Как я и говорил ранее, мы не будем искать легких путей. Вообще говоря, верхнее меню можно сделать, используя такие известные сниппеты, как Ditto, Wayfinder... но ведь это не для нас ;).

Мы же сделаем все с нуля. Зачем нам (мне) это понадобилось, ведь есть уже готовые варианты? Потому что мне так больше нравится. Потому что я считаю подобный подход наиболее верным. Потому что, умея разрабатывать свое, мы без труда сможем разобраться в чужом коде, изменить, исправить ошибки или дополнить его необходимым в конкретном случае функционалом. Да и, в конце концов, сделать свое – это зачастую так приятно!

Дальнейшее повествование будет подразумевать, что читатель обладает минимальным знанием PHP программирования. Теоретически те люди, которые совсем не понимают код, смогут скопировать код и воспроизвести все те действия, о которых пойдет речь ниже. Однако, возможно, для них будет лучше использовать готовые сниппеты а-ля Ditto, поскольку они предоставляют кучу возможностей для внешнего конфигурирования, не влезая в основной код.

Я не хочу дискутировать на тему, что лучше для пользователя – брать готовое или разрабатывать свое... И в том, и в другом случае есть свои плюсы и минусы. Лично для меня плюсов во втором варианте больше. Также и каждый из читателей определится сам.

Итак, взглянем вновь на наш шаблон . Поскольку разных программных частей в нашем сайте довольно много, а начинать с чего-то нужно, поэтому запрограммируем…

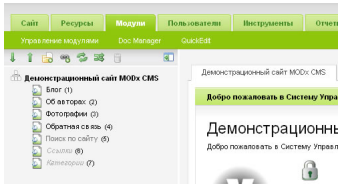
Верхнее меню

Под термином "верхнее меню" я понимаю набор ссылок на страницы в верхней части сайта (см. рисунок ниже):

Заранее обращу внимание, что активная ссылка в этом меню выделяется на белом фоне оранжевым цветом. В данном случае на рисунке выделена закладка "Блог".

Первый пример создания сниппета я опишу очень подробно, в дальнейшем я буду останавливаться в основном на наиболее существенных деталях.

Сравним с нашим деревом сайта в системе управления, которое мы построили в предыдущей статье:



Как видно из рисунка, в дереве сайта выделяются четыре документа (а именно "Блог", "Об авторах", "Фотографии" и "Обратная связь"), которые и создадут впоследствии ссылки в верхнем меню.

Также напомню, мы заранее скрыли документы, которые не хотим показывать в меню. Например, в настройках документа с названием "Поиск по сайту" убрали флажок "Показывать в меню", а оставшиеся два документа "Ссылки" и "Категории" мы скрыли на сайте, убрав флажок в настройках документа "Публиковать" (закладка "Настройки страницы" в настройках документа).

Таким образом, мы еще в предыдущей статье подготовили верхнее меню в системе управления.

Перейдем теперь к вопросу отображения наших действий непосредственно на сайте.

Большую часть программ на сайте выполняют т.н. " сниппеты ", т.е. отдельные куски кода (их также можно понимать как отдельные функции или подпрограммы) на PHP. Поэтому, чтобы реализовать в MODx вывод верхнего меню, мы также должны создать новый сниппет, запрограммировать его и добавить вызов этого сниппета в шаблоне в нужном месте.

Зайдем в систему управления, откроем закладку "Ресурсы" -> "Управление ресурсами" -> закладка "Сниппеты" и нажмем на ссылку "Новый сниппет". В поле "Название сниппета" впишем "TopMenu" без кавычек и пока просто сохраним пустой сниппет без кода. После сохранения мы увидим название нашего сниппета на закладке "Сниппеты".

Напомню, что в нашем шаблоне верхнее меню мы вынесли в чанк "TOPMENU". Переключимся на закладку "Чанки" и откроем чанк " TOPMENU ". В содержимом этого чанка мы увидим следующий код:

```
1.
2. <ul class="box">
3.     <li id="active"><a href="#">Блог<span class="tab-l"></span><span class="tab-r"></span></a></li>
<!-- Active -->
4.     <li><a href="#">Об авторах<span class="tab-l"></span><span class="tab-r"></span></a></li>
5.     <li><a href="#">Фотографии<span class="tab-l"></span><span class="tab-r"></sp
```

```
an
></
a
></
li
>
    6.      <li><a href="#">Обратная связь<span class="tab-l"></span><span class="tab-r"></
span
></
a
></
li
>
    7.  </ul>
    8.
```

