

tagManager 2.0

<http://modx-shopkeeper.ru/>

=====

Компонент для фильтрации товаров в каталоге и управления фильтрами.

Функции:

- Управление фильтрами. Изменение порядка сортировки, переименование названий групп фильтров и др.
- Групповое изменение значений доп. параметров товаров.
- Простая фильтрация товаров в каталоге (без ajax).
- Ajax-фильтрация товаров в каталоге с выводом количества найденных товаров по каждому фильтру.
- Динамическое изменение вида вывода товаров (изменение шаблонов).
- Поддержка мультивалютности.

=====

tmFilters - сниппет выводит фильтры. Вызывать кэшированным (без "!").

tmCatalog - сниппет выводит товары каталога и фильтрует. Вызывать не кэшированным (с "!").

tmRefresh - плагин для обновления значений фильтров при сохранении документа.

Необходимо установить сниппет getProducts (<http://modx.com/extras/package/getproducts>).

=====

Сниппет tmFilters

Параметры сниппета:

filterOuterTpl - Чанк блоки с фильтрами. По умолчанию - tm2_filterOuterTpl.

filterTpl - Чанк одного фильтра. По умолчанию - tm2_filterTpl.

filterNumericOuterTpl - Чанк блока с фильтрами с числовыми значениями. По умолчанию - tm2_filterOuterTpl.

filterNumericTpl - Чанк фильтра с числовыми значениями. По умолчанию - tm2_filterNumericTpl.

jsMap - Генерация JS-карты данных товаров для возможности вывода количества найденных товаров по каждому фильтру (1 - да, 0 - нет). По умолчанию: 0.

toPlaceholder - Имя плейсхолдера для вывода. Если не задано, выводится в месте вызова сниппета.

style - Подключить CSS-файл стилей (1 - да, 0 - нет). По умолчанию: 1.

jsScript - Подключить все необходимые JS-скрипты (1 - да, 0 - нет). По умолчанию: 1.

filtersType - Тип фильтрации. Возможные значения: filters, filters_simple. По умолчанию: filters.

categoryId - ID документа откуда нужно брать сохраненные фильтры. По умолчанию - текущий.

Используется библиотека jQuery.

Если у вас на сайте уже используется jQuery, повторно она подключаться не будет (сделана автоматическая проверка).

Для тонкой настройки открыть в текстовом редакторе файл

/assets/components/tag_manager2/js/web/filters.js

или

/assets/components/tag_manager2/js/web/filters_simple.js

Настройки вверху, в коде есть комментарии.

Если используется ajax-фильтрация (filtersType = filters), в форму фильтрации добавить скрытое поле (см. пример 3):

```
<input type="hidden" name="page_id" value="[*id]" disabled="disabled" />
```

Параметр "filters_type" в filters.js

Для "filters_type" = "default" (показ числа товаров по каждому фильтру и блокирование пустых вариантов)

или "only_block" (только блокирование пустых вариантов) необходимо включить параметр "jsMap" у сниппета "tmFilters".

Если "filters_type" = "none", "jsMap" рекомендуется выключить.

Внимание!

При аякс-фильтрации все параметры для getProducts (через getPage) должны указываться в наборе параметров.

Имя используемого набора параметров нужно указать в настройках системы - tag_mgr2.propertySetName.

Этот набор параметров должен быть привязан к сниппету getPage. Сниппет getPage должен использовать getProducts (element=getProducts).

Так же имя набора параметров можно указать в параметрах шаблона категории товаров (см. ниже).

=====

Сниппет tmCatalog

Сниппет не имеет параметров, т.к. является оберточным сниппетом для "getPage" и "getProducts".

Используются параметры из набора параметров для сниппета "getPage", указанного в настройках системы - tag_mgr2.propertySetName.

Можно указывать параметры для "getPage" и "getProducts".

Пример:

```
[[!tmCatalog?
&parents=-1`
&resources=`110,111,112`
&tpl=`product`
]]
```

В данном случае будут выводиться только товары с ID = 110,111,112.

Можно использовать для поиска, выводить найденные ID товаров с помощью отдельного сниппета и фильтровать найденные товары.

