

Details

Name: Name des Templates. Wird benötigt beim definieren vom Menüpunkt. Muss einzigartig sein, kann ansonsten frei gewählt werden.

Tabellenname: Der Name der Tabelle in der Datenbank.

ID Feld Name: Das Datenbankfeld, welches den Primärschlüssel der Tabelle enthält. Heisst in den meisten Fällen 'id' oder ähnlich. Wird für Navigation und verbundene Tabellen benötigt.

SQL Kondition: Optionale SQL Bedingungen, um die angezeigten Einträge einzuschränken. Muss in SQL Syntax geschrieben sein. Z.B. "alter >= 18 AND alter <= 24" um nur Personen mit alter von 18 bis 24 zu laden. Hier sind einige Variablen via Twig Templating verfügbar:

- URL Parameter (weiter unten erklärt). Z.B.: {{ urlParameters.sektionId }}
- ID der eingeloggten Users: {{ userid }}
- Username der eingeloggten Users: {{ username }}

Sortierung: Hier kann optional eine oder mehrere Sortierungen angegeben werden, in SQL Format. Z.B. "alter ASC" um nach Alter (tief zu hoch) zu sortieren.

Limit: Hier kann ein Limit gesetzt werden, wie viele Einträge geladen werden. Kann eine Nummer sein, oder leer gelassen werden (kein Limit).

Suche Aktivieren: Wenn angewählt, zeigt ein Suchfeld oben am Template an, welches alle Felder der Tabelle durchsuchen kann.

Pagination Aktivieren: Wenn angewählt, spaltet die Einträge in mehrere Seiten auf, die durch die Navigation unter dem Template angesteuert werden können. Funktioniert nur, wenn eine Seitenlänge angegeben ist (siehe unten).

Seitenlänge: Definiert die Länge einer Seite, wenn Pagination Aktivieren angewählt ist.

Templates

Header: HTML snippet, welches vor dem Template angezeigt wird.

Template: HTML snippet, welches für jeden geladenen Eintrag in der Tabelle angezeigt wird. Hier sind einige Variablen via Twig Templating verfügbar:

- Daten des aktuellen Eintrages. Z.B.: {{ data.name }}
- URL Parameter (weiter unten erklärt). Z.B.: {{ urlParameters.sektionId }}
- URL zu Detailseite: {{ detailUrl }}. Auch als Anchor {{ detailLink }} oder als Button {{ detailButton }}.
- URL zu Editseite: {{ editUrl }}. Auch als Anchor {{ editLink }} oder als Button {{ editButton }}.
- URL zu Kontaktseite: {{ contactUrl }}. Auch als Anchor {{ contactLink }} oder als Button {{ contactButton }}.

Footer: HTML snippet, welches nach dem Template angezeigt wird.

Detail Template: HTML snippet, welches die Detailseite eines Eintrages darstellt. Hier sind einige Variablen via Twig Templating verfügbar:

- Daten des aktuellen Eintrages. Z.B.: `{{ data.name }}`
- URL Parameter (weiter unten erklärt). Z.B.: `{{ urlParameters.sektionId }}`
- URL zu Detailseite: `{{ detailUrl }}`. Auch als Anchor `{{ detailLink }}` oder als Button `{{ detailButton }}`.
- URL zu Editseite: `{{ editUrl }}`. Auch als Anchor `{{ editLink }}` oder als Button `{{ editButton }}`.
- URL zu Kontaktseite: `{{ contactUrl }}`. Auch als Anchor `{{ contactLink }}` oder als Button `{{ contactButton }}`.

Detailseite Zeigen: Aktiviert die Detailseite wenn angewählt. Ansonsten wird der User auf die Liste weitergeleitet.

Felder

Hier werden alle Tabellenfelder definiert, die später im Template und auf der Detailseite verfügbar sind (wie `{{ data.name }}`).

Name: Der Name des Feldes in der Datenbank. Bestimmt auch wie das Feld im Template angesprochen wird. Z.B.: Name = "geburtsdatum" -> `{{ data.geburtsdatum }}` im Template.