Этот код как раз и создает наше меню. Закомментируем его и добавим вызов сниппета "TopMenu" в чанке:

```
    1.
    2.  <!--
    3.  <ul class="box">
    4.      <li id="active"><a href="#">Блог<span class="tab-l"></span><span class="tab-r"></
span
></
a
></
li
>
    5.      <li><a href="#">Об авторах<span class="tab-l"></span><span class="tab-r"></span
></
a
></
li
>
    6.      <li><a href="#">Фотографии<span class="tab-l"></span><span class="tab-r"></sp
an
></
a
></
li
>
    7.      <li><a href="#">Обратная связь<span class="tab-l"></span><span class="tab-r"></
```

span
></
a
></
li
>

8.
9. -->
10. [[TopMenu]]
11.

Здесь снова придется немного отвлечься, чтобы пояснить смысл этой конструкции [[TopMenu]]. Дело в том, что именно с помощью конструкций вида [[SnippetName]] или [!SnippetName!] мы можем делать вызовы сниппетов в любом месте своих шаблонов, чанков и даже просто на любой отдельной странице сайта.

При этом конструкция [[SnippetName]] обозначает вызов *кэшируемого* сниппета, т.е. вызов динамической подпрограммы, результат выполнения которой будет подсчитан и выполнен один раз, а впоследствии при загрузке страницы, где вызывается данный сниппет, результат будет неизменным, т.к. повторный вызов сниппета уже не происходит. Таким образом, мы экономим ресурсы своего веб-сервера (а это всегда имеет очень важное значение при высокой посещаемости ресурса).

Однако существуют ситуации, когда необходимо все время выполнять код сниппета заново и кэшировать результаты нельзя. В таких случаях используется конструкция [!SnippetName!], которая всегда заставит сниппет выполняться без кэширования. По аналогии, данная конструкция называется вызовом *некэшируемого* сниппета.

Итак, сохраним чанк "TOPMENU" и обновим страницу сайта. Хм, как ни странно, но верхнее меню исчезло. Но так ли это удивительно на самом деле? Закомментировав HTML код меню в чанке, мы скрыли его отображение в браузере (проверьте это, взглянув в исходный код HTML страницы). А наш сниппет "TopMenu" ничего не делает, поскольку в него еще ничего не добавлено. Исправим же этот недостаток :).

Перейдем снова на закладку "Сниппеты", откроем созданный сниппет "TopMenu" и попробуем протестировать его возможности… Терпение, мои продвинутые читатели,

не всем знакомы эти детали.

Для начала напишем простейший код (обычный PHP код):

```
1.
2. <?php
3. echo "Testing…";
4. ?>
5.
```

Перед сохранением выберем "Продолжить редактирование", т.к. нам придется еще не раз изменить содержимое нашего сниппета, и после этого сохраним сниппет.
Обновим страницу сайта и увидим на месте верхнего меню… ну, по правде говоря, на первый взгляд мы не увидим почти никаких изменений, кроме слегка расширившегося синего фона меню. Нажмем "CRTL+A", чтобы выделить весь текст на странице сайта, и увидим, что все-таки наш сниппет вывел на месте меню текст "Testing…", просто цвет текста совпадает с цветом фона.

Изменим код сниппета на следующий:

```
1.
2. <?php
3. echo "<span style='color:#FFFFFF;'>Testing...</span>";
4. ?>
5.
```

Теперь мы ясно видим, что сниппет наш работает и даже (!) выводит некоторый текст. Что-ж, это прекрасно, но маловато для нашей задачи, поскольку мы должны добиться, чтобы наш сниппет выводил ссылки из системы управления, причем в точно таком же HTML коде, который мы закомментировали в чанке "TOPMENU".

И снова небольшое отвлечение...

Вся система взаимосвязей документов в MODx построена по принципу: каждый "родительский документ" содержит от нуля до множества "дочерних документов" ("parent" –> "childs").

Каждый документ в базе данных MODx имеет свой уникальный идентификатор "ID" – это то число, которое мы видим в скобках в дереве сайта рядом с каждым из документов.

Кстати говоря, этот уникальный идентификатор несет лишь одно единственное значение – он однозначно определяет конкретный документ в системе управления *и ничего более*
! Специально делаю акцент на этом факте, поскольку встречал неоднократные попытки изменить эти идентификаторы в самых разнообразных целях... Запомнить нужно сразу, что это просто бессмысленно, поэтому не пытайтесь менять эти цифры. На них вообще не стоит обращать много внимания, обычно эти цифры используются просто для генерации ссылок на определенные документы.

