



Разработка шаблона Joomla! Необходимый минимум

Мигачева Марина Евгеньевна
Минск, 2017



1. Общее
2. Виды шаблонов
3. Файловая система и `templateDetails.xml`
4. `index.php`
5. Добавление содержимого
6. Работа со стилями
7. Пример
8. Установка шаблона (если создавали локально)



Веб-страницы, которые вы видите в своем браузере состоят из кода HTML, который взаимодействует с другими файлами – изображениями, каскадными таблицами стилей, сценариями.

PHP – язык, предназначение которого **упрощать создание динамических страниц.**

В Joomla! базовую структуру кода определяет именно php – в частности **шаблонный файл index.php**



Перед тем, как начать создание своего шаблона, необходимо **представлять сам процесс работы данного компонента** в CMS.

Действие 1.

Система берет **файл шаблона index.php** и на первом этапе выход представляет собой **выполнение php-конструкций**;

все вне этих тегов (`<?php ... ?>`) также выполняется, но в остальном игнорируется (вывод буферизуется, но в клиентскую часть не выводится)



Действие 2.

Результат, который получили и буферизовали на первом шаге, теперь анализируется для элементов **<jdoc: include>**.

Для каждого найденного **элемента jdoc** загружается соответствующий класс визуализатора и вызывается его метод визуализации - результат из используемого метода заменяет сам элемент **<jdoc: include>** в выходном буфере и отображается в клиентской части



Виды шаблонов

Существуют **2 вида шаблонов** для Joomla:

- **front-end** (лицевая сторона)
- **back-end**(административная панель).

Рассмотрим **только front-end**, то есть ту часть сайта, которая создана для пользователей (ваших клиентов), а не для администраторов.

Все **front-end шаблоны** находятся в папке **/templates/**, в ней есть несколько папок, одна из которых **system** (её не трогаем), а все остальные папки – это и есть установленные нами (или по умолчанию) шаблоны.



Файловая система

ВАЖНО:

файловая система вашего шаблона по своей структуре максимально приближена к классической во front-end разработке.

Важным отличием для Joomla является наличие:

- **php-файла** как основного шаблона для страниц;
- возможных нескольких дополнительных **php-файлов**;
- обязательного **xml-файла**, который еще называют манифест-файл.

Следует учитывать, что в разных CMS логика будет несколько отличаться



ВАЖНО:

совсем не обязательно делать **все** описанные ниже шаги;

если структура шаблона вас устраивает, то можно изменять только стилевой файл;

если есть необходимость, то внести изменения в структуру body;

... и так далее... **все зависит от ваших задач**



Файловая система

Структура шаблона может отличаться от предложенной за исключением минимального перечня объектов

css

html

images

img

js

language

less

component.php

error.php

index.php

offline.php

templateDetails.xml

template_preview.png

template_thumbnail.png





Файловая система

ВАЖНО:

Для рассмотрения дальнейших действий **создадим копию шаблона protostar** и будем работать с ее структурой как стартовой, внося изменения в код и в структуру

The screenshot shows the Joomla! administrator interface for template management. A table lists available templates. The 'New' template is highlighted with a red rectangular box.

Изображение	Шаблон	Версия	Дата	Автор
	Параметры Beez3 Предварительный просмотр недоступен. Вы можете включить предварительный просмотр в настройках.	3.1.0	25 November 2009	Angie Radtke a.radtke@derauftritt.de http://www.der-auftritt.de
	Параметры New Предварительный просмотр недоступен. Вы можете включить предварительный просмотр в настройках.	1.0	4/30/2012	Kyle Ledbetter admin@joomla.org
	Параметры Protostar Предварительный просмотр недоступен. Вы можете включить предварительный просмотр в настройках.	1.0	4/30/2012	Kyle Ledbetter admin@joomla.org



Файловая система

Для этого выполняем последовательность действий – выбираем шаблон (кликом на названии) – Копировать шаблон – вводим имя нового шаблона – он становится доступен в общем списке – выбираем для редактирования

Менеджер шаблонов: Изменить шаблон

Копировать шаблон Предварительный просмотр шаблона Управление каталогами Новый файл Закрыть

Редактор Создать переопределение Описание шаблона

css
html
images
img

Выберите файл

Вам доступен целый ряд параметров для настройки в шаблонов поддерживает исходные файлы, файлы из



ВАЖНО:

созданные каталоги **не должны быть пустыми.**

Если на первых порах, Вы не добавляете в них файлы, то **создайте файл, к примеру index.html** (с минимальным содержимым, пустое body) **и поместите его во все пустые папки**, во избежание ошибок установки. Эти файлы являются заглушками и, по требованиям CMS, должны отображать пустую страницу при обращении к этим папкам напрямую.

// очистим все папки (удалим находящиеся в них файлы и подпапки, оставив только объекты корневого уровня и содержимое папки language)



Файловая система

//

очистим все папки (удалим находящиеся в них файлы и подпапки, оставив только объекты корневого уровня, графические объекты и содержимое папки language);

в пустые папки положим "пустой" файл index.html;

внесем изменения в оставшиеся файлы и заменим/дополним своими объекты;



Графические файлы

в корне шаблона расположены два графических файла - **превью будущего шаблона**, они имеют размеры 640x480 и 206x150 соответственно;

при создании своего шаблона необходимо их подготовить и разместить, чтобы ваш шаблон не был «безликим» в системе



templateDetails.xml

Для того, чтобы создаваемый шаблон работал, необходимо создать файл **templateDetails.xml** – в корне папки разрабатываемого шаблона.