=====

Вывод товаров из таблицы, созданной в MIGXDB (<http://modx.com/extras/package/migx>)

Если для хранения товаров используется отдельная таблица в БД, созданная с помощью MIGXDB, в настройках компонента "tagManager2" в параметрах "tag_mgr2.className" и "tag_mgr2.packageName" указать название класса и название компонента вашей таблицы в БД.

Настройки "className" и "packageName" можно указать в параметрах шаблона категории товаров:

1. Открыть шаблон категории товаров.
2. Перейти на вкладку "Параметры".
3. Разблокировать параметры по умолчанию и добавить параметры:

prodClassName - Имя класса таблицы БД (например "ShopContent").

prodPackageName - Имя пакета таблицы БД (например "shop").

prodPropertySetName - имя набора параметров для снippetsа getPage, который нужно использовать для фильтрации для текущего шаблона (не обязательный параметр).

4. Сохранить изменения.

Внимание!

Обязательно укажите все ID шаблонов товаров через запятую в настройках системы в параметре "tag_mgr2.prod_templates".

=====

=====

Множественные значения фильтров

Если нужно сделать возможность задавать множественные значения для параметра, нужно создать TV параметр с типом ввода "Флажки (checkbox)" или "Список (множественный выбор)".

Поле "Возможные значения" заполнить по такому принципу:

акция==#акция# | лидер продаж==#лидер продаж# | новинка==#новинка#

Разделитель "#" можно изменить в конфигурации компонента - tag_mgr2.guard_key.

Имена таких TV нужно указать в скрипте filters.js в массиве

'multitags': ['tag']

А также имена полей с множественными значениями нужно указать в настройках системы в параметре "tag_mgr2.multitags" (можно несколько через запятую).

=====

=====

Динамическое изменение шаблона вывода списка товаров

На данный момент сделана готовая возможность изменения шаблона с помощью JS-скрипта /assets/components/tag_manager/js/web/view_switch.js.

Скрипт ставит куку и обновляет вывод.

1. Добавить скрипт на страницу

```
<script type="text/javascript" src="/assets/components/tag_manager2/js/web/view_switch.js"></script>
```

2. В набор параметров "getPage", который используется для вывода товаров, добавить параметр "tpl_list" со списком шаблонов (чанков) через запятую.

Например: "product,product_list".

3. В тело страницы вставить ссылки для изменения вида. У элемента-контейнера должен быть id="viewSwitch".

Пример:

```
<span id="viewSwitch">
    Показывать:
    &nbsp;
    <a rel="nofollow" href="#" class="active">картинками</a>
    &nbsp;
    <a rel="nofollow" href="#">списком</a>
</span>
```

```
=====
=====
```

Мультивалютность

Поддерживается мультивалютность, используемая Shopkeeper. По настройке мультивалютности читать документацию Shopkeeper.

Для работы tagManager с мультивалютностью необходимо выполнить:

1. Если используется аякс-фильтрация, открыть в текстовом редакторе /assets/components/tag_manager2/js/web/filters.js

и включить мультивалютность - multi_currency = true.

2. В этом же файле указать правильное название TV параметра или поля с ценой - price_field.

3. В настройках системы в параметре tag_mgr2.priceName нужно указать название TV параметра или поля с ценой (по умолчанию "price").

4. Версия Shopkeeper должна быть не меньше 2.3.4.

=====

=====

Стилизация селектов

Для стилизации селектов рекомендуется использовать jQuery-плагин SelectBoxIt - <http://gregfranko.com/jquery.selectBoxIt.js/index.html>.

Данный плагин поддерживается tagManager.

Пример подключение стиля и скриптов SelectBoxIt:

```
<link href="/assets/template/js/jquery.selectBoxIt.js/src/stylesheets/jquery.selectBoxIt.css" type="text/css" rel="stylesheet" />
```

```
<script src="http://code.jquery.com/ui/1.10.4/jquery-ui.min.js" type="text/javascript"></script>
```

```
<script src="/assets/template/js/jquery.selectBoxIt.js/src/javascripts/jquery.selectBoxIt.min.js" type="text/javascript"></script>
```

```
<script type="text/javascript">
```

```
$(document).ready(function(){
```

```
    $("select").selectBoxIt();
```

```
});
```

```
</script>
```

=====

=====

Автоматическое обновление списков значений фильтров

tmRefresh - плагин для обновления фильтров.

