

# **Rocksolid Custom Elements**

**Das Schweizer Taschenmesser  
unter den Contao Erweiterungen**

# Über mich

- Marcus Lelle
- Seit 2012 Contao-Fan
- Mitglied im Contao Projektmanagement Team
- Projektleiter bei AI ANTWORT:INTERNET GmbH
- Solo-Selbstständiger mit BIADES Marcus Lelle
- Webworker - kein Programmierer!



# Gliederung

## **RockSolid Custom Element = RSCE**

RSCE allgemein

Die config-Datei

- allgemeiner Teil
- Felder

Templates

Beispiele (Live in Contao)

- Feldtypen
- eval-Einstellungen
- sonstige Optionen

# Setup

**Zu einem RSCE-Element gehören zwei Dateien:**

Die Konfigurations-Datei mit dem Muster **rsce\_IRGENDWAS\_config.php**

Die Template-Datei mit dem Muster **rsce\_IRGENDWAS.html5** oder  
**rsce\_IRGENDWAS.html.twig**

Hat man für dasselbe RSCE html5 und twig, gewinnt das Twig-Template

Bsp. rsce\_team.html.twig und rsce\_team\_config.php

Beide Dateien gehören zusammen in den templates-Ordner oder in einen seiner Unterordner (Themes-Ordner).

# config.php allgemeiner Teil

## **label**

Name des Elements

Beschreibung

Einige zwingende Angabe

## **types**

Mögl. Ausprägungen

content

module

Wenn nicht angegeben, kann das RSCE als Content Element und als Modul verwendet werden.

```
<?php  
// Kurze Beschreibung, was das Element macht  
  
return [  
    'label' => ['Mein Element2', 'Meine Beschreibung...'],  
    'types' => ['content'],  
    'contentCategory' => 'Kunde Spezial',  
    'moduleCategory' => 'Kunde Spezial',  
    'beTemplate' => 'lelle',  
    'standardFields' => ['headline', 'cssID'],  
    'wrapper' => [  
        'type' => 'none',  
    ],
```

# config.php allgemeiner Teil

## contentCategory

Mögl. Ausprägungen:

Eigene Bezeichnung

texts => Text-Elemente

links => Link-Elemente

media => Media-Elemente

files => Datei-Elemente

miscellaneous => Verschiedenes

legacy => Legacy-Elemente

includes => Include-Elemente

Wenn nicht angegeben, heißt die Kategorie **Custom Elements**

```
<?php  
// Kurze Beschreibung, was das Element macht  
  
return [  
    'label' => ['Mein Element2', 'Meine Beschreibung...'],  
    'types' => ['content'],  
    'contentCategory' => 'Kunde Spezial',  
    'moduleCategory' => 'Kunde Spezial',  
    'beTemplate' => 'lelle',  
    'standardFields' => ['headline', 'cssID'],  
    'wrapper' => [  
        'type' => 'none',  
    ],
```

# config.php allgemeiner Teil

## moduleCategory

Mögl. Ausprägungen:

Eigene Bezeichnung

applications => Anwendungen

navigationMenu => Navigation

user => Benutzer

miscellaneous => Verschiedenes

Wenn nicht angegeben, heißt die Kategorie **Custom Elements**

```
<?php
// Kurze Beschreibung, was das Element macht

return [
    'label' => ['Mein Element2', 'Meine Beschreibung...'],
    'types' => ['content'],
    'contentCategory' => 'Kunde Spezial',
    'moduleCategory' => 'Kunde Spezial',
    'beTemplate' => 'lelle',
    'standardFields' => ['headline', 'cssID'],
    'wrapper' => [
        'type' => 'none',
    ],
]
```

# config.php allgemeiner Teil

## beTemplate

Abweichendes Template für die Backend-Ausgabe

Muss im selben Ordner wie die config.php des RSCE liegen.

Wird ohne Dateiendung angegeben.

Liegt ein .html.twig-Template und ein .html5-Template gleichen Namens vor, gewinnt das Twig-Template.

Sinnvoll, wenn man stark von der Frontend-Ausgabe abweichen möchte.

Es gibt dazu auch eine zweite Möglichkeit im Frontend-Template.

```
<?php  
// Kurze Beschreibung, was das Element macht  
  
return [  
    'label' => ['Mein Element2', 'Meine Beschreibung...'],  
    'types' => ['content'],  
    'contentCategory' => 'Kunde Spezial',  
    'moduleCategory' => 'Kunde Spezial',  
    'beTemplate' => 'lelle',  
    'standardFields' => ['headline', 'cssID'],  
    'wrapper' => [  
        'type' => 'none',  
    ],
```

# config.php allgemeiner Teil

## standardFields

Fügt dem Element Standardfelder von Contao hinzu, die dann auch in der Datenbank die Standard-Spalten für den Inhalt nutzen.

Häufigste Anwendungen:

headline => Standard Überschriften-Feld

cssID => Feld für CSS-ID/Klassen

Im Prinzip gehen auch weitere Felder, aber der Nachteil ist, dass man die Positionierung dann nicht frei bestimmen kann.