Данный файл содержит важную информацию о шаблоне:

- общие данные,
- описание файловой структуры,
- описание структуры макета,
- описание языковых вариаций



Что входит в данный файл:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE install PUBLIC "-//Joomla! 2.5//DTD template 1.0//EN" "https://www.joomla.org/xml/dtd/2.5/template-install.dtd">
3 <extension version="3.1" type="template" client="site">
```

В этой части больше всего интересует часть **type="template"** и **client="site"**, так как мы работаем с шаблоном для клиентской части;

Версию можно прописать свою (если, к примеру, ваш шаблон работает только начиная с ветви 3x)



Блок 2:

<name>название шаблона**</name>**

<version>версия разработки**</version>**

<creationDate>дата создания**</creationDate>**

<author>имя автора**</author>**

<authorEmail>почта автора**</authorEmail>**

<copyright>данные по копирайту**</copyright>**

<description>краткое описание шаблона**</description>**



ВАЖНО:

обратите внимание, что описание задается специальной языковой константой

<description> краткое описание шаблона **</description>**

```
4 ▾ <name>new</name>
5 ▾ <version>1.0</version>
6 ▾ <creationDate>11/10/2017</creationDate>
7 ▾ <author>New Author</author>
8 ▾ <authorEmail>na@e-mail.e-mail</authorEmail>
9 ▾ <copyright>Copyright (C) 2017 Open Source</copyright>
10 ▾ <description>TPL_NEW_XML_DESCRIPTION</description>
```



templateDetails.xml

Блок 3:

в обертке **<files>** перечисляем все файлы, лежащие в корне папки шаблона (**filename**) и папки (**folder**)

```
11 ▾ <files>
12 ▾   <filename>component.php</filename>
13 ▾   <filename>error.php</filename>
14 ▾   <filename>offline.php</filename>
15 ▾   <filename>favicon.ico</filename>
16 ▾   <filename>index.php</filename>
17 ▾   <filename>templateDetails.xml</filename>
18 ▾   <filename>template_preview.png</filename>
19 ▾   <filename>template_thumbnail.png</filename>
20 ▾   <folder>css</folder>
21 ▾   <folder>html</folder>
22 ▾   <folder>images</folder>
23 ▾   <folder>img</folder>
24 ▾   <folder>js</folder>
25 ▾   <folder>language</folder>
26 ▾   <folder>less</folder>
27 ▾ </files>
```



ВАЖНО:

данный блок важен, так как при установке шаблона в CMS дает информацию о том какие файлы и каталоги будут скопированы в папку шаблона.

Соответственно тег `<folder>` определяет имя копируемой папки, а `<filename>` — имя копируемого файла. При этом указывается информация **только верхнего уровня**, вложенные файлы и папки, будут скопированы автоматически.



Блок 4:

описание позиций будет зависеть от структуры макета, который вы реализуете (речь идет не об общих решениях, а о конкретном под определенный макет)

ВАЖНО: в позициях будут размещаться модули, по этой причине детализация по количеству позиций может быть самой разной, к примеру, шапка, навигация, первый блок контента, две позиции под две колонки контента, футер



templateDetails.xml

К примеру,

описание позиций заключено в **<positions>**, где каждая из позиций описана в **<position>**

```
<positions>
  <position>position-0</position>
  <position>position-1</position>
  <position>position-2</position>
  <position>position-3</position>
  <position>position-4</position>
  <position>position-5</position>
  <position>footer</position>
</positions>
```



Блок 5:

необходимо добавить **информацию о поддерживаемых локализациях**, то есть другими словами требуется указать файлы, в которых будут содержаться **значения для используемых языковых констант**

```
<languages folder="language">
  <language tag="en-GB">en-GB/en-GB.tpl_new.ini</language>
  <language tag="en-GB">en-GB/en-GB.tpl_new.sys.ini</language>
  <language tag="ru-RU">ru-RU/ru-RU.tpl_new.ini</language>
  <language tag="ru-RU">ru-RU/ru-RU.tpl_new.sys.ini</language>
</languages>

</extension>
```



ВАЖНО: информация о доступных локализациях прописывается в блоке **<languages>**, то есть данный блок содержит пути к так называемым **файлам словарям**.

Данные файлы **содержат значения для языковых констант**, то есть по сути переводы.

Вместе с тегом, указывается **атрибут folder**, в котором прописано **имя каталога с локализациями**.



ВАЖНО:

в теге **languages** используется **атрибут folder**; дело в том, что языковые файлы для клиентской части и языковые файлы для административной части имеют одинаковые названия и поэтому не могут находиться в одной и той же директории пакета шаблона, соответственно, для их объявления необходимо указывать директорию



В Joomla, используется **два вида файлов для осуществления перевода**: обычные файлы формата **.ini** и файлы **sys.ini**.

Файлы .ini содержат данные о переводе форм, настроек или любые другие строки, которые нужно перевести на определенный язык.

Файлы sys.ini используются для отображения описания расширений во время установки или в менеджерах (шаблонов, компонентов и т.д.). Таким образом, для одного языка локализации нужны два вышеуказанных файла; **пути к ним прописываются в теге, который используется вместе с атрибутом tag, где указывается код языка конкретной локализации.**



templateDetails.xml

Опишем в данных языковых файлах формата **ini** следующие
КОНСТАНТЫ:

```
1 ; Joomla! Project
2 ; Copyright (C) 2017 Open Source.
3 ; License GNU General Public License version 2 or later; see LICENSE.txt, see LICENSE.php
4 ; Note : All ini files need to be saved as UTF-8
5
6 TPL_NEW_XML_DESCRIPTION="description on english"
7
8
9
```

Аналогичное содержание для остальных файлов (под другие языковые пакеты)



Опишем в данных языковых файлах формата **sys.ini** следующие
КОНСТАНТЫ:

```
1 |; Joomla! Project
2 |; Copyright (C) 2017 Open Source.
3 |; License GNU General Public License version 2 or later; see LICENSE.txt, see LICENSE.php
4 |; Note : All ini files need to be saved as UTF-8
5
6 TPL_NEW_POSITION_POSITION-0="Header"
7 TPL_NEW_POSITION_POSITION-1="Navigation"
8 TPL_NEW_POSITION_POSITION-2="Breadcrumbs"
9 TPL_NEW_POSITION_POSITION-3="Center"
10 TPL_NEW_POSITION_POSITION-4="Left Col"
11 TPL_NEW_POSITION_POSITION-5="Center Col"
12 TPL_NEW_POSITION_FOOTER="Footer"
13 TPL_NEW_XML_DESCRIPTION="description on english"
14
```

Аналогичное содержание для остальных файлов (под другие языковые пакеты); к примеру, на русском значение первой константы будет записано как «Шапка» и т.д.



ВАЖНО:

имя констант для позиций шаблона формируется по правилу:

префикс **TPL_** + имя шаблона **NEW** + строка **_POSITION_** + имя определенной позиции в верхнем регистре

TPL_NEW_POSITION_POSITION-0="Header"



Сделав активной нашу создаваемую тему, заданные настройки становятся видимыми и доступными, **к примеру,**

The screenshot shows the Joomla! administrator interface for configuring the 'Menu' module. The top navigation bar includes the Joomla! logo and the title 'Модули: Меню'. Below the navigation bar are several action buttons: 'Сохранить', 'Сохранить и закрыть', 'Сохранить и создать', 'Сохранить копию', 'Закреть', and 'Справка'. The main content area shows the configuration for the 'Main menu ru-RU' module. The 'Меню' section is active, and the 'Выбор меню' dropdown is set to 'Main Menu (ru-RU)'. The 'Базовый пункт меню' is set to 'Текущий', and the 'Начальный уровень' is set to '1'. The 'Последний уровень' is set to 'Все'. The 'Показывать подпункты меню' option is set to 'Нет'. A red circle highlights the 'Позиция' dropdown menu, which is currently set to 'position-7'. The dropdown menu shows a search bar and a list of available positions: 'Unused [position-13]', 'New', 'Footer [footer]', 'Position 0 [position-0]', 'Position 1 [position-1]', 'Position 2 [position-2]', 'Position 3 [position-3]', 'Position 4 [position-4]', 'Position 5 [position-5]', and 'Protostar'.

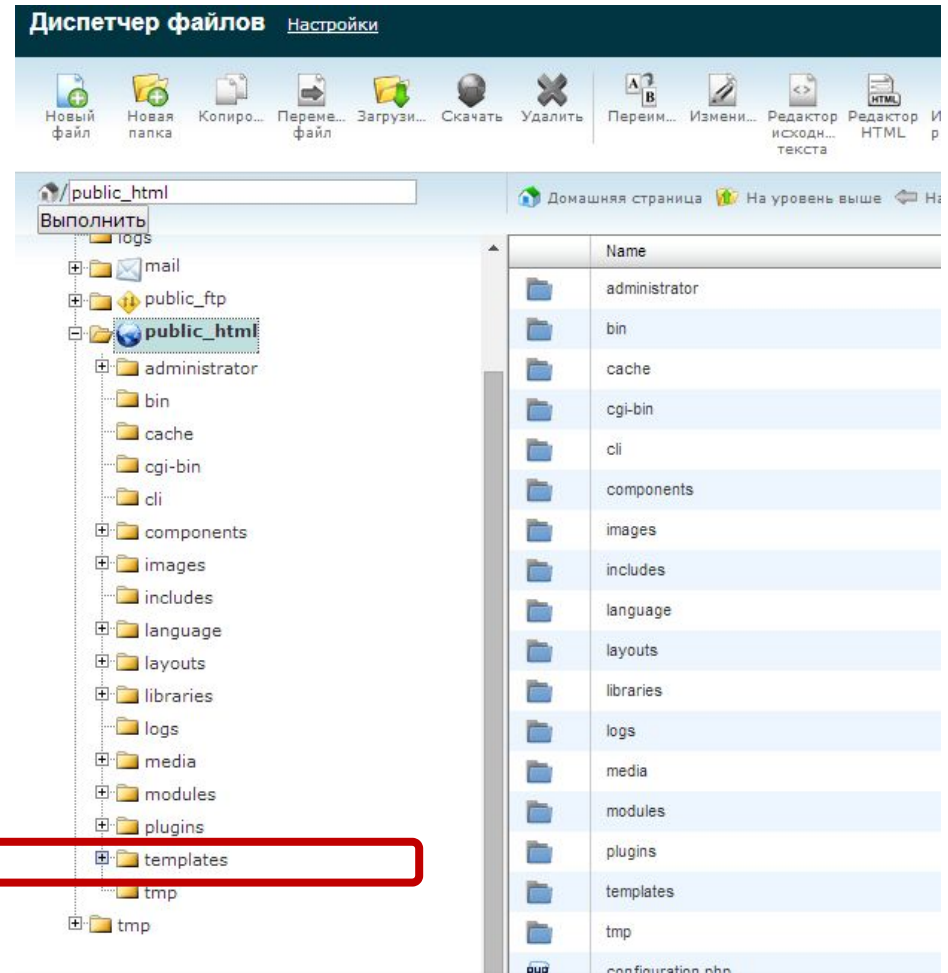


Пример. Дополнительно

Шаблон можно создавать и не опираясь на копию

Для того, чтобы создать базовую структуру шаблона, необходимо для начала

- перейти к папке /template своего сайта;
- создать внутри нее новую папку (скажем с именем ourtemplate).





Пример. Дополнительно

Диспетчер файлов Настройки Найти

Новый файл **Новая папка** Копировать... Переместить файл... Загрузить... Скачать Удалить... Переименовать... Изменить... Редактор исходного текста... Редактор HTML... Изменить разрешения... Вид Извлечь Сжать

/public_html/templates

Выполнить

Name	Size
bee3	4 KB
protostar	4 KB
system	4 KB
ind...	31 byte

Новая папка

Создать новую папку

Имя новой папки:

Новая папка будет создана в:

Create New Folder Cancel



Пример. Дополнительно

ВАЖНО: Используя возможности вашего сервера (панели управления, ftp-соединения), создайте требуемую структуру и необходимые файлы, что были рассмотрены ранее.

Работа с остальными файлами проводится аналогично.

Иначе говоря, **выбор способа за вами**, главное, соблюдать необходимые требования к структуре и содержанию.



файл index.php содержит описание логики, по которой будут формироваться страницы нашего ресурса, соответственно, его структура должна быть сформирована по требованию создаваемого макета и вопросам безопасности.

ВАЖНО: для детального изучения особенностей можно взять за основу любой базовый шаблон



ВАЖНО:

в файл index.php можно перенести код из файла index.html, а затем заменять поочередно статичные решения на соответствующие из CMS;

скопировать все остальные файлы и проверить пути, упоминаемые в атрибутах;

НО: в дальнейшем примере мы будем корректировать/заменять код, взятый из шаблона (для дополнительного ориентирования и сравнения)



ВАЖНО:

изучив структуру и содержимое файла index.php можно сказать о том, что решений много, они разнообразны и сложны; в первую очередь нас будет интересовать следующее:

- **формирование структуры страницы;**
- **подключение стилей;**
- **формирование head;**



Изучив содержимое файла index.html, взятого из шаблона, можно утверждать, что основа верхнего уровня осталась та же (найдите эти объекты в коде):

