

LinkorCMS FAQ: Часто задаваемые вопросы по системе

В этом дополняющемся FAQ рассмотрены вопросы, которые наиболее часто задавались на официальном форуме системы. Хотя он и пишется для LinkorCMS v1.4, но большая часть ответов подходит и для версии 1.3. В противном случае различия будут оговариваться. Так же еще меньшая часть ответов действительна и для версии 1.2, но, по причине ее устаревания и почти не встречающейся в природе, она специально оговариваться не будет. Также будет рассмотрена некоторая культура поведения на форуме.

Последняя версия CMS на момент написания: 1.4.1.

Дата последнего редактирования FAQ: **.03.2015.

Культура поведения на форуме

- Выбирайте подходящий раздел для вашей темы. Все разделы имеют довольно говорящие описания. Не стоит создавать дубли тем. Создав десять одинаковых тем с вашим вопросом, вы не ускорите и не улучшите ответ на него, а только покажете свое невежество. Также перед созданием убедитесь, что такой вопрос уже не задавался в данном FAQ или на форуме с помощью встроенного поиска и поисковой системы. Если в точности такой же (или очень похожий) вопрос освещался, то воспользуйтесь информацией в найденной теме. А если ее недостаточно - спросите в ней. В противном случае создайте тему про ваш вопрос в подходящем разделе. Если вы все же не смогли подобрать раздел, то создайте в разделе Общение, потом ее переместят в подходящий.
- Давайте темам осмысленные имена. Не стоит называть их как "Вопрос", "Помогите", "Не работает". Давайте название, которое точно и наиболее полно раскрывает содержание темы. Например, "Вопрос по установке модулей", "Помогите с шаблоном блока меню", "Не работает поиск по сайту".
- Точно и подробно опишите проблему в первом посте (сообщении) темы. Если это технический вопрос, то не лишнем привести версию CMS, версию PHP, тип базы данных (файловая или MySQL), полный текст ошибок (если есть. предварительно включайте показ всех ошибок), адрес сайта. В некоторых вопросах эта информация очень помогает дать точный ответ. Но иногда даже ее недостаточно и приходится смотреть сайт глубже.
- Форум - это не чат. Не создавайте несколько сообщений подряд просто так (например, если не прошло хотя бы несколько дней с целью поднятия (up/apa) темы). Дописывайте текст в предыдущее сообщение, вместо мультипостинга (создание в короткий промежуток времени нескольких сообщений подряд от одного и того же автора). Так же не оффтопьте (не пишите сообщений хоть и с полезной информацией, но не по теме. это часто касается вопросов, которым уместно создать отдельную тему) и не флудите (не пишите сообщений, не содержащие полезной информации. к этому относится всякое выражение эмоций, смайлики и прочее).

Описание модуля Страницы

В системе есть модуль для создания статических страниц (то есть таких, где просто отображается html текст без какой-либо серверной логики или PHP кода). Но, так как он объединен с редактором блока меню, то иногда вызывает вопросы у начинающих пользователей системы.

В модуле можно создать два типа записей: страницу и ссылку. Первая представляет собой статическую страницу, которая отображает заданный в ней текст во адресу /name.html или index.php?name=pages&file=name (где name - ссылка, указываемая при редактировании страницы) при его вызове. Вторая же является просто произвольной ссылкой в блоке меню. При этом, если вы хотите чтобы в блоке меню была ссылка на созданную статическую страницу, то нет необходимости создавать ссылку. Достаточно при редактировании страницы отметить соответствующую опцию.

Так же список ссылок и статей в модуле представляет из себя сортирующийся вложенный список того, как должен происходить вывод ссылок в блоке меню.

Подключение JS скриптов / галерей (лайтбоксы и подобные) / редакторов текста / встраивание счетчиков

Способы различны, все зависит от ваших целей, скриптов и версии системы.

Подключение JS скриптов

Javascript скрипты подключаются в HTML документах с помощью тега <script>, причем код скрипта может быть как ставлен в тег напрямую, так и указанием ссылки на файл с кодом.

Если необходимо вставить скрипт в заголовок элемента (секция <head>), то можно воспользоваться следующими методами класса шаблонизатора:

```
System::site()->AddJS($script);  
System::site()->AddJSFile($filename, $local, $inc, $params);
```

Первый метод принимает всего один аргумент \$script с текстом вашего скрипта, который будет вставлен в заголовок документа. Второй же метод предназначен для вставки файла со скриптом и имеет четыре аргумента: \$filename - ссылка или адрес скрипта; \$local логическая переменная, указывающая на то, что является ли файл локальным; \$inc - вставлять ли скрипт ссылкой на файл с кодом или вставить в страницу код из файла (грубо говоря, актуально только для локального файла); \$params - дополнительные атрибуты для тега <script> при вставке файлом.