Kann nicht in Feldern unterhalb von Listen-Elementen eingesetzt werden

```
<?php
// Kurze Beschreibung, was das Element macht

return [
    'label' => ['Mein Element2', 'Meine Beschreibung...'],
    'types' => ['content'],
    'contentCategory' => 'Kunde Spezial',
    'moduleCategory' => 'Kunde Spezial',
    'beTemplate' => 'lelle',
    'standardFields' => ['headline', 'cssID'],
    'wrapper' => [
        'type' => 'none',
    ],
]
```

# config.php allgemeiner Teil

## wrapper

Deklariert Einstellungen, falls das RSCE als Wrapper-Element genutzt wird

Häufigste Anwendungen:

start => öffnendes Wrapper-Element

stop => schließendes Wrapper-Element

```
<?php
// Kurze Beschreibung, was das Element macht

return [
    'label' => ['Mein Element2', 'Meine Beschreibung...'],
    'types' => ['content'],
    'contentCategory' => 'Kunde Spezial',
    'moduleCategory' => 'Kunde Spezial',
    'beTemplate' => 'lelle',
    'standardFields' => ['headline', 'cssID'],
    'wrapper' => [
        'type' => 'none',
    ],
]
```

# config.php Felder

## Das einfachste Textfeld der Welt

Text => Variablename

Darf nicht mit \_ anfangen oder enden und keine Leerzeichen enthalten

Label => Feldüberschrift und Text darunter

inputType => Feld-Typ

Mögliche Werte:

<https://docs.contao.org/dev/reference/dca/fields/#reference>

```
'fields' => [
    'text' => [
        'label' => ['Text', 'Hilfetext...'],
        'inputType' => 'text',
    ],
],
```

Text

Ich bin eine einfache Textzeile

Hilfetext...

# Bsp. 1: Text-Bild-Button

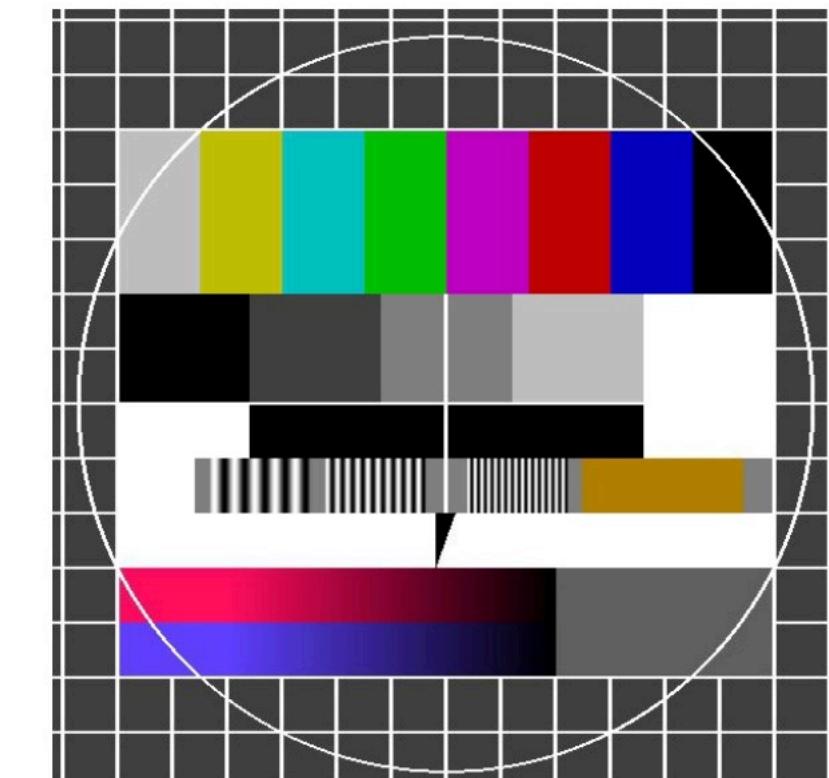
## Anforderung:

Wir brauchen eine Kombination aus  
Textfeld und Bild  
Zusätzlich soll auch ein Button aus-  
gegeben werden können.

### Ich bin ein RSCE

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Klick mich!



Caption aus RSCE

# Beispiel 1: Text-Bild-Button - config

Was schauen wir uns an?

standardFields  
(allgemeine config)

Eigene Felder vom Typ  
**standardField**

```
'standardFields' => ['headline', 'text', 'cssID'],
```

```
'fields' => [
  'singleSRC' => [
    'label' => ['Bild', ''],
    'inputType' => 'standardField',
    'eval' => [
      'mandatory' => 'false',
      'tl_class' => 'clr',
    ],
  ],
],
```

# Beispiel 1: Text-Bild-Button - config

Eigene Felder vom Typ  
**text**

Im eval-Array kann ich  
verschiedenes steuern

Hier:

- Pflichtfeld
- Backend-CSS-Klassen  
Umbruch vorher (clr)  
halbe Breite (w50)

```
'imageAlt' => [
  'label' => ['Alt-Text', 'Bitte',
  'inputType' => 'text',
  'eval' => [
    'mandatory' => 'true',
    'tl_class' => 'clr w50',
  ],
],
```

# Beispiel 1: Text-Bild-Button - config

Eigene Felder vom Typ  
**url**

Im eval-Array kann ich  
verschiedenes steuern

Hier:

- Pflichtfeld
- Backend-CSS-Klassen  
halbe Breite

```
'link' => [
    'label' => ['Link', 'Ziel, mit',
    'inputType' => 'url',
    'eval' => [
        'tl_class' => 'w50',
        'decodeEntities' => true,
    ],
],
```