```
<!DOCTYPE html>  
<html>  
<head>  
  
</head>  
<body>  
  
</body>  
</html>
```



Однако, для формирования структуры страницы в шаблоне этого мало, добавим (или найдем в коде и оставим) **следующий фрагмент:**

```
<?php  
$doc = JFactory::getDocument();  
?>
```

ВАЖНО: обратите внимание, где в коде должен быть расположен данный компонент



класс конструктор **JFactory** и его статический **метод** **getDocument()** позволяет получить объект глобального класса `jDocumentSite`, который представляет документ, иначе - формирующуюся страницу.



- **формирование структуры страницы;**
- **подключение стилей;**
- **формирование head;**

создадим (или найдем в коде и оставим) **следующий фрагмент:**

```
<?php  
$doc = JFactory::getDocument();  
$doc->addStyleSheet(JUri::base().'templates/'. $doc->template.'/css/style.css');  
?>
```




Index.php

Этой строкой вызываем на исполнение у данного объекта **метод `addStyleSheet()`**, который регистрирует файл стилей к подключению, путь к которому передается в качестве первого аргумента.

```
<?php
$doc = JFactory::getDocument();
$doc->addStyleSheet(JUri::base(). 'templates/'. $doc->template. '/css/style.css');
?>
```



статический **метод `base()`**, **класса `JUri`** возвращает **базовый путь к сайту**, к которому мы добавляем имя папки с шаблонами (`templates`), **затем имя активного шаблона, которое хранится в свойстве `template` объекта `jDocumentSite`**, и конечно же, указываем каталог и имя подключаемого файла

```
<?php
$doc = JFactory::getDocument();
$doc->addStyleSheet(JUri::base(). 'templates/'. $doc->template. '/css/style.css');
?>
```

ИЛИ

```
<?php
$doc = JFactory::getDocument();
$doc->addStyleSheet(JUri::base(). '$this->template. '/css/style.css');
?>
```



ВАЖНО:

имеются иные варианты подключения стилей, к примеру,

```
<link rel="stylesheet"
href="<?php echo $this->baseurl ?> /templates/ourtemplate/css/template.css"
type="text/css" />
```

НО:

данная конструкция основывается на базовый url ресурса, что может быть неудобно, если формат будет, к примеру, не `http://мой_домен.com`, а `http://мой_домен.com/сайт`

```
<link rel="stylesheet"
href="<?php echo $this->baseurl ?>/templates/ourtemplate/css/template.css"
type="text/css" />
```



- **формирование структуры страницы;**
- **подключение стилей;**
- **формирование head;**

Для формирования содержимого head необходимо прописать следующую конструкцию:

```
<jdoc:include type="head" />
```



Тег **<jdoc:include />** с набором определенных атрибутов, используется в Joomla для подключения различных элементов сайта, **в данном случае на его месте будет подключено содержимое тега head, то есть зарегистрированные стили, кодировка страниц и тег title.**



ИТОГО:

- **формирование структуры страницы;**
- **подключение стилей;**
- **формирование head;**

```
<!DOCTYPE html>
<html>
<head>
  <?php
      $doc = JFactory::getDocument();
      $doc->addStyleSheet(JUri::base().'templates/'.$this->template.'/css/style.css');
  ?>
  <jdoc:include type="head" />
</head>
<body>

</body>
</html>
```



ИЛИ так:

```
<?php
    defined('_JEXEC') or die;
    $doc = JFactory::getDocument();
?>

<!DOCTYPE html>
<html>
<head>
    <?php
        $doc->addStyleSheet(JUri::base(). 'templates/'. $this->template. '/css/style.css');
    ?>
    <jdoc:include type="head" />
</head>
<body>

</body>
</html>
```



ИЛИ так:

```
<?php
    defined('_JEXEC') or die;
    $doc = JFactory::getDocument();
    $doc->addStyleSheet(JUri::base().'templates/'. $this->template.'/css/style.css');
?>

<!DOCTYPE html>
<html>
<head>
    <jdoc:include type="head" />
</head>
<body>

</body>
</html>
```




Здесь добавилась еще одна строка

```
defined( '_JEXEC' ) or die;
```

Это **стандартная проверка безопасности** Joomla, а именно, как это работает?



В файле **index.php в корневой директории** (где ставили пакет CMS) создается константа командой `define` с именем `"_JEXEC"` и значением `"1"`.