Если же необходимо вставить скрипт в тело документа (секцию <body>), то скрипт можно вставить напрямую в нужное место шаблона темы оформления. Если же шаблонизатор системы портит скрипт, ошибочно находя в нем свои теги, то можно окружить скрипт шаблонным блоком {noblocks}{/noblocks} (актуально для LinkorCMS v1.4), который отключает шаблонизацию HTML кода, окруженного этим блоком.

Так же можно создать шаблонную переменную с кодом вашего скрипта (что важно для LinkorCMS v1.3, где нет тега noblocks), используя следующий метод класса шаблонизатора:

```
System::site()->SetVar($Block, $VarName, $Value, $SubId = 0);
```

Он принимает следующие аргументы: \$Block - имя шаблонного блока, в котором необходимо создать переменную (можно воспользоваться глобальным блоком template); \$VarName - желаемое имя для вашей переменной; \$Value - значение шаблонной переменной (в данном случае это должен быть код вашего скрипта); \$SubId - ?.

Далее в нужном месте шаблона вызываете созданную шаблонную переменную как {page.name} и шаблонизатор в это место вставит ваш Javascript код (пояснения: page является алиасом template; переменные, созданные во вложенном блоке не могут использоваться в родительском, но так как

template - глобальный блок, то в случае его использования на этот факт можно не обращать внимание).

Вставить вызов метода можно, например, в файл init.php темы оформления.

Так же скрипты можно вставлять в статические страницы и текстовые блоки в админ-панели сайта, так как в них для разметки используется HTML без каких-либо ограничений. В некоторых случаях, когда используется визуальный редактор, последний может вмешиваться во вставляемый код, что может изменить ожидаемый результат. В таких ситуациях стоит на время отключать визуальный редактор в разделе Расширения -> Плагины (или Плагины в LinkorCMS v1.3).

Встраивание счетчиков

Так как большинство счетчиков представляют гибриды HTML и JS кода (вставки прозрачной картинки 1x1 пикселей и JS кода для получения детальной информации), то для этого работает все то, что описано в предыдущем разделе.

Для встраивания счетчиков в существующий блок статистики, можно вставить их код в контентный шаблон блока Онлайн: /templates/ваша_тема/block/content/online.html.

Подключение сторонних визуальных HTML редакторов в админ-панель

*

Подключение галерей

*

Вставка PHP кода в шаблон / Встраивание сторонних скриптов в систему

Начиная с LinkorCMS v1.3, шаблонизатор при считывании шаблонов использует include, а не file_get_content(), так что PHP код в шаблонах вставляется просто с помощью PHP тегов:

```
<?php
echo 'Hello!';
?>
```

С помощью этого можно поставить скрипты бирж ссылок (Sape и подобные) и прочие на сайт. При подключении шаблона, скрипт исполнится и его вывод будет находиться в месте вставки.

Если же скрипт более сложный и принимает аргументы ("имеет несколько страниц"), то вставка в какие-то существующие страницы сайта может не подходить. В таком случае можно встроить скрипт как простейший модуль (тем более довольно часто такие скрипты выполняют функции, которые схожи с функциями модулей). Пример простейшего модуля можно скачать по ссылке: [link11](#).

Дублирование (создание копий) модулей / Изменение адреса модуля

Создать копии модулей возможно, но потребуются некоторое время для кропотливой работы и некоторые знания языка PHP. Вся суть сводится к созданию новой папки в папке modules с именем копии, а так же всех ресурсов модуля: таблицы в базе данных, настройки, шаблоны и

прочее. Разве что шаблоны и скрипты могут остаться общими, если не требуется их изменение для корректной работы копии модуля или "подгонка" под сайт. Далее изменяется сам код модуля. Необходимо исправить все ссылки и места, использующие какие-либо ресурсы.

Переименование модуля можно считать частным случаем дублирования модуля. В таком случае достаточно переименовать папку с модулем и изменить все ссылки в нем. Какие-другие изменения могут не потребоваться, но в любом случае переименовать модуль проще, чем его скопировать (ведь он должен стать обособленным от своего родителя).