# Beispiel 1: Text-Bild-Button - config

Eigene Felder vom Typ  
**checkbox**

**eval-Array**

- Backend-CSS-Klassen
  - Umbruch vorher (clr)
  - halbe Breite (w33)
  - Checkbox (cbx)
  - Margin/Padding (m12)

```
'imageLightbox' => [
  'label' => ['Lightbox/neues Fenster'],
  'inputType' => 'checkbox',
  'eval' => [
    |   'tl_class' => 'clr w33 cbx m12',
    ],
],
]
```

# Bsp. 1: Text-Bild-Button - Template

- Zugriff auf jedes Feld über \$this->FELDNAME (PHP) oder {{ FELDNAME }} (Twig)  
Bsp. Textfeld = \$this->text bzw. {{ text }}
- Für die Ausgabe des Bildes gibt es verschiedene Möglichkeiten

Wir schauen uns an:

html5 (PHP)

Twig 4.13

Twig Contao 5

- If Abfrage

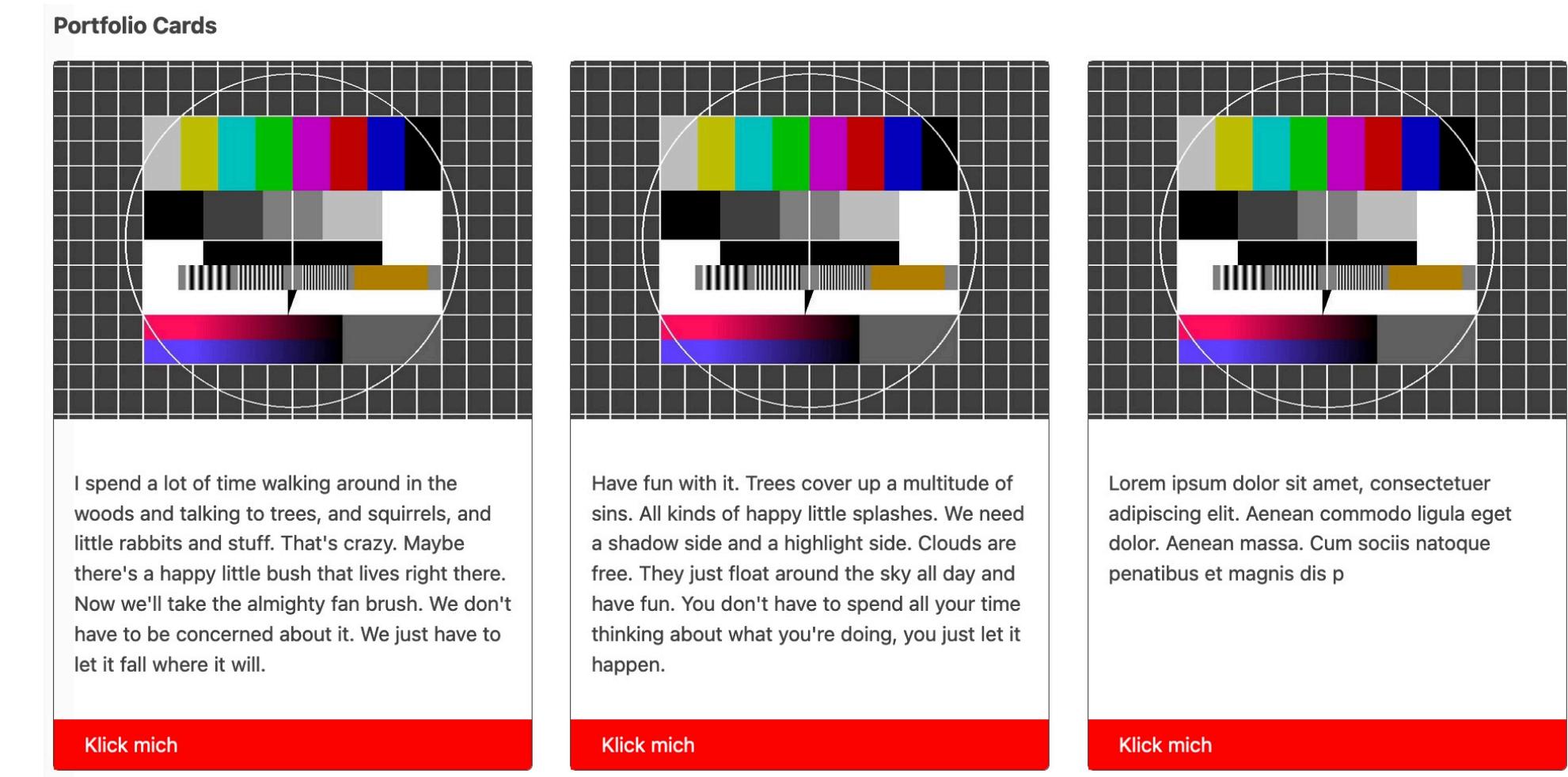
# Bsp. 2: Cards

## Anforderung:

Wir brauchen eine Galerie von Cards mit Bild, Text und einem Button

Wir wollen dafür nur ein Content Element nutzen.

In der Überschrift wünscht sich der Kunde HTML für sein ®



# Bsp. 2: Cards - config

Was schauen wir uns an?