В базе данных MODx для каждого документа также создано специальное поле "parent". Значением данного поля является число, обозначающее либо уникальный идентификатор родительского документа, либо, если документ находится в корне дерева, нуль. Таким образом, всегда можно однозначно определить, какой именно документ является для данного родительским.

Чтобы наглядно увидеть то, о чем мы сейчас говорили, откройте phpMyAdmin, выберите свою базу данных и найдите таблицу {PREFIX}site_content, где {PREFIX} – Ваш префикс, который Вы ввели при установке. Вы увидите множество полей, в которых сохраняются определенные данные документов, в том числе "ID" – уникальный идентификатор, "parent" – номер родительского документа, "pagetitle" – заголовок страницы и другие.

Итак, используя данную информацию о принципе хранения и связи документов в MODx, мы можем понять, как получить нужные данные для вывода ссылок верхнего меню: нам нужно найти в базе данных все документы, которые находятся в корне дерева сайта, т.е. имеют в поле "parent" значение нуль.

Используя SQL язык, подобный запрос описывается как-то так (Вы можете попробовать ввести данный запрос в поле ввода SQL в phpMyAdmin, предварительно заменив "modx_" на свой префикс):

```
1.
2. SELECT *
3. FROM `modx_site_content`
4. WHERE `parent` = 0;
5.
```

Однако такой запрос возвратит нам абсолютно все документы из корня сайта, что не совсем правильно, исходя из основной задачи – вывести ссылки только на те документы, которые имеют:

1. опубликованный статус (в БД за этот пункт отвечает поле "published", где значение = 1 обозначает, что документ опубликован, а значение = 0 - неопубликован).
2. неудаленные (поле "deleted", где 1 - удален, а 0 – не удален),
3. и у которых установлена опция "Показывать в меню" (поле "hidemenu", где 1 – скрывать, а 0 – показывать в меню).

Кроме того, забегая немного вперед, мы сразу отсортируем документы по параметру "Позиция в меню", который будет определять позицию каждой ссылки в нашем меню.

Ну, с точки зрения SQL, это совсем несложная задача и решается она так:

```
1.
2. SELECT *
3. FROM `modx_site_content`
4. WHERE `published` = 1
5. AND `parent` = 0
6. AND `deleted` = 0
7. AND `hidemenu` = 0
8. ORDER BY `menuindex` ASC;
9.
```

Теоретически все SQL запросы можно выполнять в сниппетах напрямую с помощью PHP скриптов, подключая каждый раз базу данных заново и делая множество других рутинных операций, повторяя их раз за разом… Но, согласитесь, это нивелировало бы смысл использования фреймворка, коим безусловно является наша система управления, т.к. MODx, помимо прочих своих достоинств, предоставляет готовый набор средств программного интерфейса (API, Application Programming Interface). API – это программные функции, которые унифицируют и облегчают многие процессы обработки данных.

Используем одну из упомянутых функций API " getDocumentChildren " в нашем сниппете. Функция "getDocumentChildren" получает в виде параметров следующие данные:

- $id – номер родительского документа,
- $active – выбирать только опубликованные или неопубликованные документы (1 или 0 соответственно),
- $deleted - выбирать только удаленные или неудаленные документы (1 | 0),
- $fields – поля, которые выбираются из БД,
- $where – специальные условия, т.е. условие WHERE в SQL запросе,
- $sort – поле, по которому должна проводиться сортировка результатов
- $direction – направление сортировки, может принимать значения ASC или DESC, т.е. сортировка от меньшего к большему значению или наоборот
- $limit – ограничение запроса, т.е. условие LIMIT в SQL запросе

```
1.
2.  <?php
3.
4.  $results = $modx->getDocumentChildren(
5.      $id = 0,
6.      $active = 1,
7.      $deleted = 0,
8.      'id, pagetitle, published, menuindex, deleted, hidemenu, menutitle',
9.      $where = 'hidemenu = 0',
10.      $sort='menuindex',
11.      $dir='ASC',
12.      $limit
13.  );
14.
15.   print ("<pre style='color:#FFFFFF;font-size:12px;'>");
16.
17.  foreach($results as $key => $value) {
18.       print_r ($value);
```

```
19. }
20.
21.  print ("</pre>");
22. ?>
23.
```

Сохраните сниппет и обновите страницу. В результате выполнения обновленного сниппета "TopMenu" Вы увидите список из массивов и их значений, отсортированный по значениям поля "menuindex" от меньшего значения к большему. Попробуйте поменять параметр $dir='ASC' на $dir='DESC' – в результате массивы перестроятся и первым документом будет выведен документ с наибольшим значением поля "menuindex".