В процессе копирования модуля можно выделить его название в отдельную константу и написать инсталлятор/деинсталлятор (если его нет). В таком случае, при необходимости в будущем создать еще одну копию модуля, можно будет обойтись изменением константы и, к примеру, названия папки с ресурсами.

Описание шаблонизатора системы

*

Строение тем оформления

Все темы оформления, предназначенные для пользовательской части сайта, имеют свою собственную папку в директории /templates/. В ней же располагается и тема default_admin, которая предназначается для админ-панели сайта (в LinkorCMS v1.3 эта папка отсутствует, а тема располагается в /base/admin/template/).

Установить тему для использования на сайте можно в разделе Общие настройки в админ-панели сайта. Начиная с LinkorCMS v1.4, темы оформления считаются расширениями и должны иметь файл описания info.xml. При его отсутствии система не будет "видеть" вашу тему. Для предыдущих версий системы его наличие не требуется.

В комплекте с системой поставляются две темы оформления: IgrimTheme и default (появилась в LinkorCMS v1.3.4, а в v1.4 была переименована в default). Они довольно простые по строению и могут использоваться как основа для ваших тем или изучения строения тем оформления.

Каждая тема содержит следующие папки:

block - шаблоны с внешним оформлением боковых блоков

block/content - шаблоны, отвечающие за оформление контента боковых блоков

images - папка для изображений, которые используются в теме (так же часть изображений находится в папке /images/ в корне сайта)

java - папка для javascript файлов, используемых в теме (множество других скриптов, используемых системой, можно найти в папке /scripts/ в корневой директории сайта; но редактируйте их осторожно и только в случае крайней необходимости)

module - папка для шаблонов оформления модулей системы

style - папка с CSS файлами оформления темы

table - папка, содержащая файлы для построения таблиц шаблонизаторов (в LinkorCMS v1.4 оставлена для совместимости)

themes - папка, которая содержит альтернативные файлы общего шаблона оформления страницы, которые могут устанавливаться в админ-панеле для модулей вместо theme.html

Так же в корне папки с темой оформления содержатся следующие файлы:

404.html - шаблон страницы ошибки HTTP 404

alert_message.html - шаблон вывода ошибок в верхней части страницы

box.html - шаблон оформления центральных блоков (box'ов), который используется при выводе контента модулями

breadcrumbs.html - шаблон оформления для так называемых "хлебных крошек" или навигационной цепочки от корня сайта до текущей страницы

comment.html - шаблон оформления комментария

comment_form.html - шаблон оформления формы добавления комментария

edit_comment.html - шаблон оформления формы изменения комментария

footer.html - шаблон оформления футера (нижней части страницы, подвала) страницы; содержание шаблона подставляется шаблонизатором в переменную блока {page.footer}; файл может отсутствовать и обычно используется для разгрузки шаблона theme.html

header.html - шаблон оформления хейдера (верхней части страницы, шапки); содержание шаблона подставляется шаблонизатором в переменную блока {page.header}; файл может отсутствовать и обычно используется для облегчения шаблона theme.html

info.xml - файл с информацией о теме оформления

init.php - PHP код инициализации темы (например, подключение скриптов и файлов стилей, запускается однократно системой при обработке запроса)

login.html - шаблон, используемый для отображения формы авторизации в центральном блоке

message.html - шаблон для вывода сообщений с помощью модуля Сообщения

navigation.html - общий шаблон для пагинатора постраничной навигации

navigation_item.html - шаблон элемента для пагинатора

no_access.html - шаблон для вывода установленного сообщения при закрытом сайте (offline mode)

out.html - шаблон для промежуточной страницы переадресации

theme.html - общий шаблон страницы

Выбор типа базы данных (файловая или MySQL)

Система для хранения контента и настроек может использовать два типа баз: файловая БД и MySQL. При использовании первой, система хранит все данные в файлах, не требуя наличия сервера MySQL и подключения к нему. По умолчанию все данные файловой БД лежат по пути /db/linkorcmsdb/ (указывается при установке системы). Каждая таблица имеет два файла: *.FRM содержит данные, находящиеся в таблице, а *.MYD - ее структуру и служебную информацию.

Не смотря на простоту и независимость файловой БД, это имеет и минусы. Невозможно сделать быструю, мощную и надежную базу данных, которая бы являлась простой, написанной на PHP и не работающей в режиме сервиса (демона, службы).