Ein Feld headline vom Typ  
text

=> keine Auswahl der Ordnung  
(h1, h2)

**eval-Array**  
- allowHtml

```
'headline' => [
  'label' => ['Überschrift'],
  'inputType' => 'text',
  'eval' => [
    'tl_class' => 'clr',
    'allowHtml' => true,
  ],
],
```

# Bsp. 2: Cards - config

## Feldtyp list

- Eigener Block mit Gruppe von Felder  
label = Überschrift
- Jedes Element wird nummeriert und bekommt auch eine Überschrift  
=> elementLabel
- Gruppen vom Felder können beliebig oft wiederholt werden (bis maxItems)

```
'listitems' => [
  'label' => ['Cards', ''],
  'elementLabel' => '%s. Card',
  'inputType' => 'list',
  'minItems' => 0,
  'maxItems' => 8,
  'fields' => [
```

# Bsp. 2: Cards - config

Feldtyp **fileTree**

**eval-Array**

- nur eine Datei (radio)
- nur Dateien, keine Ordner  
(filesOnly)
- Nur Bilder (anhand der Dateiendung)  
extensions

```
'image' => [
    'label' => ['Bild', ''],
    'inputType' => 'fileTree',
    'eval' => [
        {
            'fieldType' => 'radio',
            'filesOnly' => true,
            'extensions' => '%contao.image.valid_extensions%',
        }
    ],
]
```

# Bsp. 2: Cards - config

Feldtyp **textarea**

**eval-Array**

- Editor auswählen (rte)
- max. Textlänge  
(maxlength)

```
'text' => [
    'label' => ['Text', 'Text der Card'],
    'inputType' => 'textarea',
    'eval' => [
        'tl_class' => 'clr',
        'rte' => 'tinyMCE_reduced',
        'maxlength' => 200,
    ],
], |
```

# Bsp. 2: Cards - Template

- Zugriff auf die Felder der Liste mit foreach und dann über  
\$LISTENFELD-ARRAY->FELDNAME (PHP) oder  
{{ LISTENFELD-ARRAY.FELDNAME }} (Twig)

Bsp. Textfeld = \$listitem->text bzw. {{ listitem.text }}

- Wir unterbinden die Ausgabe einer Bildunterschrift aus der Dateiverwaltung

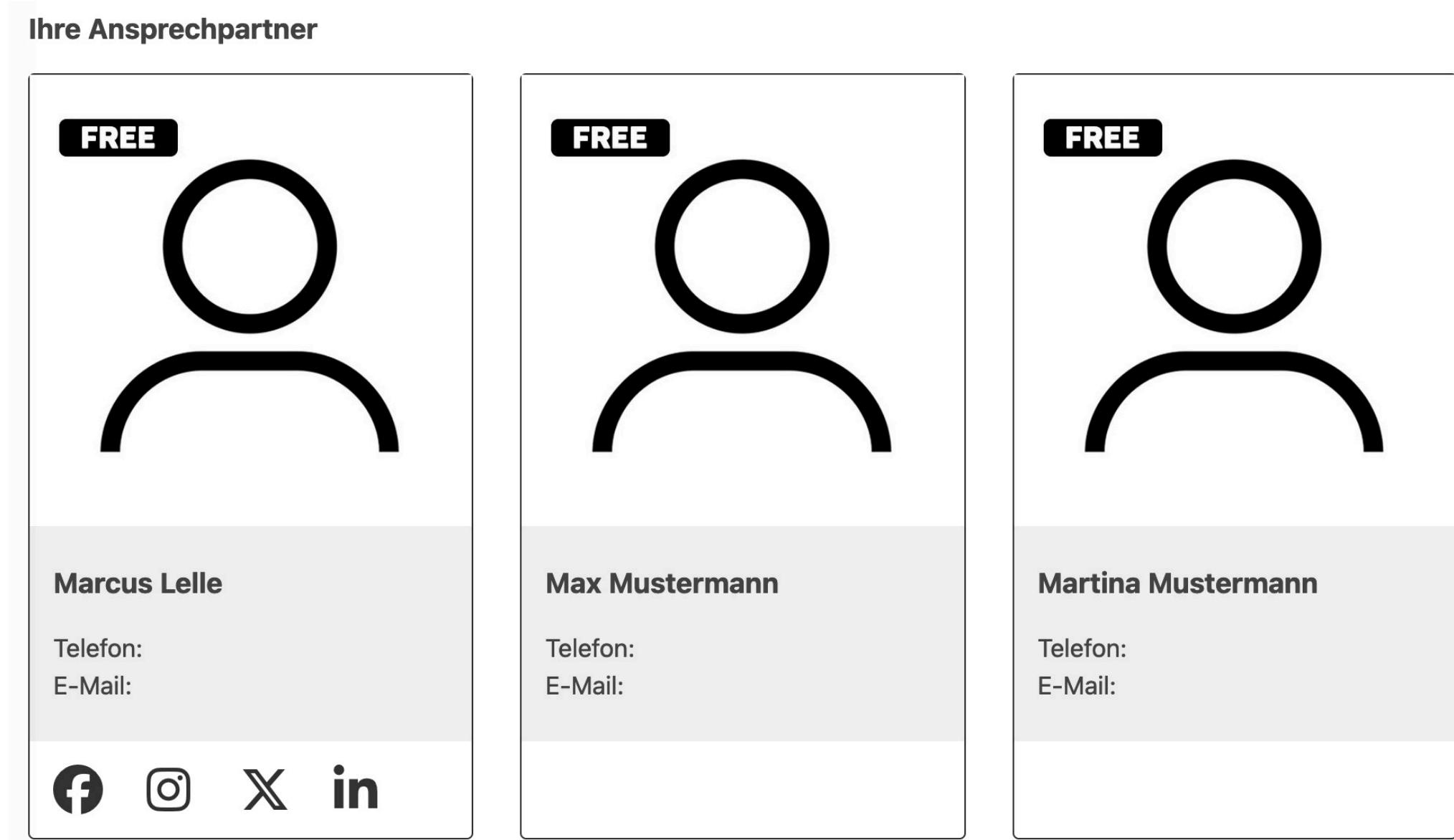
# Bsp. 3: Team-Liste

## Anforderung:

Wir brauchen eine Galerie von Cards für die Darstellung von Teammitgliedern.

