



Alkacon Software GmbH & Co. KG

An der Wachsfabrik 13
DE - 50996 Köln

Amtsgericht Köln
HRA 32185

Vertreten durch:
Alkacon Verwaltungs GmbH

Geschäftsführer:
Alexander Kandzior

Amtsgericht Köln
HRB 88218

Tel: +49 (0)2236 3826 - 0
Fax: +49 (0)2236 3826 - 20

Email: info@alkacon.com

Web: <http://www.alkacon.com>

Alkacon Software GmbH & Co. KG

Dokumentation

Eigenes CSS und JavaScript einbinden

Erstellt am: Donnerstag, 18. April 2019

Version: 1.0

Für die

(Erz)Bistümer Aachen, Bamberg und Mainz

1 Übersicht

Das für die (Erz-)Bistümer der Community verwendete „Apollo“ Template ermöglicht es, eine CSS-Datei einzubinden und CSS-Klassen an Elementen zu setzen. Dadurch können eigene Styles gezielt auf Elemente angewendet werden.

Außerdem können Sie eine eigene JavaScript-Datei einbinden, um zusätzliche interaktive Funktionen bereitzustellen.

2 Warnung

Das für die (Erz-)Bistümer der Community verwendete „Apollo“ Template beinhaltet bereits eine umfangreiche CSS und JavaScript-Konfiguration. Die Dateien hierfür werden von Alkacon Software im Template-Modul verfügbar gemacht.

Durch das Hinzufügen eigener CSS / JavaScript Dateien können Sie diese Vorgabe-Konfiguration umfangreich anpassen.

Alkacon Software garantiert NICHT, dass nach einem Update des Template-Moduls von Ihnen gemachte individuelle CSS / JavaScript Anpassungen weiterhin funktionieren.

Technisch ausgedrückt können sich sowohl die HTML-Struktur einer Seite als auch das CSS bzw. JavaScript in der Vorgabe-Konfiguration bei einem Update des Template-Moduls ohne Vorwarnung umfangreich ändern.

Alkacon Software führt nach einem Update keine Tests durch, ob Ihre individuellen CSS / JavaScript Anpassungen von diesen Änderungen betroffen sind.

Zu Testzwecken können individuelle CSS / JavaScript Anpassungen auf dem Testserver der (Erz-)Bistümer hinterlegt und dort vor dem Update des Live-Systems durch den Verantwortlichen für die Anpassung getestet werden. Bei Bedarf kann Alkacon Software dafür eine angepasste Site mit Inhalten von einem Live-System exportieren und in das Testsystem importieren.

3 Eigenes CSS einbinden

Erstelle Sie Ihr eigenes CSS zunächst in einem lokalen Editor als `.css`-Datei.

Lasen Sie diese Datei dann in den Start-Folder Ihrer Site bzw. Subsite hoch.

Der Dateiname im OpenCms muss `custom.css` sein.

3.1 Konfigurieren des eigenen Style-Sheets

Eine in den Start-Ordner einer Site hochgeladene Datei `custom.css` wird automatisch eingebunden.

Wenn Sie einen individuellen Style nur für eine Subsite setzen möchten, so müssen Sie dafür die Datei-Eigenschaft **apollo.template.css** am Start-Ordner der Subsite setzen. Geben Sie als Wert den Pfad zu Ihrer hochgeladenen `custom.css` CSS-Datei an. Der in der Eigenschaft angegebene Pfad muss immer mit „/“ beginnen und auch mit „/“ enden.

Beispiel: Sie haben eine Datei `custom.css` in den Ordner `/meine/subsite/` hochgeladen. Setzen Sie nun die Eigenschaft **apollo.template.css** am Ordner `/meine/subsite/` auf den Wert `/meine/subsite/`.