```
Viewer: index.php
Файл  Кодировка  Синтаксис  Поиск  Настройки
14  if (version_compare(PHP_VERSION, JOOMLA_MINIMUM_PHP, '<'))
15  {
16      die('Your host needs to use PHP ' . JOOMLA_MINIMUM_PHP . ' or higher to run this version of
17  }
18
19  // Saves the start time and memory usage.
20  $startTime = microtime(1);
21  $startMem = memory_get_usage();
22
23  /**
24   * Constant that is checked in included files to prevent direct access.
25   * define() is used in the installation folder rather than "const" to not error for PHP 5.2 and
26   */
27  define('_JEXEC', 1);
28
29  if (file_exists(__DIR__ . '/defines.php'))
30  {
31      include_once __DIR__ . '/defines.php';
32  }
33
34
```

Режим: php | utf-8 | Загружено 100% : 1420 из 1420 байт | READONLY | 1 : 0



В файлах подключаемых компонентов (в нашем случае это шаблон) **при помощи функции `defined("_JEXES")` проверяем существование константы `"_JEXES"`.**

Если константа не найдена результат работы функции будет `false` и в этом случае **останавливается скрипт с помощью функции `die`** (псевдоним функции `exit()`)



Можно использовать дополнительные переменные:
переменные приложения, пользователя, меню, языка

```
<?php
    defined('_JEXEC') or die;
    $doc = JFactory::getDocument();
    $doc->addStyleSheet(JUri::base().'templates/'.$this->template.'/css/style.css');

    $app = JFactory::getApplication();
    $user = JFactory::getUser();
    $menu = $app->getMenu();
    $lang = JFactory::getLanguage();
?>

<!DOCTYPE html>
```



Добавление содержимого

Пока остановим работу над head и перейдем к body:

для формирования позиции отображения контента используем **<jdoc:include />** с атрибутом **type="modules"**, а также необходимо использовать **атрибут name**, в котором будет указано имя создаваемой позиции

ВАЖНО: имена позиций указывали в установочном файле

ВАЖНО: для формирования **блока отображения основного контента**, необходимо в **атрибуте type**, указать **значение component**



Добавление содержимого

Иными словами, для функциональности шаблона нужны **позиции и модули**

Каждый модуль должен быть связан с определенной позицией, а именно:

```
<jdoc:include type="modules" name="position-1" />
```



Добавление содержимого

Исходя из тех позиций, которые мы предположили для нашего шаблона, можно создать **следующую основу для body**:

```
</head>
<body>
  <jdoc:include type="modules" name="position-0" />
  <jdoc:include type="modules" name="position-1" />
  <jdoc:include type="modules" name="position-2" />
  <jdoc:include type="modules" name="position-3" />
  <jdoc:include type="modules" name="position-4" />
  <jdoc:include type="component" />
  <jdoc:include type="modules" name="position-5" />
</body>
</html>
```



Добавление содержимого

```
1 <?php
2     defined('_JEXEC') or die;
3     $doc= JFactory::getDocument();
4     $doc->addStyleSheet(JUri::base().'templates/'.$doc->template.'/css/style.css');
5
6 ?>
7
8 <!DOCTYPE html>
9 <html>
10 <head>
11     <jdoc:include type="head" />
12 </head>
13 <body>
14     <jdoc:include type="modules" name="position-0" />
15     <jdoc:include type="modules" name="position-1" />
16     <jdoc:include type="modules" name="position-2" />
17     <jdoc:include type="modules" name="position-3" />
18     <jdoc:include type="modules" name="position-4" />
19     <jdoc:include type="component" />
20     <jdoc:include type="modules" name="position-5" />
21 </body>
22 </html>
23
24
```




ВАЖНО:

в одну и ту же позицию можно разместить **несколько** модулей, но полученная по умолчанию очередность может не подойти, значит, необходимо упорядочить расположение модулей в одной и той же позиции, **к примеру,**

The screenshot shows the Joomla! Modules Manager interface for the 'Модули (сайт)' section. The interface includes a toolbar with buttons for 'Создать', 'Изменить', 'Дублировать', 'Опубликовать', and 'Снять с публикации'. Below the toolbar, there are search and filter options. The main table lists modules with columns for 'Состояние', 'Заголовок', and 'Позиция'. A red box highlights the first two rows of the table:

Состояние	Заголовок	Позиция
<input checked="" type="checkbox"/>	Breadcrumbs	position-0
<input checked="" type="checkbox"/>	Language Switcher	position-0
<input checked="" type="checkbox"/>	Main menu en-GB	position-7
<input checked="" type="checkbox"/>	Main menu ru-RU	position-7



необходимо активировать колонку сортировки и, захватив мышью нужный модуль, перетянуть его выше или ниже

Система ▾ Пользователи ▾ Меню ▾ Материалы ▾ Компоненты ▾ Расширения ▾ Справка ▾

Модули (сайт)

Создать | Изменить | Дублировать | Опубликовать | Снять с публикации | Разблокировать

Порядок

Нажмите для сортировки по этой колонке

Состояние	Заголовок	Позиция	Тип
<input checked="" type="checkbox"/>	Breadcrumbs	position-0	Навигатор сайта
<input checked="" type="checkbox"/>	Language Switcher	position-0	Переключение языка
<input checked="" type="checkbox"/>	Main menu en-GB	position-7	Меню
<input checked="" type="checkbox"/>	Main menu ru-RU	position-7	Меню



Добавление содержимого

ВАЖНО:

не копируйте код (из презентации, web-страниц и прочего);

проверяйте написание дополнительно в профессиональных редакторах (подсветка ошибок);



ПРОВЕРКА В КЛИЕНТСКОЙ ЧАСТИ:

теперь можно проверить отображение контента в клиентской части

—

- [Вы здесь:](#)
- [Главная](#)

Главная

[Материал \(ru-ru\)](#)

—

- [Печать](#)
- [E-mail](#)

Подробности

Автор: Joomla
Категория: [Категория \(ru-ru\)](#)
Опубликовано: 22 ноября 2017
Просмотров: 2

Lorem ipsum ad his scripta blandit partiendo, eum fastidii accumsan euripidis in, eum liber h legimus senseritdefiniebas an eos. Eu sit tincidunt incorrupte definitionem, vis mutat affert pe fugit veritus placeratper. Ius id vidit volumus mandamus, vide veritus democritum te nec, ei e



Добавление содержимого

ВАЖНО:

так как на текущий момент мы не задавали никаких стилевых решений, то контент выведен линейно и с применением стилевых решений браузера

Прежде чем сбросить стили и прописать свои **необходимо добавить теги-контейнеры** для решения вопроса с расположением объектов и их стилизацией



Добавление содержимого

Для этого будем использовать классические теги HTML, к **примеру:**