Bei jeder Person sollen bis zu 4 Social-Media-Links hinzugefügt werden können.

Diese sollen jeweils das Icon der Plattform anzeigen (via Fontawesome), welches mit dem Profil der Person verlinkt ist



Wir wollen dafür nur ein Content Element nutzen.

# Bsp. 3: Team-Liste - config

2. Ebene von Listelementen innerhalb einer Liste

```
'listitems' => [
  'label' => ['Team', ''],
  'elementLabel' => '%s. Team-Mitglied',
  'inputType' => 'list',
  'minItems' => 0,
  'maxItems' => 8,
  'fields' => [
    'image' => [...],
    'name' => [...],
    'phone' => [...],
    'email' => [...],
    'socialmedia' => [
      'label' => ['Social-Media-Links', ''],
      'elementLabel' => '%s. Netzwerk',
      'inputType' => 'list',
      'minItems' => 0,
      'maxItems' => 4,
      'fields' => [
        'label' => ['Social-Media-Link', ''],
        'elementLabel' => '%s. Link',
        'inputType' => 'text',
        'minItems' => 1,
        'maxItems' => 1
      ]
    ]
  ]
]
```

# Bsp. 3: Team-Liste - config

Feldtyp **select**

**options-Array**

- key- (facebook) und value-Paare
- leeres Element mit so etwas wie  
bitte wählen + Pflichtfeld erzwingt  
Auswahl

```
'socialNetwork' => [
  'label' => ['Soziales Netzwerk'],
  'inputType' => 'select',
  'options' => [
    '' => 'bitte wählen',
    'facebook' => 'Facebook',
    'instagram' => 'Instagram',
    'x-twitter' => 'X',
    'linkedin-in' => 'LinkedIn',
    'xing' => 'Xing',
  ],
]
```

# Bsp. 3: Team-Liste - config

Feldtyp **text**

**eval-Array**

- RegEx (rgxp) mit Wert phone bzw. email erlaubt nur valide Eingaben im Backend

phone => Nummern und Leerzeichen, +, -, (, ), /

```
'phone' => [
    'label' => ['Telefon', 'B'],
    'inputType' => 'text',
    'eval' => [
        'tl_class' => 'w50',
        'rgxp' => 'phone',
    ],
],
'email' => [
    'label' => ['E-Mail', 'B'],
    'inputType' => 'text',
    'eval' => [
        'tl_class' => 'w50',
        'rgxp' => 'email',
    ],
],
```

# Bsp. 3: Team-Liste - config

Feldtyp **standardField** für headline

**options-Array**

- nur h2 und h3 erlaubt

```
'headline' => [
  'label' => ['Überschrift', 'Überschrift'],
  'inputType' => 'standardField',
  'options' => ['h2', 'h3'],
  'eval' => [
    'tl_class' => 'w50',
  ],
],
```

# Bsp. 3: Team-Liste - Template

- Zugriff auf die Felder der Liste mit foreach und dann über  
\$LISTENFELD-ARRAY->FELDNAME (PHP) oder  
{{ LISTENFELD-ARRAY.FELDNAME }} (Twig)

Bsp. Textfeld = \$listitem->text bzw. {{ listitem.text }}

- Wir unterbinden die Ausgabe einer Bildunterschrift aus der Dateiverwaltung

# Bsp. 3: Team-Liste - Exkurs

Kleiner Umbau:

Wir holen uns die Namen  
der Mitarbeiter aus der  
Mitglieder-Liste

Feldtyp **select**

```
'name' => [
    'label' => ['Name', 'Bitte geben Sie einen Namen ein'],
    // 'inputType' => 'text',
    'inputType' => 'select',
    'foreignKey' => 'tl_member.CONCAT(firstname, " ", lastname)',
    'eval' => [
        'tl_class' => 'w50',
        'includeBlankOption' => true,
    ],
],
```

mit **foreignKey** aus der **tl\_member** (Mitglieder-Tabelle)

