

Aktionen mit verschiedenen Icons pro Datensatz

Wie man in die Listen von Contao eigene Aktionen einträgt, habe ich ja bereits im Artikel "Eine eigene Aktion in eine Contao-Übersicht einfügen" beschrieben. Bei diesem Vorgehen wird für alle Einträge der Liste das gleiche Icon angezeigt. Es handelt sich also um ein globales Icon. Es ist nicht möglich verschiedene Icons pro Listeneintrag anzuzeigen (um z.B. einen Status abzubilden). Wie man individuelle Icons für eine Aktion anzeigt, ist Thema dieses Artikels.

Schritt 1: Aktion mit Standard-Icon

Zu nächst legen wir eine Aktion mit Standard-Icon an. Dies geschieht wie in dem Beitrag „Eine eigene Aktion in eine Contao-Übersicht einfügen“.

Schritt 2: Den Button-Callback eintragen

Nun geben wir den `button_callback` bekannt. Dieser erstellt den Link für den Aufruf unserer Aktion mit dem individuellen Icon. Wir tragen dazu in die Datei mit dem Namen unserer Tabelle im `dca`-Verzeichnis unserer Erweiterung eine Klasse und einen Methodennamen ein (z.B.

`demoErweiterung/dca/tl_beispiel.php`).

```
1 $GLOBALS['TL_DCA']['tl_beispiel']['list']['operations']['testaktion'] = array(  
2   'label'     => &$GLOBALS['TL_LANG']['tl_beispiel']['testaktion'],  
3   'href'     => 'key=reset',  
4   'icon'     => 'system/modules/demoErweiterung/html/standardIcon.png',  
5   'button_callback' => array('demoErweiterungsKlasse', 'makeIconMethode')  
6 );
```

Wichtig ist hier die Zeile

`'button_callback' => array('demoErweiterungsKlasse', 'makeIconMethode')`.

Durch die Methode `makeIconMethode` der Klasse `demoErweiterungsKlasse` wird also das Icon erstellt. Hier eine Auflistung der Parameter:

<code>\$arrRow</code>	the current row
<code>\$href</code>	the url of the embedded link of the button
<code>\$label</code>	label text for the button
<code>\$title</code>	title value for the button
<code>\$icon</code>	url of the image for the button
<code>\$attributes</code>	additional attributes for the button (fetched from the array key "attributes" in the DCA)
<code>\$strTable</code>	the name of the current table
<code>\$arrRootIds</code>	array of the ids of the selected "page mounts" (only in tree view)
<code>\$arrChildRecordIds</code>	ids of the childs of the current record (only in tree view)
<code>\$bInCircularReference</code>	determines if this record has a circular reference (used to prevent pasting of an cutted item from an tree into any of it's childs).
<code>\$strPrevious</code>	id of the previous entry on the same parent/child level. Used for move up/down buttons. Not for root entries in tree view.
<code>\$strNext</code>	id of the next entry on the same parent/child level. Used for move up/down buttons. Not for root entries in tree view.

(Danke an Xtra aus dem Contao-Forum, der diese Liste im Forum gepostet hat.)

Contao erwartet von der Methode die Rückgabe eines Links, der wie folgt aufgebaut werden kann.

```
1 return '<a href="'. $this->addUrl($href). "'  
2 title="'. $this->specialchars($title). "''. $attributes.'>'  
3 . $this->generateImage($icon, $label).  
4 '</a> ';
```

Vorher legt man für `$icon` den Pfad zum gewünschten Icon fest.

Beispiel:

Möchte man für jede gerade Id ein anderes Icon darstellen schreibt man in die Callback-Methode vor dem return:

```
1 if($arrRow %2 == 0){  
2     $icon = 'system/modules/demoErweiterung/html/anderesIcon.png';  
3 }  
4  
5 return '<a href="'. $this->addUrl($href). "'  
6 title="'. $this->specialchars($title). "''. $attributes.'>'  
7 . $this->generateImage($icon, $label).  
8 '</a> ';
```

Für die ungeraden Id's wird das Standardicon verwendet. Natürlich lassen sich beliebige Eigenschaften der Tabellenzeile analysieren und nicht nur das Icon, sondern auch das Linkziel lässt sich anpassen. Hier sind der Phantasie keine Grenzen gesetzt.