bx:containerfilter

Mit dem Tag `containerfilter` kann man eine Containerliste (spezifiziert durch die Container-ID oder den Container-Namen) nach Bedingungen filtern und aktivierte Elemente, die zutreffen, in einer Schleife durchlaufen lassen. Der Tag-Inhalt wird dabei für jedes gefundene Element ausgeführt.

Falls {titel} angegeben ist, wird in der Administration ein Eingabefeld für die Filter-Bedingungen angezeigt (zu finden rechts in der Toolbox unter "Tag-Einstellungen"). Alternativ kann auch eine Bedingung direkt im Tag angegeben werden.

Dieses Tag kann unter *Tag-Einstellungen* konfiguriert werden.

mehrere Elemente (statische Angabe)

```
<bx:containerfilter.{titel} pool="(ContainerID | Containername)" force="list | top |
standalone | match" [max="{n}"] [orderby="{feldname}"] [desc] [startindex="{n}"] [sql]
[evalbx] [nulls-at-end] [id-descending-order] [force-parameter-reset] [loopable]>
  Inhalt
</bx:containerfilter.{titel}>
```

Die Parameter stehen statisch im Quelltext.

titel

frei wählbarer Name des Filters

pool

ID oder Name des Containers (ab Vers. 2.6.0)

force

Eine passende Wahl des Modus anhand der gewünschten Funktionen (einer oder mehrere Datensätze, Filterung ja/nein, Blättern ja/nein) trägt erheblich zur Performance-Steigerung bei, falls im Container sehr viele Datensätze vorhanden sind. So empfiehlt sich z.B. immer ein Modus ohne Blättern zu wählen, falls das Blättern nicht gebraucht wird. Die folgende Tabelle gibt Aufschluss über die verschiedenen Modi und zeigt jeweils, ob die Datensätze anhand der Filterung eingeschränkt werden bzw. ob ein Blättern möglich ist. **Achtung**: Fehlt die Modusangabe, wird bei Vorhandensein des Standard-Requestparameters `recordid` eine Detailausgabe (`force=single`) ausgegeben, ansonsten eine Liste (`force=list`). Eine explizite Angabe von `force` wird empfohlen!

list

Liste von Datensätzenfür Standard-Blätterlisten

top

Liste von Datensätzen z.B. zur Anzeige der 3 aktuellsten News als Teaser

single

ein einzelner Datensatz, Übergabe der ID im Request Detailseite mit Einzel-Blätterfunktion zu einer Standard-Blätterliste

match

ein einzelner Datensatz, Übergabe der ID im Request geeignet für eine Detailseite mit Berechtigungs-Checks

standalone

ein einzelner Datensatz, Übergabe der ID im Request funktioniert wie bx:record

max

maximale Anzahl der auszugebenden Datensätze (z.B. max="5")

orderby

Feldangabe, nach der die Ausgabe sortiert sein soll (Datum, Name etc.) siehe auch bei `nulls-at-end` oder`id-descending-orderNEU:`Der Parameter orderby nimmt jetzt mehrere Feldnamen, getrennt durch Komma, Leerzeichen, Semikolon oder Pipe an und es kann über eine Einzelverknüpfung sortiert werden (siehe Beispiel) (ab V2.7) Dem Feldnamen kann man ein Pluszeichen für aufsteigende Sortierung (A..Z) oder ein Minuszeichen für absteigende Sortierung (Z..A) voranstellen (wenn kein `+` oder `-` vorangestellt ist, wird nach `desc` geschaut)

startindex

Stelle, an der die Ausgabe beginnen soll (Beginn bei 0)

desc

dreht die Sortierrichtung um

sql evalbx

statt Filterangaben kann in den Tag-Einstellungen eine SQL angegeben werden (siehe Beispiel)
wenn angegeben, können in der SQL bx-Tags benutzt werden

nulls-at-end

null-Felder werden am Ende ausgegeben

id-descending-order

nach ID absteigend sortieren

force-parameter-reset

bewirkt, daß bei einer Containerfilter-Schachtelung der innere Container bei jedem Durchlauf des äußeren neu initialisiert wird (intern, Debugging) ab v.2.6.6

loopable

Filterangaben innerhalb eines Designloops

Beispiele

Containerfilter-Aufruf im Quelltext

```
<bx:containerfilter.Mitglieder pool="Mitglieder" force="list" max="10" orderby="Name" desc
startindex="5">
```

```
  <bx:recordfield.Name/><br>
</bx:containerfilter.Mitglieder>
```

Es werden maximal 10 Namen der im Container "Mitglieder" gespeicherten Datensätze ausgegeben. Die Datensätze sind nach Name (orderby) sortiert, allerdings absteigend (z-a), da "desc" angegeben wurde. Die Ausgabe startet mit dem 5. Datensatz. Filterungen können dann beim Menüpunkt unter "Tag-Einstellungen" (rechts unter "Tools") vorgenommen werden. Siehe auch "Filter".