```
</neaa>
<body>
  <div class="main">
    <div class="header">
      <jdoc:include type="modules" name="position-0" />
    </div>
    <div class="nav">
      <jdoc:include type="modules" name="position-1" />
    </div>
    <div class="help">
      <jdoc:include type="modules" name="position-2" />
    </div>
    <div class="content">
      <div class="col_left">
        <jdoc:include type="modules" name="position-3" />
      </div>
      <div class="col_right">
        <jdoc:include type="modules" name="position-4" />
      </div>
      <div class="block_center">
        <jdoc:include type="component" />
      </div>
    </div>
    <div class="footer">
      <jdoc:include type="modules" name="position-5" />
    </div>
  </div>
</body>
</html>
```



Добавление содержимого

ВАЖНО:

для определения, какие теги, сколько и в какой вложенности вам будут нужны – **ориентируемся на ваш макет**



ВАЖНО:

при формировании url в качестве значения к другим атрибутам, к примеру, к изображениям (атрибут src), необходимо правильно прописывать адрес:

```

```




Работа со стилями

ВАЖНО:

в Joomla3 в файле **media/jui/css/bootstrap.css** имеется все необходимое для простейшего стилевого файла – возьмите все это содержимое в ваш стилевой файл **при необходимости** (либо его часть)



Работа со стилями

Подключим два файла: **сброс стилей** и **обычный стилевой**

Сброс стилей необходимо создать и внести в него соответствующие решения

Стилевой оставим пока что без изменений

```
1 <?php
2     defined('_JEXEC') or die;
3     $doc= JFactory::getDocument();
4     $doc->addStyleSheet(JUri::base(). 'templates/'. $doc->template. '/css/reset.css');
5     $doc->addStyleSheet(JUri::base(). 'templates/'. $doc->template. '/css/style.css');
6
7 ?>
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12     <jdoc:include type="head" />
13 </head>
14 <body>
15     <jdoc:include type="modules" name="position-0" />
16     <jdoc:include type="modules" name="position-1" />
```

// проверьте в клиентской части, что сброс стилей применился



ЗАДАНИЕ

- на имеющейся основе выполните следующее:
- добавьте модули, распределите их по позициям, измените очередность линейного следования;
- пропишите стилевые решения, позволяющие сформировать визуализацию клиентской части



Магазин игрушек

Игрушки ручной работы для солнечных улыбок у вас дома

Улица • 111 • Город • Область
телефон: ###-##-##-##-###

Открыт каждый день с **10:00** до **20:00**

Приходите поиграть

Детство - замечательная пора. Нам нравится делать его ярким и запоминающимся. Дарить радость детям, развивать их творчество, создавать сказку в настоящем мире. Приходите в наш магазин поиграть или напишите нам, чтобы оставить заказ на понравившиеся вам игрушки

migaicheva.by/index.php/svyazhites-s-nami

- Главная
- Свяжитесь с нами**
- Наши игрушки
- О магазине
- Новости
- Интернет-магазин



Код выглядит следующим образом:

```
12     </head>
13     <body id="block_primer" class="site">
14
15     <div id="main_header">
16         <jdoc:include type="module" name="search" />
17         <jdoc:include type="modules" name="position-0" />
18         <jdoc:include type="modules" name="position-1" />
19         <jdoc:include type="modules" name="position-2" />
20         <jdoc:include type="modules" name="position-3" />
21         <jdoc:include type="modules" name="position-4" />
22         <jdoc:include type="modules" name="position-5" />
23     </div>
24     <div class="body">
25     <div class="container">
26     <div class="span8">
27         <jdoc:include type="message" />
28         <jdoc:include type="component" />
29     </div>
30     <div class="span4">
31         <jdoc:include type="modules" name="position-6" />
32         <jdoc:include type="modules" name="position-7" />
33     </div>
34
35     </div>
36     </div>
37     </body>
38 </html>
```



В качестве стилевого файла был создан файл **template.css**, а его содержимое было взято из файла **media/jui/css/bootstrap.css**, добавлен **свой стилевой файл** и обратите внимание на строку 4

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <?php Jhtml::_('bootstrap.framework'); ?>
5         <?php
6         defined('_JEXEC') or die;
7         $doc = JFactory::getDocument();
8         ?>
9         <?php $doc->addStyleSheet('templates/.'.$this->template.'/css/template.css'); ?>
10        <?php $doc->addStyleSheet('templates/.'.$this->template.'/css/custom.css'); ?>
11        <jdoc:include type="head" />
12    </head>
```



Внесем в ваш (пустой) стилевой файл некоторые изменения.

ВАЖНО: для изменения стилей отдельных объектов придется все равно изучать исходный стилевой файл – чтобы иметь возможность достучаться до некоторых объектов системы (к примеру, к ссылкам или заголовкам материалов)



К примеру, это могут быть:

```
48 |
49 | /*меняем свойства блока заголовка - этот селектор мы создали сами #main_header
50 | и свойства селекторов, которые взяли из первого стилевого файла - .container, .span8
51 | .span4 - это их ширина и размещение*/
52 |
53 | #main_header {
54 |     background-image: url(../images/fon2.jpg);
55 |     background-position: center center;
56 |     background-repeat: no-repeat;
57 |     padding: 0 6%;
58 |     width: 88%;
59 | }
60 |
61 | .container {
62 |     width: 90%;
63 |     padding-bottom: 20px;
64 | }
65 |
66 | .span8 {width: 70%;}
67 |
68 | .span4 {
69 |     width: 20%;
70 |     padding-top: 70px;
71 | }
72 |
73 |
```