События: OnDocFormSave, OnCacheUpdate

По умолчанию данный плагин отключен. Для активации найти плагин в списке и нажать "Активировать плагин".

Если для плагина включено событие "OnCacheUpdate", при активации в админке "Сайт" -> "Обновить сайт", автоматически обновятся

все списки значений фильтров. Если товаров очень много (больше 3000), это действие может занимать продолжительное время. В этом случае

рекомендуется отключить событие "OnCacheUpdate" и использовать cron - скрипт `/assets/components/tag_manager2/cron_updatefilters.php`.

Основной файл конфигурации cron обычно находится по адресу: `/etc/crontab`

Пример для запуска ежедневно в 02:00 ч.:

```
00 02 * * * php -f '/var/www/modxsite/assets/components/tag_manager2/cron_updatefilters.php' > /dev/null
```

Внимание!

Если используется плагин "tmRefresh" с включенным событием "OnCacheUpdate", то удаление значений тегов работать не будет, т.к. при обновлении они снова будут появляться.

Если нужно удалять, то данное событие плагина нужно отключить или просто вместо удаления делать деактивацию значения фильтра.

```
=====
```

```
=====
```

JS-callback функции в filters.js

При использовании аякс-фильтрации (filter.js) есть возможность использовать callback функции:

`tm_onFilterBefore( state_data )` - перед отправкой запроса на фильтрации. `state_data` - данные активных фильтров.

`tm_onFilterAfter( total, pageCount, onPageLimit )` - после фильтрации. `total` - общее число товаров; `pageCount` - число страниц; `onPageLimit` - число товаров на одной странице.

Просто создайте функцию и она будет вызвана в нужный момент.

Пример:

```
<script type="text/javascript">
```

```
function tm_onFilterAfter( total, pageCount, onPageLimit ){
```

```
    $('#total').text(total);
```

```
}  
</script>
```

В данном примере в элементе с id="total" при фильтрации будет обновляться число найденных товаров.

```
=====
```

```
=====
```

Подключение JS-скриптов вручную.

```
[[tmFilters?  
&jsScript=`0`  
...
```

Необходимые JS-скрипты и CSS-файлы:

```
<link type="text/css" href="/assets/components/tag_manager2/css/web/tm-style.css" rel="stylesheet">  
<script type="text/javascript" src="/assets/components/tag_manager2/js/web/jquery-1.10.2.min.js"></script>  
<script type="text/javascript" src="/assets/components/tag_manager2/js/web/jquery-ui-  
1.10.3.custom.min.js"></script>  
<script type="text/javascript" src="/assets/components/tag_manager2/js/web/jquery.history.js"></script>  
<script type="text/javascript" src="/assets/components/tag_manager2/js/web/filters.js"></script>
```

```
<script type="text/javascript">  
//Filters initialization  
$(document).ready(function(){ tmFilters.init(); } );  
</script>
```

```
=====
```

```
=====
```

Пример 1. Простая фильтрация с чекбоксами (множественный выбор).

```
[[tmFilters?
```

```
&filterOuterTpl=`tm2_filterOuterTpl`  
&filterTpl=`tm2_filterTpl`  
&filterNumericOuterTpl=`tm2_filterOuterTpl`  
&filterNumericTpl=`tm2_filterNumericTpl`  
&filtersType=`filters_simple`  
&toPlaceholder=`filters`  
]]
```

```
<div id="filters">  
  <form action="[[~[[*id]]]" method="get">  
  
    <table>  
      <tr>  
        <td>  
          Сортировать по:  
        </td>  
        <td>  
          <select class="f_sortby" name="sortby" style="width:100px;">  
            <option value="pagetitle">названию</option>  
            <option value="price">цене</option>  
            <option value="publishedon">дате</option>  
          </select>  
        </td>  
      </tr>  
      <tr>  
        <td>  
          В порядке:  
        </td>  
        <td>  
          <select class="f_sortdir" name="sortdir" style="width:100px;">  
            <option value="asc">возрастания</option>  
            <option value="desc">убывания</option>  
          </select>  
        </td>  
      </tr>
```

```
<tr>

  <td>

    Выводить по:

  </td>

  <td>

    <select class="f_limit" name="limit" style="width:100px;">

      <option value="10">10</option>

      <option value="20">20</option>

      <option value="40">40</option>

    </select>

  </td>

</tr>

</table>
```

