Автор: А.Волос
29.04.2011 04:01 - Обновлено 04.05.2011 03:07

Установочный файл **mod_mvc.xml** собственно устанавливает дерево каталогов для модуля
и поля для формы, с которой будут сниматься данные, введенные пользователем.

Следует обратить внимание на типы полей в форме:

type="filelist"
type="text"
type="section"
type="category"
type="radio"

Вообще первое, что делает функция getList() класса MVC - это считывает данные из вышеуказанной формы. Данные, которые ввел пользователь.
Из каждого поля, используя при этом его имя и функцию get():

```
$db =& JFactory::getDBO();
$source_sec = $params->get('source_sec');
$source_cat = $params->get('source_cat');
$show_all = $params->get('show_all');
$count = $params->get('count');
```

Далее подготавливается запрос, через который будут получены все данные для статьи:

```
if($show_all == 0){ $where = 'sectionid='.$source_sec.' AND catid='.$source_cat.' AND '; }else{ $where = ''; }
//этот запрос позволяет получить все необходимые праметры для статьи,
//чтобы сформировать на нее ссылку и пр.
$query = 'SELECT a.*, s.title as stitle, c.title as ctitle, u.name, ' . ' CASE WHEN CHAR_LENGTH(a.alias) THEN CONCAT_WS(&quot;:&quot;, a.id, a.alias) ELSE a.id END as slug,'.
' CASE WHEN CHAR_LENGTH(c.alias) THEN CONCAT_WS(&quot;:&quot;, c.id, c.alias) ELSE c.id END as catslug'.
```

```
  ' FROM #__content AS a' .
  ' LEFT JOIN #__users AS u ON u.id = a.created_by' .
  ' INNER JOIN #__categories AS c ON c.id = a.catid' .
  ' INNER JOIN #__sections AS s ON s.id = a.sectionid' .
  ' WHERE '.$where.
  ' s.published = 1' .
  ' AND c.published = 1' .
  ' ORDER BY id DESC';
```

Структуру этого запроса стоит рассмотреть подробнее.

```
  ---------------------------------
integer CHAR_LENGTH(str string)
Функции возвращают длину строки str. Поддерживает многобайтовые символы.
Примеры:
 mysql> select CHAR_LENGTH('text');
       -> 4
-------------------
mysql> SELECT CASE 1 WHEN 1 THEN 'one'
   ->     WHEN 2 THEN 'two' ELSE 'more' END;
       -> 'one'
-----------------------------------------
CONCAT_WS(separator,str1,str2,...)
```

CONCAT_WS() stands for Concatenate With Separator and is a special form of CONCAT(). The first argument is the separator for the rest of the arguments.
The separator is added between the strings to be concatenated.
The separator can be a string, as can the rest of the arguments.
If the separator is NULL, the result is NULL.

```
    mysql> SELECT CONCAT_WS(',','First name','Second name','Last Name');
       -> 'First name,Second name,Last Name'
mysql> SELECT CONCAT_WS(',','First name',NULL,'Last Name');
       -> 'First name,Last Name'
```

CONCAT_WS() does not skip empty strings. However, it does skip any NULL values after the separator argument.

```
SELECT table1.* FROM table1
    LEFT JOIN table2 ON table1.id=table2.id
    WHERE table2.id IS NULL;
```

Этот пример находит все строки в таблице table1 с величиной id, которая не присутствует в таблице table2 (т.е.
все строки в table1, для которых нет соответствующих строк в table2). Конечно, это предполагает, что
table2.id объявлен как NOT NULL.

CASE value WHEN [compare_value] THEN result [WHEN [compare_value] THEN result ...] [ELSE result] END

CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result] END

The first version returns the result where value=compare_value.
The second version returns the result for the first condition that is true.
If there was no matching result value, the result after ELSE is returned,
or NULL if there is no ELSE part.

```
mysql> SELECT CASE 1 WHEN 1 THEN 'one'
    ->     WHEN 2 THEN 'two' ELSE 'more' END;
        -> 'one'
mysql> SELECT CASE WHEN 1>0 THEN 'true' ELSE 'false' END;
        -> 'true'
mysql> SELECT CASE BINARY 'B'
    ->     WHEN 'a' THEN 1 WHEN 'b' THEN 2 END;
        -> NULL
```

The return type of a CASE expression is the compatible aggregated type of all return values,

but also depends on the context in which it is used. If used in a string context,
the result is returned as a string. If used in a numeric context, the result is returned
as a decimal, real, or integer value.