Программистам со стажем, наверное, понятно, что полученный результат уже дает все, что нужно, чтобы построить готовое меню со ссылками. Ну, почти все. В любом случае я таки продолжу: перепишем код PHP, чтобы максимально приблизиться к желаемому результату.

```
1.
2. <?php
3.
4. $results = $modx->getDocumentChildren(
5.     $id = 0,
6.     $active = 1,
7.     $deleted = 0,
8.     'id, pagetitle, published, menuindex, deleted, hidemenu, menutitle',
9.     $where = 'hidemenu = 0',
10.     $sort='menuindex',
11.     $dir='ASC',
12.     $limit
13. );
14.
15. $items = "";
16. $output = "";
17.
18. foreach($results as $key => $value) {
19.     $items .= "<li>
20.     <a href="#">".$value["pagetitle"]."
21.     <span class="tab-l"></span><span class="tab-r"></span></a></li>n";
22. }
23.
```

```
24. if ($items != "") {
25.     $output = "<ul class="box">n";
26.     $output .= $items;
27.     $output .= "</ul>n";
28. }
29.
30. return $output;
31.
32. ?>
33.
```

Детально код разбирать не будем, поскольку это самый обычный PHP код, не более того. Если кому-то не понятен смысл каких-либо конструкций, прошу в комментарии, а лучше – на соответствующие форумы поддержки.

Сохраним новый код сниппета и обновим страницу. В результате выполнения кода мы увидим практически то, что и хотели получить:

Т.е. это уже автоматически сгенерированные ссылки, структура которых полностью повторяет структуру документов в дереве MODx. Чтобы проверить это, попробуйте создать какой-нибудь тестовый документ в корне сайта и обновите страницу.

Однако это еще не все. Многие уже наверняка заметили, что ссылки есть, но ссылок нет... Парадокс :). Я имею ввиду, что названия документов в меню выводятся, однако ссылки на них не работают. Это логично, поскольку пока в коде ссылок выводится "#" вместо реальных путей.

Чтобы решить эту задачку, необходимо узнать еще об одной крайне полезной возможности MODx: адрес любой внутренней страницы сайта можно получить с помощью следующей конструкции [~id~], где id – это уникальный номер нужного документа, т.е. тот самый номер, указанный в скобках рядом с названием каждого

документа в дереве сайта. Таким образом, добавив такую конструкцию [~1~] в шаблоне/чанке/содержимом страницы,

1. если у нас включены дружественные ссылки в настройках MODx, то при выводе мы получим:
1. index – алиас документа "Блог", если мы ввели "index" как алиас документа, либо
2. 1.html, если мы не вводили ничего в поле "Псевдоним" для документа "Блог"

3. если дружественные ссылки отключены, то увидим текст index.php?id=1

Перепишем сниппет, используя эту информацию:

```
1.
2. <?php
3.
4. $results = $modx->getDocumentChildren(
5.     $id = 0,
6.     $active = 1,
7.     $deleted = 0,
8.     'id, pagetitle, published, menuindex, deleted, hidemenu, menutitle',
9.     $where = 'hidemenu = 0',
10.     $sort='menuindex',
11.     $dir='ASC',
12.     $limit
13. );
14.
15. $items = "";
16. $output = "";
17.
18. foreach($results as $key => $value) {
19.     $items .= "<li>
20.     <a href="[~".$value["id"]."~]">".$value["pagetitle"]."
21.     <span class="tab-l"></span><span class="tab-r"></span></a></li>n";
22. }
23.
24. if ($items != "") {
25.     $output = "<ul class="box">n";
26.     $output .= $items;
27.     $output .= "</ul>n";
28. }
```

```
29.
30. return $output;
31.
32. ?>
33.
```

Таким образом, мы изменили # на [~".$value["id"]."~], т.е. фактически для каждого документа из массива подставляется его уникальный ID внутри конструкции [~id~]. В результате мы получаем меню с работающими ссылками.

Мы практически достигли идеала... Однако и теперь еще остается одна деталь, которую нужно обязательно учесть: дизайнер определил, что активная ссылка у нас должна быть подсвечена белым фоном и цвет ссылки соответственно должен быть изменен на оранжевый.