Beispiel: Folgender Filter gibt nur die Mitglieder aus, die im Vorstand sind (Verknüpfung mit einem Container, der "Rollen" enthält über das Feld "Rolle", ID ist die DS-ID der Rolle "Vorstand"):

zugehöriger Filter

```
<filter>
  <filter-object>
    <field>Rolle</field>
    <type>8</type>
    <static-value>12FF89E7458</static-value>
  </filter-object>
</filter>
```

gleiches Beispiel wie oben, nur mit einer SQL:

Containerfilter-Aufruf im Quelltext

```
<bx:containerfilter.Mitglieder pool="Mitglieder" sql>
  <bx:recordfield.Name/><br>
</bx:containerfilter.Mitglieder>
```

zugehöriger Filter

```
SELECT m.*
FROM bxc_mitglieder m
WHERE
  m.ACTIV = 'j'
  AND m.Rolle = '12FF89E7458'
ORDER BY m.Name DESC
LIMIT 5,10;
```

Wieder gleiches Beispiel, allerdings wird die Rolle über den Request angegeben

Containerfilter-Aufruf im Quelltext

```
<bx:containerfilter.Mitglieder pool="Mitglieder" sql evalbx>
  <bx:recordfield.Name/><br>
</bx:containerfilter.Mitglieder>
```

zugehöriger Filter

```
SELECT m.*
FROM bxc_mitglieder m
WHERE
  m.ACTIV = 'j'
  AND m.Rolle = '<bx:pagedata.request name="rolle"/>'
ORDER BY m.Name DESC
LIMIT 5,10;
```

mehrere Sortierparameter: ** (ab V2.7)**

Der Parameter orderby nimmt jetzt mehrere Feldnamen, getrennt durch Komma, Leerzeichen, Semikolon oder Pipe an. Dem Feldnamen kann man ein Pluszeichen für aufsteigende Sortierung (A..Z) oder ein Minuszeichen für absteigende Sortierung (Z..A) voranstellen.

zugehöriger Filter

```
<bx:containerfilter.Name orderby="-Datum,+Titel"> <!-- wenn kein + oder - vorangestellt ist,
wird nach desc geschaut -->

<bx:containerfilter.Name orderby="-Datum,Titel" [asc]>
<bx:containerfilter.Name orderby="Datum,+Titel" desc>□□□□<!-- ergeben die gleiche Sortierung wi
das Beispiel oben -->

<bx:containerfilter.Name orderby="-Jahr -Monat -Tag">
<bx:containerfilter.Name orderby="Jahr Monat Tag" desc>        <!-- ergeben auch beide die
selbe Sortierung.
```

Sortierung über Einzelverknüpfung: (ab V2.7)Sortierung über Einzelverknüpfung ist auch für jedes Feld möglich:

zugehöriger Filter

```
<bx:containerfilter.Name orderby="Kategorie/Titel;-Preisgruppe/Nummer">
```

Dreimal anders und doch das Gleiche

zugehöriger Filter

```
<bx:containerloop orderby="-Datum,+Titel">
<bx:containerloop orderby="-Datum,Titel" [asc]>
<bx:containerloop orderby="Datum,+Titel" desc>
```

noch ein Beispiel

zugehöriger Filter

```
<bx:containerloop orderby="-Jahr -Monat -Tag"> ist das gleiche wie
<bx:containerloop orderby="Jahr Monat Tag" desc>
```

mehrere Elemente (dynamische Angaben im Request)

```
<bx:containerfilter.{titel} pool="(ContainerID | Containername)" [force="list"]
[maxparam="{param}"] [orderbyparam="{param}"] [directionparam="{param}"]
[indexparam="{param}"]>
  Inhalt
</bx:containerfilter.{titel}>
```

Die Parameter werden im Request übergeben.