Значит выражение:

'CASE WHEN CHAR_LENGTH(a.alias) THEN CONCAT_WS(":", a.id, a.alias) ELSE a.id END as slug,'.
'CASE WHEN CHAR_LENGTH(c.alias) THEN CONCAT_WS(":", c.id, c.alias) ELSE c.id END as catslug'.

означает:

В СЛУЧАЕ ЕСЛИ значение поля a.alias имеет положительную длину (то есть поле alias заполнено данными)
ТОГДА строка будет построена так (a.id : a.alias)
ИНАЧЕ a.id КОНЕЦ как slug

То есть теперь в колонке slug содержатся примерно такие значения (строки): id=92:my-neghmye
в столбце catslug содержатся примерно такие значения (строки): catid=1:latest-news

То есть ссылка, формируемая таким способом:

```
//формируем ссылку
$lists[$i]->link = JRoute::_(ContentHelperRoute::getArticleRoute($row->slug, $row->catslug, $row->sectionid));
```

выглядит так:

index.php?option=com_content&view=article&id=92:my-neghmye&catid=1:latest-news&Itemid=18

```
//Выполняем запрос
//Если лимит не указан пользователем то выводим все
if($count == ''){	$db->setQuery($query);	}else{	$db->setQuery($query, 0, $count);
}
//Загружаем ответ в массив
$rows = $db->loadObjectList();
```

Вот такие переменные берутся из массива (полученной таблицы):

$row->slug - ИД : алиас статьи
$row->catslug - ИД : алиас категории
$row->sectionid - ИД секции
$row->introtext
$row->fulltext

Функция возвращает ассоц. массив, который содержит:
ссылку на статью $lists[$i]->link
ссылку на картинку $lists[$i]->image , которая так же является ссылкой на статью
интротекст статьи $lists[$i]->introtext
переданные в файл параметры $lists[$i]->paramss (эти параметры доступны по умолчанию)

И так в цикле для каждой статьи.
Хороший класс MVC определен в файле helper.php

JRoute::
JModuleHelper::
JText::
JFactory::
JURI::

Вообще надо попробовать создавать формы с различными полями!

JFactory::getDocument();
JFactory::getDBO();

CASE WHEN .. THEN .. ELSE .. END ..

Далее в файле mod_mvc.php фрагмент кода:

```
//Подключаем CSS
jimport('joomla.document.html.html');
$document =& JFactory::getDocument();//получим документ
$link = JURI::root().'modules/mod_mvc/tmpl/css/style.css'; //путь к нашему css-файлу
$attribs = array('type' => 'text/css');
$document->addHeadLink(JRoute::_($link), 'stylesheet', 'rel', $attribs);

//Подключаем шаблон (который указан из админке)
$template = JModuleHelper::getLayoutPath('mod_mvc', $tmpl);
if (file_exists($template)) {	require($template);	} else {	echo
JText::_('ERROR_TEMPLATE');
}
```

Функция addHeadLink() даст такой результат в заголовке html-страницы:

```
<link rel=&quot;stylesheet&quot; href=&quot;/templates/system/css/system.css&quot;
type=&quot;text/css&quot; />
<link rel=&quot;stylesheet&quot; href=&quot;/templates/system/css/general.css&quot;
type=&quot;text/css&quot; />
```

<link rel="stylesheet" href="/templates/rhuk_milkyway/css/template.css" type="text/css" />
<link rel="stylesheet" href="/templates/rhuk_milkyway/css/blue.css" type="text/css" />
<link rel="stylesheet" href="/templates/rhuk_milkyway/css/blue_bg.css" type="text/css" />

Для того, чтобы лучше разобраться как устроены модули, естественно надо посмотреть исходный
код и других модулей в joomla, а так же самому написать несколько новых модулей.

Вверх