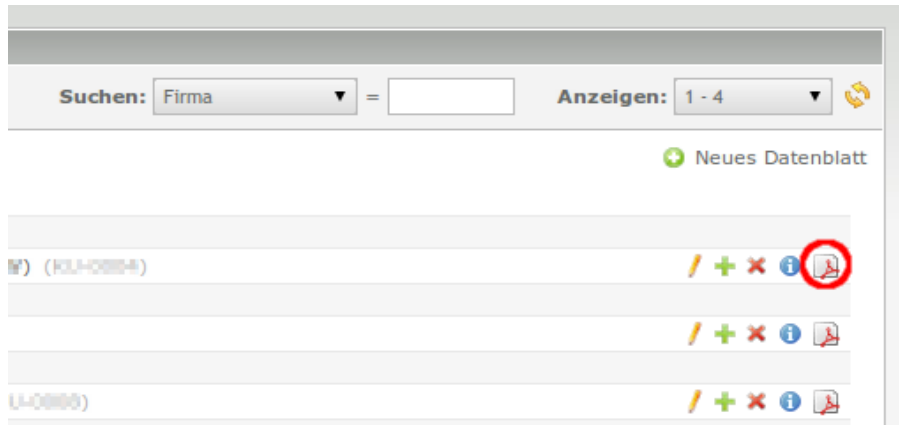


Eine eigene Aktion in eine Contao-Übersicht einfügen

Heute möchte ich zeigen, wie man eine eigene Aktion in eine Contao-Übersicht einfügt. Es wird ein neues Icon hinter die bestehenden eingefügt und mit einer eigenen Aktion verbunden. Ich erläutere dies am Beispiel einer PDF-Übersicht, die ich mir geschrieben habe.

Es soll am Ende ungefähr so aussehen:



Erster Schritt: dca

Zuerst wird im Ordner `dca`, in der Datei der Tabelle (`"TABELLENENNAME".php`) in der Rubrik `list` unter `operations` ein neuer Eintrag erzeugt: (Für eine bessere Übersicht habe ich die Standardaktionen auf dem Bild gekürzt.)

```
1  $GLOBALS['TL_DCA'][$TABELLENENNAME] = array
2  (
3
4
5  // Config
6  'config' => array
7  ( ... ),
8
9
10 // List
11 'list' => array
12 ( ... ),
13   'operations' => array
14   (
15       'edit' => array
16       ( ... ),
17       'copy' => array
18       ( ... ),
19       'delete' => array
20       ( ... ),
21       'show' => array
22       ( ... ),
23       'create' => array
24       (
25           'label' => &$GLOBALS['TL_LANG'][$TABELLENENNAME]['create'],
26           'href' => 'key=create',
27           'icon' => 'system/modules/ORDNER_DER_ERWEITERUNG/html/PDF_file_acrobat.gif'
28       )
29   )
30 ),
```

Unter 'href' wird der Parameter eingegeben, der die Aktion auslöst. In unserem Fall ist es 'key=create'. Wird also die Seite mit dem Parameter 'key=create' aufgerufen, wird unsere Aktion ausgeführt.

Schritt 2: config.php

Nun muss die Aktion bekannt gegeben werden. Dies geschieht in der Datei `config.php` im Verzeichnis `config`. Neben dem normalen Eintrag für das Menü ('`tables`' und '`icon`') wird hier der Eintrag mit dem Namen unserer Aktion erstellt (das ist der Parameter aus der Datei `dca.php`). In unserem Fall `create`. Diesem wird ein Array mit dem Namen der Klasse ('`datasheetPDF`') und der Methode ('`generate`'), die beim Aufruf ausgeführt werden sollen übergeben.

```
1 array_insert(
2     $GLOBALS['BE_MOD'], 0, array(
3         'ÜBERSCHRIFT' => array(
4             'MENÜPUNKT' => array(
5                 'icon' => 'system/modules/ORDNER_DER_ERWEITERUNG/html/PDF_document_red.png',
6                 'tables' => array('TABELLENNAME'),
7                 'create' => array('datasheetPDF', 'generate')
8             )
9         )
10    );
```

Schritt 3: PHP-Klasse

Nun muss noch direkt im Ordner der Erweiterung eine Datei angelegt werden. Diese muss den Namen der Klasse tragen, den wir in Schritt 2 bekannt gegeben haben. Hier rein kommt die PHP-Klasse mit der Methode, die unsere Aktion ausführt, es können aber auch beliebige weitere Methoden enthalten sein. Man kann sich also die PHP-OOP zu Nutze machen. Es ist eine gute Idee, die Klasse je nach Aufgabe von einer Contao-Klasse abzuleiten (z.B.: von `system`). So kann man das Contao-Framework für seine Erweiterung nutzen, was die Arbeit erheblich erleichtert.

```
1 class datasheetPDF extends System {
2
3     public function generate(){
4         // Mach mir ein PDF!
5     }
6 }
```

Beispielhafter Inhalt der Datei `/ORDNER_DER_ERWEITERUNG/datasheetPDF.php`.

Das Einbinden unserer Klassen-Datei erfolgt automatisch anhand des Dateinamens, so dass wir uns um nichts weiter kümmern müssen. Wenn wir nun in der Übersicht auf das neue Icon klicken, wird unsere Methode aufgerufen und die Aktion ausgeführt. Wenn man weiß wie es geht ist es gar nicht schwer.

Jetzt viel Spaß beim Ausprobieren.