Чтобы добиться этого, мы снова приоткроем секреты MODx CMS :). В API скрыта функция $modx-> documentIdentifier , которая возвращает значение уникального идентификатора текущей страницы. Она нам понадобится для определения активной страницы и выделения ее в меню:

```
1.
2. <?php
3.
4. $results = $modx-> getDocumentChildren (
5.     $id = 0,
6.     $active = 1,
7.     $deleted = 0,
8.     'id, pagetitle, published, menuindex, deleted, hidemenu, menutitle',
9.     $where = 'hidemenu = 0',
10.     $sort='menuindex',
11.     $dir='ASC',
12.     $limit
13. );
14.
15. $cid = $modx->documentIdentifier;
16.
17. $items = "";
18. $output = "";
19.
```

```
foreach($results as $key => $value) {
    if ($value["id"] == $cid) {
        $active = " id="active"";
    }
    else {
        $active = "";
 }
    $items .= "<li".$active.">
    <a href="[~".$value["id"]."~]" title="".$value["pagetitle"]."">".$value["pagetitle"]."
    <span class="tab-l"></span><span class="tab-r"></span></a></li>n";
}

if ($items != "") {
    $output = "<ul class="box">n";
    $output .= $items;
    $output .= "</ul>n";
}

return $output;

?>

```

Ну как, получилось? Получилось!

Но Вы же не подумали, что на этом все и закончится? И правильно. Мы ставим себе самую высокую планку, мы хотим задействовать максимум возможностей MODx. А поэтому еще одна небольшая деталь, которую мы упустили.

Посмотрим внимательно на название полей, которые мы запрашиваем с помощью функции getDocumentChildren: 'id, pagetitle, published, menuindex, deleted, hidemenu, menutitle'. Среди них есть такое поле, как "menutitle". Как следует из названия, в данном поле может храниться *заголовок меню*. В системе управления также имеется поле ввода "Пункт меню". Это поле заполнять необязательно. Однако логика в том, что если это поле заполнено, то мы должны заменить текст ссылки в меню на

введенный пользователем. Ну, так сделаем это:

```
1.
2. <?php
3.
4. /*******************************
5. Название: TopMenu
6. Цель: Вывод верхнего меню
7. Проект: Демосайт MODx
8. *******************************/
9.
10. $results = $modx->getDocumentChildren(
11.     $id = 0, // ID родительского документа
12.     $active = 1, // Выбираем только опубликованные документы
13.     $deleted = 0, // Выбираем только неудаленные документы
14.     'id, pagetitle, published, menuindex, deleted, hidemenu, menutitle', // Выбираем поля из БД
15.     $where = 'hidemenu = 0', // Выбираем только те документы, которые нужно публиковать в меню
16.     $sort='menuindex', // Сортируем документы по полю menuindex
17.     $dir='ASC', // Сортируем документы по возрастанию
18.     $limit = '' // Ограничения не устанавливаем (параметр LIMIT в SQL запросе)
19. );
20.
21. $cid = $modx->documentIdentifier; //получаем ID текущей страницы
22.
23. $items = "";
24. $output = "";
25.
26. foreach($results as $key => $value) {
27.     if ($value["id"] == $cid) {
28.         $active = " id="active"";
29.     }
30.     else {
31.         $active = "";
32.     }
33.     if ($value["menutitle"] != "") {
34.         $title = $value["menutitle"];
35.     }
36.     else {
37.         $title = $value["pagetitle"];
38.     }
39.     $items .= "<li".$active.">
40.     <a href="[~".$value["id"]."~]" title="".$title.""">".$title."
41.     <span class="tab-l"></span><span class="tab-r"></span></a></li>n"; //собираем
```

пункты меню

```
42. }
43.
44. // Если удалось найти хотя бы один пункт меню,
45. // создаем HTML код меню
46. if ($items != "") {
47.     $output = "<ul class="box">n";
48.     $output .= $items;
49.     $output .= "</ul>n";
50. }
51.
52. // Возвращаем результат работы сниппета
53. return $output;
54.
55. ?>
56.
```

Попробуйте теперь ввести какой-нибудь текст в поле ввода "Пункт меню" любого документа... Все работает? Замечательно!

P.S.: Возможно, некоторые читатели будут удивлены, что при переходе по ссылкам нашего меню содержимое страниц не изменяется, хотя вроде бы, судя по пути в адресе браузера, мы переходим на новые страницы… Поверьте, это абсолютно нормально, т.к. абсолютно все страницы на текущий момент используют один и тот же шаблон. В этом шаблоне фактически мы пока сделали динамическим только верхнее меню, все остальные детали остаются неизменными. Мы обязательно займемся этим позже, а пока – без паники ;).

Статья позаимствована с официального сайта Modx