Im Formular Anzeigen: Ob das Feld auf der Editseite und der Neuseite angezeigt werden soll, bzw. ob das Feld durch den User bearbeitbar sein soll.

Label: (Nur wenn im Formular angezeigt) Wie das Feld auf der Editseite und der Neuseite bezeichnet wird.

Typ: (Nur wenn im Formular angezeigt) Der Typ des Feldes. Dies ist wichtig für die interne Validierung und die Art den Inputfeldes.

Erforderlich: Wenn angewählt, kann das Feld nicht leer sein auf der Editseite und der Neuseite.

URL Parameter

Hier können URL Parameter definiert werden, die später im Template und auf der Detailseite verfügbar sind (wie `{{ urlParameters.sektionId }}`), oder eine SQL Kondition beeinflussen können.

Name: Der Name des Parameters. Wenn dieser Name als Parameter in der URL vorhanden ist, wird der Wert im Template oder in den SQL Konditionen verfügbar gemacht. Bestimmt auch wie der Parameter im Template oder in den SQL Konditionen angesprochen wird. Z.B.: Name = "sektionId" -> `{{ urlParameters.sektionId }}` im Template und SQL Kondition.

Default: Wenn der Parameter nicht in der URL vorhanden ist, wird stattdessen der Default Wert verwendet.

In DB Einfügen: Wenn angewählt, wird der Wert des URL Parameters in die Datenbank eingetragen, wenn ein Eintrag bearbeitet oder erstellt wird. Der Parameter muss den gleichen Namen haben, wie das Datenbankfeld.

Verbundene Tabellen

Hier können mehrere Tabellen definiert werden, die auf verschiedene Arten zur Haupttabelle verbunden sind. Diese werden an die relevanten Einträge gebunden und sind im Template verfügbar. Z.B.: `{{ data.sektion }}`. Wird immer als array gespeichert, muss also in einer Schleife ausgegeben werden, wie `{% for sektion in data.sektion %} {{ sektion.name }} {% endfor %}`.

DB Name: Der Name der Tabelle in der Datenbank.

Eindeutige Interne Bezeichnung: Wie die Tabelle im Template bezeichnet wird (wie `{{ data.INTERNE_BEZEICHNUNG }}`), sowie wie die Komponente die Tabelle intern bezeichnet. Muss einzigartig sein, kann ansonsten frei gewählt werden.

Anzeigename: (Nur wenn im Formular angezeigt) Wie die verbundene Tabelle auf der Editseite und der Neuseite bezeichnet wird.

Label Feld: (Nur wenn im Formular angezeigt) Auf der Editseite und der Neuseite, das Feld der verbundenen Tabelle, welches verwendet wird, um die Optionen der verbundenen Tabelle zu betiteln.

Felder: Welche Felder von der verbundenen Tabelle geladen werden soll. Mehrere Felder werden mit Kommas getrennt. Z.B.: "name,email,geburtsdatum".

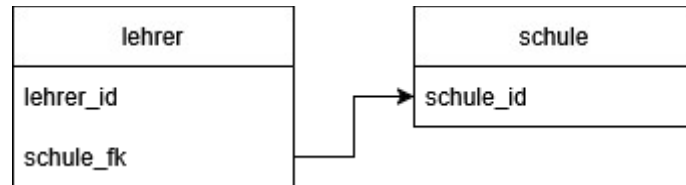
Im Formular Anzeigen: Ob die verbundene Tabelle auf der Editseite und der Neuseite angezeigt werden soll, bzw. ob die verbundene Tabelle durch den User bearbeitbar sein soll.

Verbindungstyp: Wie die verbundene Tabelle und die Haupttabelle verknüpft ist. Typen sind unten weiter erklärt.

n:1

Beispielmodell

Ein Lehrer gehört zu genau einer Schule. Eine Schule hat mehrere Lehrer.