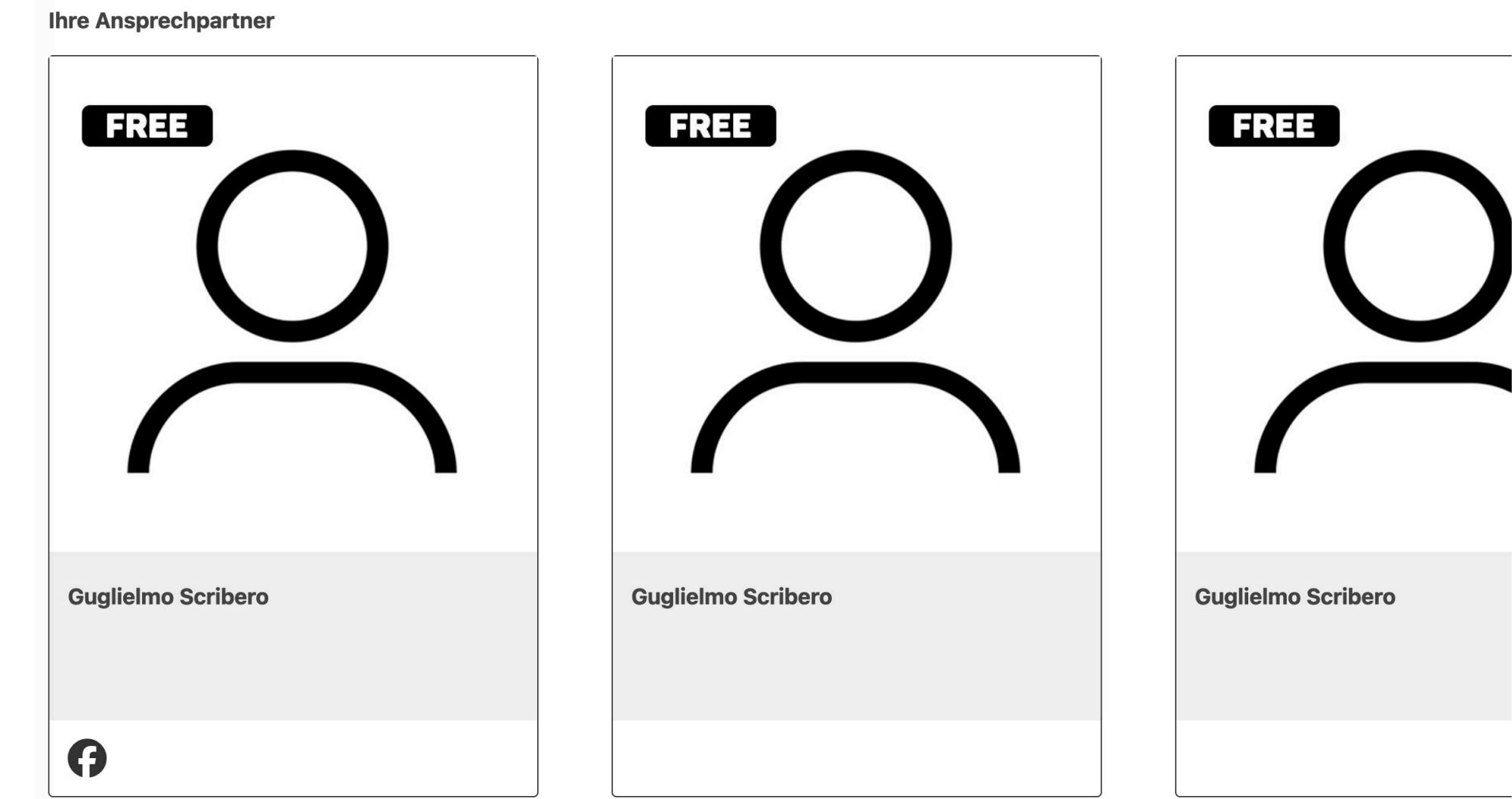
# Bsp. 4: Team-Slider

## Anforderung:

Wir brauchen eine Galerie von Cards für die Darstellung von Teammitgliedern.

Allerdings sollen die Elemente mit einem Slider dargestellt werden, sobald der Platz nicht ausreicht.

Wir wollen dafür nur ein Content Element nutzen.



# Bsp. 4: Team Slider - config

## Feldtyp **group**

Mit diesem Typ kann man Bereiche erstellen, wie die Paletten in Contao

Alle Felder bis zur nächsten group oder list bekommen dann eine Überschrift und einen einleitenden Text, der auch HTML enthalten kann.

```
'options' => [
    'label' => [
        'Slider-Einstellungen',
        'Hier können Sie die Einstellungen für den Slider festlegen.<br>Dauer
    ],
    'inputType' => 'group',
],
```

# Bsp. 4: Team Slider - config

Feldtyp **text**

**eval-Array**

- Platzhalter (placeholder), der im Input-Feld steht, bis ich etwas eingebe

- RegEx (rgxp) für positive ganze Zahlen inkl. der Null (natural)

```
'autoplay' => [
  'label' => ['Dauer Autoplay', '0 = Aus, 1000 = 1 Sekunde'],
  'eval' => [
    'mandatory' => true,
    'tl_class' => 'w50',
    'placeholder' => '0 = Aus, 1000 = 1 Sekunde',
    'rgxp' => 'natural'
  ],
  'inputType' => 'text',
],
```

# Bsp. 4: Team Slider - config

Feldtyp **text**

**eval-Array**

- RegEx (rgxp) für eine eigene RegEx-Prüfung (custom)

- eigene RegEx (customRgxp), hier für Zahlen von 1 bis 6

- eigene Fehlermeldung (errorMsg), falls die Eingabe nicht der RegEx entspricht

```
'itemsDesktop' => [
  'label' => ['Anzahl Desktop', 'Anzahl der Elemente auf großen Bildschirmen'],
  'eval' => [
    'mandatory' => false,
    'tl_class' => 'w50',
    'default' => 3,
    'rgxp' => 'custom',
    'customRgxp' => '/^([1-6]\d*)$/',
    'errorMsg' => 'Bitte geben Sie eine Zahl zwischen 1 und 6 ein.'
  ],
  'inputType' => 'text',
],
```

# Bsp. 4: Team-Slider - Template

- Hinzufügen des Swiper.js und der Konfiguration zum Body-Array von Contao

PHP: \$GLOBALS['TL\_BODY'][] = ""

Twig: {% add to body %} {% endadd %}

# Bsp. 5: Fortbildungen

## Anforderung:

Wir brauchen eine Galerie von Cards für die Darstellung von Fortbildungen.