К примеру, это могут быть:

```
5 /*Для всей страницы задаем свойства, чтобы исключить конфликты уже имеющимся классом для страницы .site,  
6 вводим свой селектор #body_primer*/  
7  
8 #block_primer {  
9     background-color:#fff7f8;  
10    font-family:Verdana;  
11    font-size:14px;  
12    line-height:2em;  
13 }  
14  
15
```



К примеру, это могут быть:

```
46
47
48 /*Название статьи (материала) задается чеез тег h2 - меняем его свойства*/
49
50 h2 {
51     font-family:Marck Script;
52     color:#ad071c;
53     font-size:35px;
54     font-weight: normal;
55     background-color: #fef8fc;
56 }
57
```

т.к. будем использовать веб-шрифт Google, то подключаем его в самом начале стилей:



```
1 /*подключаем веб-шрифт Google*/
2
3 @import url(http://fonts.googleapis.com/css?family=Marck+Script&subset=latin,cyrillic);
4
5
```



К примеру, это могут быть:

```
14
15 /*в первом блоке мы создавали html-модули - название магазина,
16 время работы, место расположения, для них также прописываем стили*/
17
18 .clr {clear:both;}
19
20 div#shopname {
21     text-align:right;
22     color:#ad071c;
23     font-size:70px;
24     padding-top:50px;
25     padding-bottom:20px;
26     font-family:Marck Script;
27 }
28
29 div#tagline {
30     float:right;
31     text-align:right;
32     color:#777;
33     font-size:30px;
34     font-family:Marck Script;
35     margin-top:20px;
36     padding-bottom:30px;
37 }
--
```



К примеру, это могут быть:

```
14
15 /*в первом блоке мы создавали html-модули - название магазина,
16 время работы, место расположения, для них также прописываем стили*/
17
18 .clr {clear:both;}
19
20 div#shopname {
21     text-align:right;
22     color:#ad071c;
23     font-size:70px;
24     padding-top:50px;
25     padding-bottom:20px;
26     font-family:Marck Script;
27 }
28
29 div#tagline {
30     float:right;
31     text-align:right;
32     color:#777;
33     font-size:30px;
34     font-family:Marck Script;
35     margin-top:20px;
36     padding-bottom:30px;
37 }
--
```



К примеру, это могут быть:

```
38
39 div#siteaddress {
40     float:left;
41 }
42
43 div#shophours {
44     float:right;}
45
46 div#shophours strong {color:#ad071c;}
47
48 div#siteaddress, div#shophours {
49     border-top:3px solid #ad071c;
50     margin-top:20px;
51     padding-top:10px;
52     font-size:16px;
53     font-family:Verdana;
54 }
55
56
```



Установка шаблона

ВАЖНО:

если вы создавали свой шаблон **локально**, то необходимо провести его установку для дальнейшего использования в админке

Установку можно провести, перенеся в файловую систему папку шаблона либо используя архив



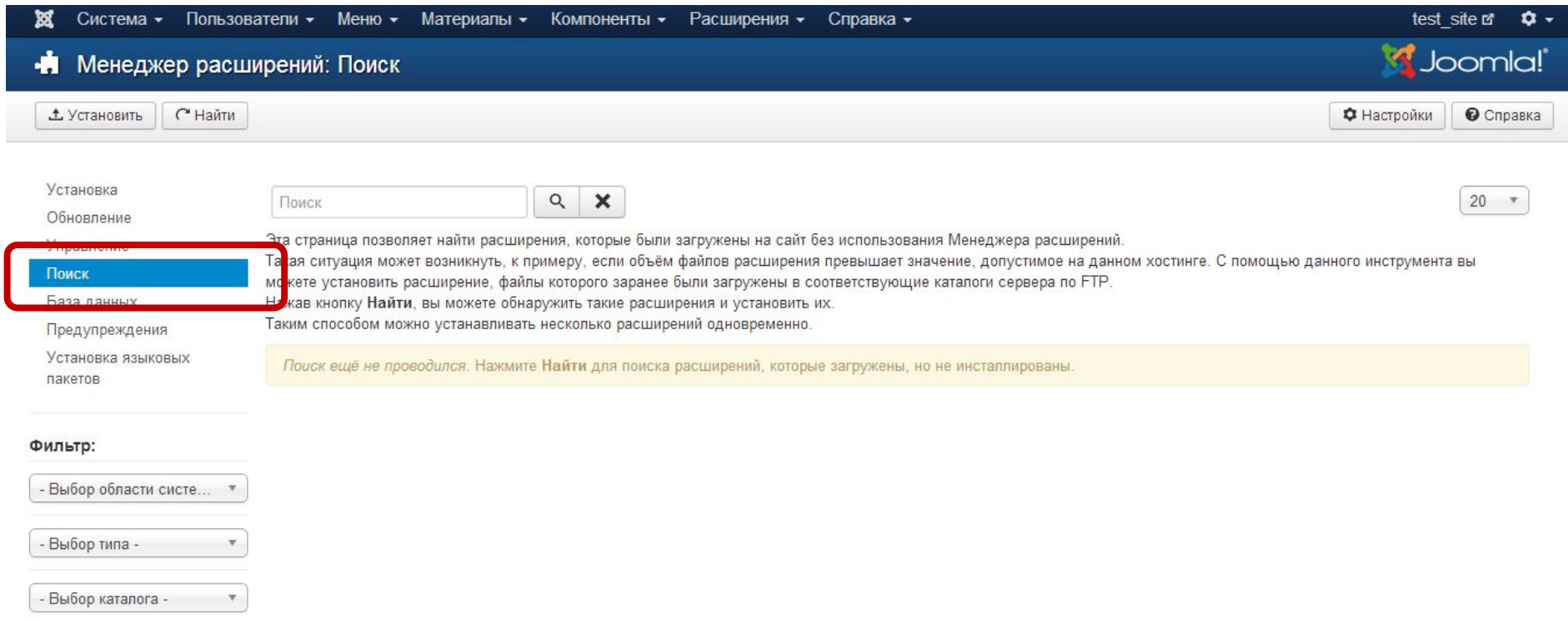
Установка шаблона

Для того, чтобы установить ваш созданный шаблон, нужно воспользоваться **Менеджером расширений** (далее вы сможете его изменять и совершенствовать, имея возможность просмотра)