Hinweis: Es wird immer nur ein CSS eingebunden. Wenn Sie also am Start-Ordner der Site eine Datei `custom.css` hinterlegt haben, und im Ordner `/meine/subsite/` ebenfalls, so wird für den Ordner und alle Unterordner von `/meine/subsite/` NICHT das CSS aus dem Start-Ordner geladen.

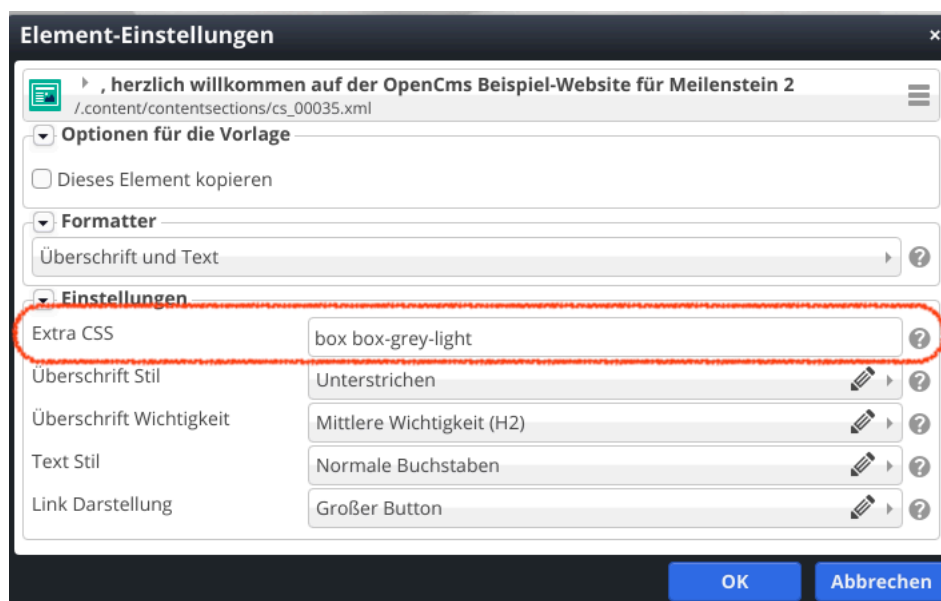
Sie können für eine Subsite das über den Start-Ordner der Site geladene CSS wieder entfernen. Setzen Sie hierfür die Eigenschaft **apollo.template.css** an dem Subsite-Ordner auf den Wert `none`.

Hinweis: Die Eigenschaft **apollo.template.css** kann in Prinzip an beliebigen Ordnern gesetzt werden, es ist nicht notwendig, dass dies tatsächlich ein Subsite-Ordner ist.

3.2 Setzen eigener CSS-Selektoren an Elementen

Das für die (Erz-)Bistümer der Community verwendete „Apollo“ Template erlaubt für viele Elemente auf der Seite eigene CSS-Selektoren zum erzeugten HTML hinzuzufügen.

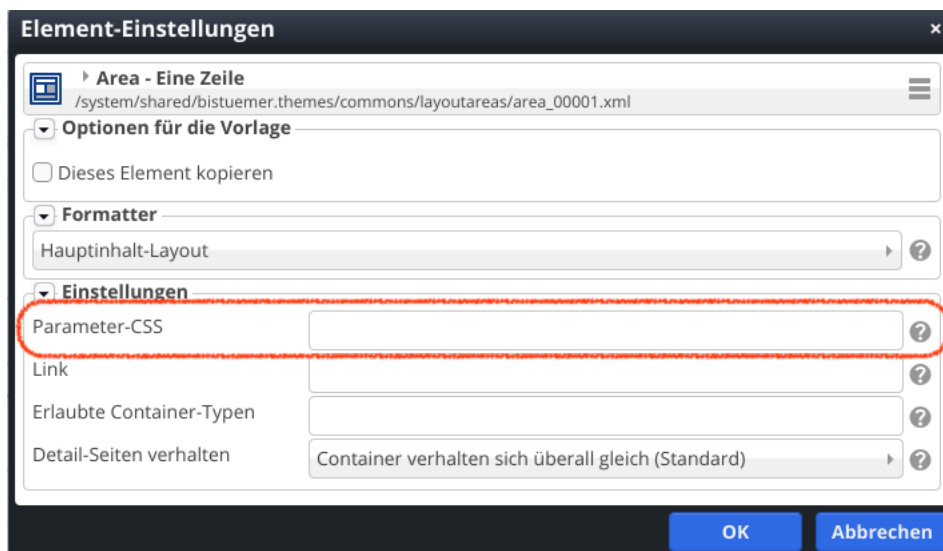
Nutzen Sie dazu die Element-Einstellung „Extra CSS“ – wie hier exemplarisch an einem Inhaltsabschnitt demonstriert:



Auf diese Weise können Sie Elemente auf der erzeugten Seite gezielt auszeichnen, um diese dann über Ihr eigenes CSS speziell zu formatieren.

3.3 Setzen eigener CSS-Selektoren in Layout-Bereichen / Zeilen

Für die Template-Elemente „Layout-Bereich“ und „Layout-Zeile“ heißt die Element-Einstellung „Parameter-CSS“ (anstatt „Extra CSS“).



Wichtig ist hier zu beachten: Die Einstellung wirkt sich nur aus, wenn der Bereich bzw. die Zeile ein eigenes Markup ausgibt und dort im „Pre-Markup“ des Inhalts „\$(param)“ an entsprechender Stelle genutzt wird. Z.B. so:



Hinweis: Bei den meisten vordefinierten Layout-Inhalten wird \$(param) im Pre-Markup entsprechend genutzt.

4 Eigenes JavaScript einbinden

4.1 Konfigurieren des eigenen JavaScript

Eine eigene JavaScript muss im Prinzip genau wie eine eigene CSS-Datei hochgeladen und eingebunden werden.

Nur muss die Datei `custom.js` genannt werden (anstatt von `custom.css`).

Für Subsites müssen Sie die Eigenschaft `apollo.template.js` entsprechend setzen (anstatt von `apollo.template.css`).

4.2 Verwenden von jQuery in eigenem JavaScript

Wenn ein eigenes JavaScript eingebunden wird, so kann dieses auf jQuery zugreifen, welches bereits durch das Template geladen wird.

Allerdings steht das jQuery Objekt bzw. \$ Objekt NICHT direkt im globalen window-Kontext zur Verfügung. Der Grund hierfür ist unter anderem, dass das Template JavaScript aus Performance-Gründen asynchron lädt.

Um Ihr eigenes JavaScript zu initialisieren verwenden Sie daher am besten die Methode `apollo.ready(handler)`

Benutzen Sie diese Methode anstatt `$document.ready(handler)`, so wird Ihre Methode sicher erst aufgerufen, wenn jQuery bereits geladen wurde.

An die Funktion `handler` werden zwei Parameter übergeben. Der erste Parameter ist die jQuery-Instanz, das zweite Parameter ist ein Boolescher Wert welcher „true“ ist, wenn der DEBUG-Modus aktiviert wurde.

Hier ein Beispiel für eine eigene Datei `custom.js` mit Verwendung von jQuery:

```
function customDemo($, DEBUG) {
    var $demobox = $(' .my-js-demo ');
    if ($demobox.length > 0) {
        if (DEBUG) console.info(".my-js-demo found");
        $demobox.on("mouseover", function(event){
            $demobox.css("background-color", "lightpink");
        });
    } else {
        if (DEBUG) console.info(".my-js-demo NOT found");
    }
}

apollo.ready(customDemo);
```

Der DEBUG-Modus für das JavaScript kann aktiviert werden, indem beim Aufruf einer Seite im Browser das Request-Parameter „jsdebug=true“ in der URL-Zeile angehängen wird.