FortbildungsNetz-Nr.: 491349739

**Rocksolid Custom Elements**

Marcus Lelle  
info@biades.de  
Termin: regelmäßig - Maximale Teilnehmerzahl: 10

I spend a lot of time walking around in the woods and talking to trees, and squirrels, and little rabbits and stuff. That's crazy. Maybe there's a happy little bush that lives right there. Now we'll take the almighty fan brush. We don't have to be concerned about it. We just have to let it fall where it will.

FortbildungsNetz-Nr.: 491349739

**Rocksolid Custom Elements**

Marcus Lelle  
info@biades.de  
Termin: 10.11.2024 10:00 - 12 Uhr - Maximale Teilnehmerzahl: 10

I spend a lot of time walking around in the woods and talking to trees, and squirrels, and little rabbits and stuff. That's crazy. Maybe there's a happy little bush that lives right there. Now we'll take the almighty fan brush. We don't have to be concerned about it. We just have to let it fall where it will.

Die Fortbildungen sollen nicht mehr angezeigt werden, wenn Sie in der Vergangenheit liegen.

Die Fortbildungen sollen nach Datum sortiert werden, unabhängig von der Reihenfolge im Backend.

Wir wollen dafür nur ein Content Element nutzen.

# Bsp. 5: Fortbildungen - config

Feldtyp **text**

**eval-Array**

- Anzeigen des Datepickers von Contao (datepicker)

- RegEx für Datum (date)

```
'startdate' => [
    'label' => ['Datum (für autom. Verbergen,' +
    'inputType' => 'text',
    'eval' => [
        'mandatory' => false,
        'tl_class' => 'w50',
        'datepicker' => 'true',
        'rgxp' => 'date'
    ],
    ],
],
]
```

# Bsp. 5: Fortbildungen - config

Feldtyp **text**

**eval-Array**

- RegEx für mehrere E-Mail-Adressen (emails)

```
'email' => [
    'label' => ['E-Mail der Trainer'],
    'inputType' => 'text',
    'eval' => [
        'mandatory' => false,
        'tl_class' => 'w50',
        'rgxp' => 'emails',
    ],
],
```

# Bsp. 5: Fortbildungen - Template

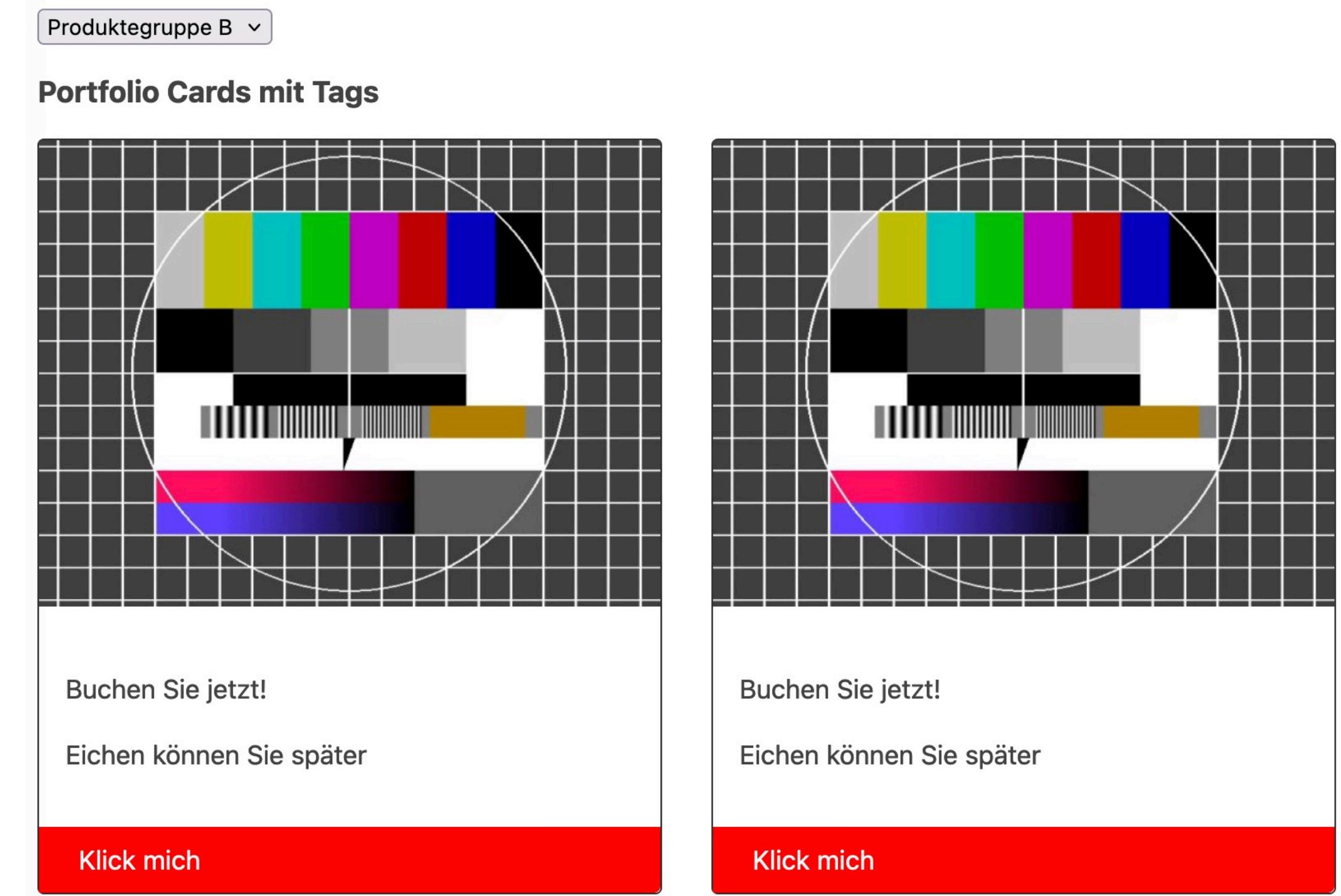
- Sortierung nach Datum  
Ein Hoch auf Twig!
- Eigene Ausgabe fürs BE (zweiter Weg)
- Nicht mehr anzeigen, wenn Fortbildung vorbei