The screenshot shows the Joomla! administrator interface. At the top, there is a navigation menu with items: Система, Пользователи, Меню, Материалы, Компоненты, **Расширения**, and Справка. The 'Расширения' (Extensions) menu is highlighted with a red box, and its dropdown menu is open, showing options: Менеджер расширений (highlighted), Менеджер модулей, Менеджер плагинов, Менеджер шаблонов, and Менеджер языков. Below the navigation bar, the page title is 'Менеджер расширений: Установка'. On the left sidebar, there is a menu with 'Установка' (Installation) selected, and other items like Обновление, Управление, Поиск, База данных, Предупреждения, and Установка языковых пакетов. The main content area shows a notification about the Joomla! extension catalog and a button 'Добавить закладку "Установить из JED"'. Below this, there are three buttons: 'Загрузить файл пакета', 'Установить из каталога', and 'Установить из URL'. The 'Установить из каталога' button is highlighted. Further down, the section 'Загрузить и установить расширение Joomla' is visible, with a 'Файл пакета' field and a 'Выберите файл' button. At the bottom of this section, there is a 'Загрузить и установить' button.



Далее перейдите на страницу **Поиск**



Система ▾ Пользователи ▾ Меню ▾ Материалы ▾ Компоненты ▾ Расширения ▾ Справка ▾ test_site ⌵ ⚙

Менеджер расширений: Поиск Joomla!

Установить Найти Настройки Справка

Установка Обновление Управление **Поиск** База данных Предупреждения Установка языковых пакетов

Поиск

Эта страница позволяет найти расширения, которые были загружены на сайт без использования Менеджера расширений. Такая ситуация может возникнуть, к примеру, если объём файлов расширения превышает значение, допустимое на данном хостинге. С помощью данного инструмента вы можете установить расширение, файлы которого заранее были загружены в соответствующие каталоги сервера по FTP. Нажав кнопку **Найти**, вы можете обнаружить такие расширения и установить их. Таким способом можно устанавливать несколько расширений одновременно.

Поиск ещё не проводился. Нажмите **Найти** для поиска расширений, которые загружены, но не установлены.

Фильтр:

- Выбор области систе... ▾

- Выбор типа - ▾

- Выбор каталога - ▾



Установка шаблона

Нажмите на кнопке **Найти** – в списке появится только что созданный шаблон

test_site

Менеджер расширений: Поиск

Настройки Справка

Установка Обновление Управление Поиск База данных Предупреждения Установка языковых пакетов

Поиск

Название	Тип	Версия	Дата	Каталог	Клиент	Автор	ID
<input type="checkbox"/> ourtemplate	Шаблон		Unknown	н/д	Сайт	Unknown	10004

Фильтр:

- Выбор области систе...
- Выбор типа -
- Выбор каталога -

Эта страница позволяет найти расширения, которые были загружены на сайт без использования Менеджера расширений. Такая ситуация может возникнуть, к примеру, если объём файлов расширения превышает значение, допустимое на данном хостинге. С помощью данного инструмента вы можете установить расширение, файлы которого заранее были загружены в соответствующие каталоги сервера по FTP. Нажав кнопку **Найти**, вы можете обнаружить такие расширения и установить их. Таким способом можно устанавливать несколько расширений одновременно.



Установка шаблона

Установите флажок напротив этого шаблона (выберите его) и нажмите **Установить**

Система ▾ Пользователи ▾ Меню ▾ Материалы ▾ Компоненты ▾ Расширения ▾ Справка

Менеджер расширений: Поиск

Установить Найти

Установка
Обновление
Управление
Поиск
База данных
Предупреждения
Установка языковых пакетов

Поиск

<input type="checkbox"/>	Название ^	Тип	Версия	Действие
<input checked="" type="checkbox"/>	ourtemplate	Шаблон		Unk

Фильтр:

- Выбор области систе... ▾

- Выбор типа - ▾

Эта страница позволяет найти расширения, которые были загружены на сайт без использования менеджера расширений. Такая ситуация может возникнуть, к примеру, если объем файлов расширения превышает лимит, установленный в настройках сайта. Вы можете установить расширение, файлы которого заранее были загружены в соответствии с инструкцией. Нажав кнопку **Найти**, вы можете обнаружить такие расширения и установить их. Таким способом можно устанавливать несколько расширений одновременно.



Установка шаблона

Установленный шаблон появится в списке доступных шаблонов и для него будет создан **стиль по умолчанию**

Менеджер шаблонов: Стили

★ По умолчанию ✎ Изменить 📄 Дублировать ✕ Удалить

Стили
Шаблоны

Фильтр: [Поиск] [Очистить]

Стиль По умолчанию

<input type="checkbox"/>	Beez3 - Default	<input type="checkbox"/>	☆
<input type="checkbox"/>	Hathor - Default	<input type="checkbox"/>	☆
<input type="checkbox"/>	isis - Default	<input checked="" type="checkbox"/>	★
<input type="checkbox"/>	ourtemplate - По умолчанию	<input type="checkbox"/>	☆
<input type="checkbox"/>	protostar - Default	<input type="checkbox"/>	☆
<input type="checkbox"/>	protostar - Default (2)	<input checked="" type="checkbox"/>	★

Фильтр:
- Выбор шаблона -
- Выбор области систе...



ИНСТИТУТ БИЗНЕСА
И МЕНЕДЖМЕНТА
ТЕХНОЛОГИЙ БГУ

Спасибо за внимание