{titel}

frei wählbarer Name des Filters

pool

ID oder Name des Containers (ab Vers. 2.6.0)

force

Ausgabemodus Liste

orderbyparam

Parametername, mit dem das Sortierfeld übergeben wird, kann auch gleich mit der Sortierrichtung gekoppelt werden, z.B. "+Name" und "-Name" für Sortierung nach Name aufsteigend bzw. absteigend **NEU**: reagiert jetzt auch auf mehrfach vorkommende Request-Parameter (siehe Beispiel) (ab V2.7)

directionparam

Parametername für die Sortierrichtung: + für aufsteigend, - für absteigend

maxparam

Parametername für die maximale Anzahl der auszugebenden Datensätze (pro Seite)

indexparam

Stelle, an der die Ausgabe beginnen soll (Standard: "index") **NEU:** wird nur noch beachtet wenn kein + (%2B) oder - im Request-Parameter vorangestellt sind (ab V2.7)

Beispiele

```
<bx:containerfilter.aktuelleAngebote pool="Artikel" force="list" maxparam="max"
orderbyparam="titel" directionparam="direction" indexparam="index">
  <bx:recordfield.Titel/> <bx:recordfield.Preis pattern="0.00"/><br>
</bx:containerfilter.aktuelleAngebote>
```

Diesen Filter findet man oft auf blätterbaren (Angebots-)Seiten. In diesem Beispiel werden alle Artikel mit Preis gelistet. Dann kann der Kunde selbst entscheiden (Form mit entsprechenden Auswahlmöglichkeiten - fehlt im Beispiel), wieviele Datensätze pro Seite angezeigt werden sollen (maxparam), nach welchem Feld sortiert werden soll (orderbyparam), z.b. Name, Preis, Datum, ob auf- oder absteigend sortiert sein soll (directionsparam).

Beim Blättern zwischen den Anzeigeseiten wird noch der Parameter "indexparam" gebraucht. Im Zusammenhang mit "maxparam" werden dann die jeweils pro Seite aufzulistenden Datensätze ermittelt.

```
<bx:containerfilter.Name orderbyparam="sort">
```

Mit dem Query-String ?sort=Name&sort=Vorname wird nun nach beiden Feldern sortiert. (ab V2.7)
Der Parameter directionparam wird nur noch beachtet wenn kein + (%2B) oder - im Request-Parameter vorangestellt sind.

ein bestimmtes Element (über Request)

```
<bx:containerfilter.{titel} pool="(ContainerID | Containername)" force="single"
idfield="{param}" [dummy]>
  Inhalt
</bx:containerfilter.{titel}>
```

Der Inhalt des Tags wird dann nur ausgegeben, wenn das gesuchte Element in der Ergebnisliste auftaucht, außer man gibt "dummy" an.

{titel}

frei wählbarer Name des Filters

pool

ID oder Name des Containers (ab Vers. 2.6.0)

force

Ausgabemodus Einzelausgabe

idfield

Parametername, der die ID des Datensatzes enthält (Standard: "recordid")

dummy

bei Fehlen des Parameters wird ein Datensatz simuliert (z.B. beim Erstaufruf von Formularen)

Beispiele

```
<bx:containerfilter.heimischeSingvoegel pool="Singvoegel" force="single" idfield="vogelid">
  <h1><bx:recordfield.Titel/></h1>
  <img src="<bx:recordfield.Bild width="250" height="180" type="path"/>"
alt="<bx:recordfield.Titel/>">
  <bx:recordfield.Beschreibung/>
</bx:containerfilter.heimischeSingvoegel>
```

Typische Detailseite. Der Requestparameter "vogelid" enthält die ID des gewünschten Datensatzes.

Möglicher Aufruf für dieses Beispiel: Elster

ein zufälliges Element

```
<bx:containerfilter.{titel} pool="(ContainerID | Containername)" force="random" [max="{n}"]>
  Inhalt
</bx:containerfilter.{titel}>
```

Es wird ein zufälliges Element aus der Ergebnisliste ausgewählt. Ab Version 2.6.2 hat man auch die Möglichkeit, mehrere Random-Datensätze anzeigen zu lassen.

{titel}

frei wählbarer Name des Filters

pool

ID oder Name des Containers (ab Vers. 2.6.0)

force

Ausgabemodus Random (zufälliges Element)

max

Anzahl der auszugebenden Datensätze (ab Version 2.6.2) Standard: 1

Beispiele

```
<bx:containerfilter.Kopfbild pool="Bilder" force="random">
  <div class="kopfbild">
    <img src="<bx:recordfield.Bild width="850" height="2000" type="path"/>"
alt="<bx:recordfield.Titel/>">
  </div>
</bx:containerfilter.Kopfbild>
```

Bei jeden Seitenaufruf wird ein anderes Kopfbild angezeigt.

