermöglicht die Anzeige mehrerer Bilder in einer Gitteransicht.





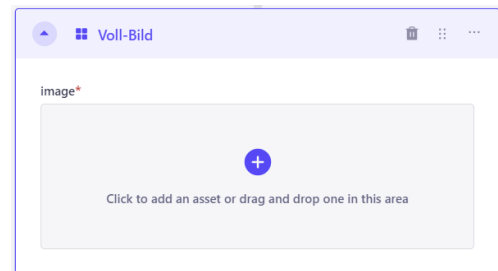
### Voll-Text

Die reine Text-Komponente ermöglicht das schreiben längerer Berichte. So zum Beispiel auch das Impressum oder die Datenschutzerklärung.



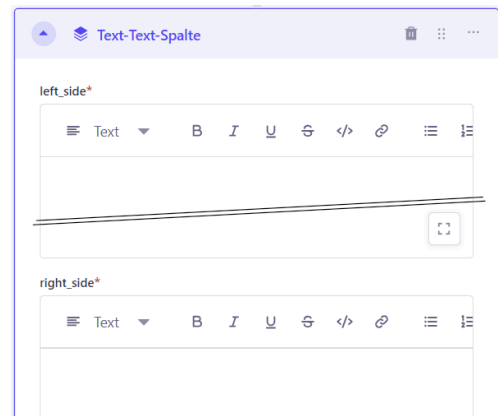
### Voll-Bild

Einzelne Bilder können auch auf Größe einer ganzen Seite angezeigt werden.



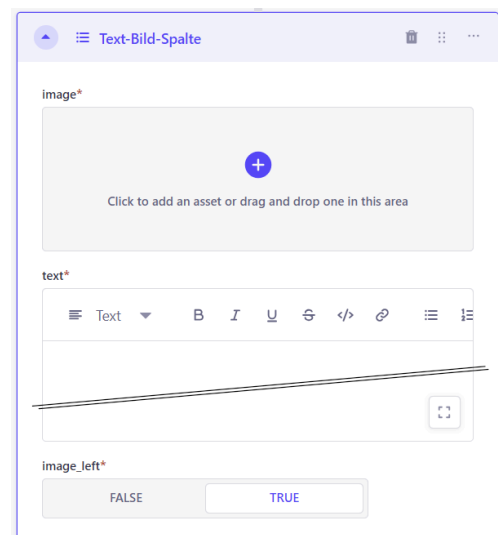
### Text-Text-Spalte

Neben dem 1-spaltigen Layout gibt es auch die Möglichkeit 2 Spalten nebeneinander anzeigen zu können. Auf kleineren Bildschirmen werden die Texte übereinander angezeigt.



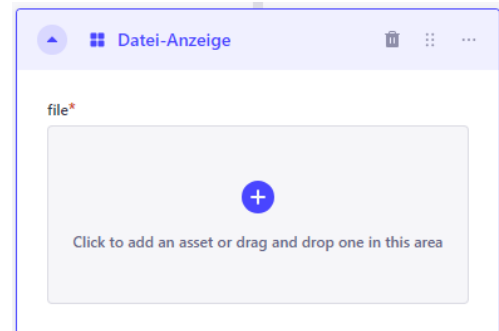
### Text-Bild-Spalte

Erweitert kann auch ein Bild in einer Spalte links oder rechts angezeigt werden. Auch hier werden die Inhalte übereinander angezeigt, wenn die Webseite auf kleineren Bildschirmen angezeigt wird.



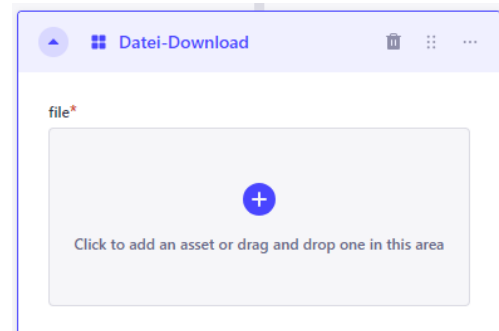
### Datei-Anzeige

Diese Komponente ermöglicht die Einbettung von Bildern oder PDFs. Wird ein PDF oder ein Bild angegeben, so wird dieses angezeigt. Ist eine Datei nicht darstellbar, wird nur der Download-Button angezeigt, sofern dieser nicht ausgeblendet ist.



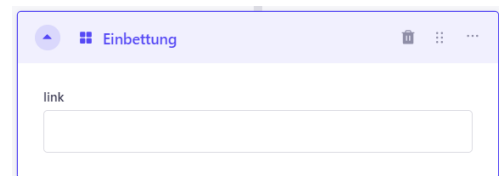
### Datei-Download

Es besteht auch die Möglichkeit, Dateien zum Download zur Verfügung zu stellen. Diese Komponente ermöglicht den Download einer jeden angegebenen Datei.



### Einbettung

Die Einbettung ermöglicht die Anzeige externer Links. Diese Links können Bilder, PDFs oder Webseiten sein. Ein Beispiel für eine Einbettung wäre die öffentliche Kalender-Ansicht des FF Admin.



## 3.3.2 Shared

### Artikel hervorheben

Mit dieser Komponente lassen sich Artikel hervorheben, welche besondere Aufmerksamkeit erhalten sollen. Dazu können dann auch noch ein Titel und eine Beschreibung angegeben werden.

Weiterhin kann zu der URL weitergeleitet werden, unter der die Artikel-Anzeige ist.



### Inhaltsauswahl

Die Inhaltsauswahl kann die Listen der Artikel, Einsätze, Fahrzeuge oder Termine anzeigen, sofern diese in der Sammlungsreferenz (Abschnitt 3.2.3) mit der Anzeige konfiguriert wurden. Hier kann dann auch eingestellt werden, ob die Ansicht auf die Detailansicht der jeweiligen Einträge weiterleiten kann.