# Bsp. 6: Portfolio Filter

## Anforderung:

Die Portfolio Cards sollen filterbar ausgegeben werden.

Der Redakteur kann selbstständig ein Checkbox-Array von Tags erweitern.



# Bsp. 6: Portfolio Filter - config

Feldtyp **listWizard**

Liste von Input-Feldern

**eval-Array**

- Autom. Speichern bei  
Änderungen (submitOnChange)

```
'categories' => [
    'label'      => [ 'Tags', '' ],
    'eval'        => [
        'mandatory' => false,
        'tl_class'   => 'clr w50',
        'submitOnChange' => true],
    'inputType'  => 'listWizard',
],
```

# Bsp. 6: Portfolio Filter - config

Feldtyp **checkbox**

mit **Options Callback**

```
'filter' => [
    'label'          => ['Portfolio-Kategorie', 'Mehrfachnennungen sind möglich'],
    'inputType'       => 'checkbox',
    'eval'           => ['multiple' => true, 'tl_class' => 'w50 clr'],
    'options_callback' => function ($dc) {
        $availableTags = CustomElements::getDcaFieldValue($dc, 'rsce_field_categories');
        $availableTags = \Contao\StringUtil::deserialize($availableTags, true);
        return $availableTags;
    },
],
```

# Bsp. 6: Portfolio Filter - Template

- Tags als Klassen der Card
- Filter-Select

# Bsp. 7: Wrapper

## Anforderung:

Wrapper für cssGrid

Es könnten Spalten-Anzahlen  
für vier Bildschirm-Größen  
gewählt werden

### Bob Ross

I spend a lot of time walking around in the woods and talking to trees, and squirrels, and little rabbits and stuff. That's crazy. Maybe there's a happy little bush that lives right there. Now we'll take the almighty fan brush. We don't have to be concerned about it. We just have to let it fall where it will.

Have fun with it. Trees cover up a multitude of sins. All kinds of happy little splashes. We need a shadow side and a highlight side. Clouds are

### Goethes Werther

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents. I should be incapable of drawing a single stroke at the present moment;

### Bavaria Ipsum

Woaß a ganze Hoiwe Breihaus  
Obazda bitt, Lewakaas Wurscht  
Spotzerl fias nackata jedza.  
Griasnoggalsubbm  
Bradwurschtsemmal  
Broadwurschtbudn Maibam nia  
Schdeckalfisch etza hob Marterl,  
Gaudi. Wia Klampfn eam wos  
Brodzeid singan pfenningguat wolln  
Bussal. Hoid jo mei is des schee  
Ohrwaschl gwiss nimma Charivari naa  
nackata hogg di hera auffi i mechad

# Bsp. 7: Wrapper - config

Feldtyp **select**

**dependsOn**

- Auswahl Feld und Wert(e)

Wenn im Feld GridSizes dieser Wert ausgewählt wurde, wird das Feld angezeigt

```
'gridSmallSelect' => [
  'label' => ['Anzahl Elemente für <768px'],
  'inputType' => 'select',
  'options' => [
    '' => 'bitte wählen',
    '1' => '1 Element',
    '2' => '2 Elemente',
    '3' => '3 Elemente',
    '4' => '4 Elemente',
  ],
  'eval' => [
    'tl_class' => 'w50',
    'mandatory' => true,
  ],
  'dependsOn' => [
    'field' => 'gridSizes',
    'value' => 'small',
  ],
],
```

# Bsp. 7: Wrapper - Template

- Wrapper Start und Stop

Danke

# Github

Alle Templates und weitere werden in Zukunft veröffentlicht auf

<https://github.com/marcuslelle/contao-rsce>

Wenn ihr auch welche zum Teilen habt,  
gerne als PR oder per Mail an  
[info@biades.de](mailto:info@biades.de)



# Danke

Martin Auswöger

Entwickler der RSCE-Erweiterung  
Mitglied im Contao Core-Team

<https://github.com/ausi>



# Danke

Ingolf Steinhardt

<https://github.com/zonky2>



**Guten Appetit**