Verknüpfte Tabelle filtern

```
<bx:containerfilter pool="(ContainerID | Containername)" base="{Anzeige-ID}"
linkfield="{Verknüpfungsfeld}">
```

Durch diese Erweiterung kann auch eine mit der Haupttabelle verknüpfte Tabelle gefiltert werden.

pool

ID oder Name des Containers (ab Vers. 2.6.0)

base

ID des verknüpften Container (Name geht leider nicht)

linkfield

Name des Feldes in der Haupttabelle, das mit der Anzeige-Tabelle verknüpft ist

Beispiele

```
<bx:containerfilter.ArtikelKategorien pool="Artikel" base="0123456789" linkfield="Kategorie">
  <a href="artikelliste.htm?kat=<bx:recorddata.id/>"><bx:recordfield.Titel/></a><br>
</bx:containerfilter.ArtikelKategorien>
```

Es sollen nur Kategorien (im Container mit der ID "0123456789") aufgelistet werden, mit der auch Artikel (über das Feld "Kategorie") verknüpft sind. Kategorien ohne Artikel werden ignoriert. Allerdings werden auch deaktivierte Kategorien angezeigt, wenn der entsprechende Artikel aktiv ist.

Bedingung im Tag

```
<bx:containerfilter pool="(ContainerID | Containername)" name="{Feldname}"
type="{Vergleichsart}" value="{Vergleichswert}" [required] [dummy]>
  Inhalt
</bx:containerfilter>
```

WICHTIG: Der Filter darf keinen Bezeichner haben. Es können auch die Parameter force, max, orderby, startindex angegeben werden (siehe oben bei `force`).

{titel}

frei wählbarer Name des Filters

pool

ID oder Name des Containers (ab Vers. 2.6.0)

name

Name des zu überprüfenden Feldes (siehe auch "Filter")

type

Art des Vergleiches (siehe auch "Filter")

value

gesuchter Wert (siehe auch "Filter") **statisch**: einfach gesuchten Wert angeben (z.B. eine ID)

dynamisch: mit Präfix (siehe auch "Präfixe")

required

wenn nicht vorhanden, dann wird nichts ausgegeben (ab Version 2.6.1)

dummy

im Detailmodus: wird kein Datensatz gefunden, wird ein Datensatz simuliert (Inhalt wird ausgegeben) Details dazu oben bei `force`

Beispiele

```
<bx:containerfilter pool="Bilder" name="Kategorie" type="8" value="1234567890"
orderby="Titel">
  <img src="<bx:recordfield.Bild width="850" height="2000" type="path"/>"
alt="<bx:recordfield.Titel/>">
</bx:containerfilter>
```

Alle Datensätze aus dem Container "Bilder", die im Feld "Kategorie" mit dem Datensatz mit der ID "1234567890" verknüpft sind, werden angezeigt.

Bedingung im Clipboard

```
<bx:containerfilter pool="(ContainerID | Containername)" force="list"
config="clipboard:config">
  Inhalt
</bx:containerfilter>
```

> **WICHTIG:** Der Filter darf keinen Bezeichner haben.

{titel}

frei wählbarer Name des Filters

pool

ID oder Name des Containers (ab Vers. 2.6.0)

force

Ausgabemodus Liste

config

dynamische Konfiguration

Beispiele

```
<bx:clipboard.cut name="config" trim>
  <filter>
    <filter-object required="false">
      <field>Invalid</field>
      <type>4</type>
      <static-value>0</static-value>
    </filter-object>
    <filter-object required="false">
      <field>Anbieter</field>
      <type>8</type>
      <special-value>PARENTCONTAINER</special-value>
    </filter-object>
  </filter>
</bx:clipboard.cut>
```

Die Bedingung wird dynamisch zusammengestellt und im weiteren Verlauf für eine Container-Filterung genutzt.

Bedingung in Systemeinstellungen

```
<bx:containerfilter.KontaktFilter pool="Kontakte" force="list" orderby="Name"
config="system:filter_kontakte">
...
</bx:containerfilter.KontaktFilter>
```

Die Filterbedingungen werden in den Systemeinstellungen gespeichert.