[[+filters]]

```
<button type="submit">Подобрать</button>

<button type="button" onclick="window.location.href = window.location.pathname; return
false;">Сбросить</button>

</form>

</div>
```

=====

```
<div id="products">

  [[!tmCatalog]]

  <div class="clearfix"></div>

</div>
```

```
<ul class="pages" id="pages">

  [[!+page.nav]]

</ul>
```

=====

=====

Пример 2. Сортировка отдельно от формы фильтров. Простая фильтрация с фильтрами в виде выпадающих списков (одиночный выбор).

```
<div id="filters">

  <form action="[~[[*id]]]" method="get">

    <input type="hidden" name="sortby" value="pagetitle" />
    <input type="hidden" name="sortdir" value="asc" />
    <input type="hidden" name="limit" value="10" />

    [[tmFilters?
      &filterOuterTpl=`tm2_filterOuterTpl_select`
      &filterTpl=`tm2_filterTpl_select`
      &filterNumericOuterTpl=`tm2_filterOuterTpl`
      &filterNumericTpl=`tm2_filterNumericTpl`
      &filtersType=`filters_simple`
    ]]

    <button type="submit">Подобрать</button>
    <button type="button" onclick="window.location.href = window.location.pathname; return false;">Сбросить</button>

  </form>
</div>
```

=====

```
<div class="sorting">

  <select class="f_sortby" name="sortby">
    <option value="pagetitle">по названию</option>
    <option value="price">по цене</option>
    <option value="publishedon">по дате</option>
  </select>

  &nbsp;

  <select class="f_sortdir" name="sortdir">
```

```

        <option value="asc">по возрастанию</option>
        <option value="desc">по убыванию</option>
    </select>

    &nbsp;

    <select class="f_limit" name="limit">
        <option value="10">10</option>
        <option value="20">20</option>
        <option value="40">40</option>
    </select>

    <div class="clearfix"></div>
</div>

```

```

<div id="products">
    [[!tmCatalog]]
    <div class="clearfix"></div>
</div>

```

```

<ul class="pages" id="pages">
    [[!+page.nav]]
</ul>

```

=====

=====

Пример 3. Аяx-филътpация, сортировка отдельно от формы фильтров.

```

[[tmFilters?
&filterOuterTpl=`tm2_filterOuterTpl`
&filterTpl=`tm2_filterTpl`
&filterNumericOuterTpl=`tm2_filterOuterTpl`
&filterNumericTpl=`tm2_filterNumericTpl`
&jsMap=`1`
&toPlaceholder=`filters`
]]

```

```
<div id="filters">

  <form action="[~[[*id]]]" method="get">

    <input type="hidden" name="page_id" value="[[*id]]" disabled="disabled" />

    [[+filters]]

    <button type="button" onclick="tmFilters.resetFilters(); return false;">Сбросить</button>

  </form>

</div>
```

=====

```
<div class="sorting">

  <select class="f_sortby" name="sortby" onchange="tmFilters.changeOrder(this);">
    <option value="pagetitle">по названию</option>
    <option value="price">по цене</option>
    <option value="publishedon">по дате</option>
  </select>

  &nbsp;

  <select class="f_sortdir" name="sortdir" onchange="tmFilters.changeOrder(this);">
    <option value="asc">по возрастанию</option>
    <option value="desc">по убыванию</option>
  </select>

  &nbsp;

  <select class="f_limit" name="limit" onchange="tmFilters.changeOrder(this);">
    <option value="4" selected="selected">4</option>
    <option value="10">10</option>
    <option value="20">20</option>
    <option value="40">40</option>
  </select>

<div class="clearfix"></div>
```

</div>

<div id="products">

[[!tmCatalog]]

<div class="clearfix"></div>

</div>

<ul class="pages" id="pages">

[[!+page.nav]]