Verbindungsinfos (Lehrer als Haupttabelle)

Haupttabelle

Foreign key zu verbundene
Tabelle:

schule_fk

Verbundene Tabelle

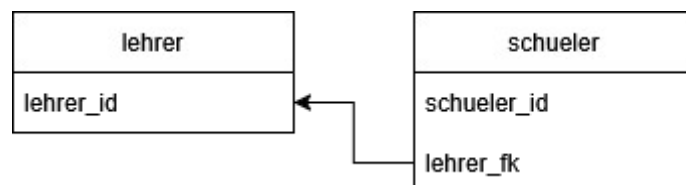
ID Feld:

schule_id

1:n

Beispielmodell

Ein Lehrer hat mehrere Schüler. Ein Schüler hat nur einen Lehrer.



Verbindungsinfos (Lehrer als Haupttabelle)

Verbundene Tabelle

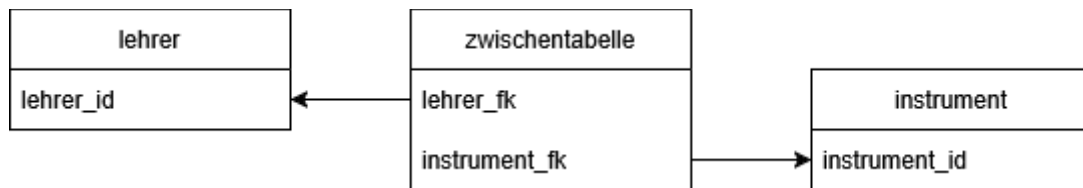
Foreign key zu
Haupttabelle:

lehrer_fk

n:n

Beispielmodell

Ein Lehrer lehrt mehrere Instrumente. Ein Instrument kann von mehreren Lehrern gelehrt werden.



Verbindungsinfos (Lehrer als Haupttabelle)

Zwischentabelle

DB Name:	<input type="text" value="zwischen_tabelle"/>
Foreign key zu Haupttabelle:	<input type="text" value="lehrer_fk"/>
Foreign key zu verbundene Tabelle:	<input type="text" value="instrument_fk"/>

Verbundene Tabelle

ID Feld:	<input type="text" value="instrument_id"/>
----------	--

Kontaktformular

Hier können Daten angegeben werden, mit denen ein Kontaktformular automatisch generiert werden. Dieses sendet ein E-Mail an die Kontaktperson, aber ohne dass die E-Mail Adresse im Frontend angezeigt wird.

E-Mail DB Feld: Das Datenbankfeld der Tabelle, welches die E-Mail Adresse der Kontaktperson enthält.

Kontaktperson Anzeigename: Wie der Name der Kontaktperson im Kontaktformular angezeigt wird, als Twig Template. Z.B.: "{{ data.vorname }}" "{{ data.nachname }}"

E-Mail Benachrichtigungen

Hier können automatische E-Mails eingerichtet werden, die die Empfänger über alle neuen und/oder editierten Einträge benachrichtigt.

Email Nach: Nach welchen Ereignissen E-Mails verschickt werden sollen.

Empfänger: Die E-Mail Adressen, welche die Benachrichtigungen erhalten sollten. Es können mehrere angegeben werden, mit Kommas getrennt.

User ID Verknüpfung

Mit dieser Option, können User berechtigt werden, spezifische Einträge zu bearbeiten. Die Tabelle muss hierfür ein Feld haben, welches die ID des berechtigten Users beinhaltet.

ID Verknüpfung Feld: Das Datenbankfeld der Tabelle, welches die ID des berechtigten Users beinhaltet. Wenn ausgefüllt wird das Template edit-only, d.h. es ist nur die Editseite sichtbar, mit dem Eintrag, welcher die ID des eingeloggten User enthält.

Rechte

Bearbeiten Erlaubt: Ob die Einträge der Tabelle bearbeitbar sein sollen.

Erstellen Erlaubt: Ob neue Einträge erstellbar sein sollen.

Formularverhalten: Zu welcher Seite navigiert werden soll, nach dem Erstellen oder Bearbeiten eines Eintrages.

Zugang: Welches Usergruppen Zugang zum Template haben soll.