В связи с этим, **строго рекомендуется** использовать в качестве базы данных MySQL при ее наличии, особенно в случае высоконагруженных порталов и/или больших объемов данных. Все платные, да и большинство бесплатных хостингов на сегодняшний день предоставляют базу данных MySQL, так что все меньше остается случаев, где использование файловой БД - это единственный выход использовать LinkorCMS.

Обнуление (порча) таблиц файловой БД / Не работает блок Онлайн, Аудит

Не смотря на то, что файловая БД улучшалась, все же остаются шансы на то, что какая-то таблица может быть повреждена или обнулена. Обычно это происходит на нагруженных записью таблицах вроде online, данные которой обновляются при каждом запросе к системе, а так же файлы Аудита (им производится запись действий администраторов в админ-панеле).

Довольно часто данный дефект проявляется как неработоспособность блока Онлайн, что является следствием повреждении таблицы online.

Решается проблема восстановлением таблиц из резервной копии, а при ее отсутствии - из базы данных установщика системы (чистые таблицы хранятся в /setup/bases/). Так же можно перейти на использование базы данных MySQL.

Конвертация между файловой базой данных и MySQL

Сконвертировать файловую базу данных в MySQL можно с помощью установщика системы. Для этого достаточно на установленной системе LinkorCMS v1.4 удалить файл блокировки /config/setup_lock.php и выбрать "Конвертировать БД".

Для LinkorCMS v1.3 необходимо скопировать БД сайта (по умолчанию располагается в /db/linkorcmsdb/) в папку с базой установщика (/setup/bases/). Далее в инсталляторе совершить установку системы на БД MySQL и бросить процесс на шаге создания администратора.

Конвертация базы данных MySQL в файловую БД возможен, но скрипта-конвертера для этого процесса не существует (и мало практического смысла этой операции).

Не создаются новые блоки на LinkorCMS v1.3

Довольно частая проблема связана с установкой расширения от Lemur'a, позволяющее настраивать на каких страницах показывать блоки, а на каких - нет. Дело в том, что это расширение добавляет в таблицу blocks два новых столбца (showin и showin_uri) и при удалении расширения путем замены модифицированных файлов модуля blocks на стандартные будет происходить ошибка в INSERT запросе из-за несоответствия столбцов. Для корректного и полного удаления модификации, необходимо удалить и эти столбцы. После этого ошибка с созданием новых блоков пропадет.

Использование ЧПУ, когда веб-сервером является не Apache

Этот вопрос не касается случаев, когда в качестве бэкенд сервера используется Apache, а в качестве фронтенд - какой-то другой (например, nginx), так как фронтенд сервер будет просто проксировать запросы от клиента к бэкенд серверу и для системы никакой разницы в окружении не будет.

Оригинально система рассчитана на использование веб-сервера Apache и ЧПУ правила, находящиеся в файле .htaccess, написаны именно под него. Но нет никаких серьезных ограничений, не позволяющие использовать LinkorCMS под другими веб-серверами. Главное реализовать аналогичные правила в имеющемся веб-сервере.

Для веб-сервера nginx можно пользоваться следующими правилами:

```
server {
    listen 80;

    server_name primer.ru;
    root /var/www/primer.ru;
    index index.php;

    location = /.htaccess { deny all; }
    location = /favicon.ico { log_not_found off; access_log off; }
    location = /robots.txt { log_not_found off; access_log off; }
    location ~ /\.(js|css|png|jpg|jpeg|gif|ico)$ { access_log off; }
    location ~ ^/(db|log|.git|cache|backup|config) { deny all; }
    location ~ ^/(?!admin.php|index.php|rss.php|setup.php).*\.php$ { deny all; }
    location / {
        if (!-e $request_filename) {
            rewrite ^/index.html$ /index.php last;
            rewrite ^/(.*)$ /index.php?ufu=$1 last;
            break;
        }
    }
    location ~ .php$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/run/php5-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}
```

Для веб-сервера Lighttpd (цитата из сообщения пользователя Forwarder с официального форума системы):

Так как Lighttpd не понимает .htaccess, то для запуска LinkorCMS в Lighttpd необходимо сделать следующее: 1. Включить mod_rewrite в файле конфигурации lighttpd.conf, убрав перед ним знак #. 2. В конец файла lighttpd.conf записать следующее:

```
url.rewrite-once = (  
    "^/(.*).(css|js|php|jpg|png|gif|.htm|.html)$" => "$0",  
    "^index.html$" => "index.php",  
    "^/?(backup|images|blocks|modules|plugins|scripts|templates|set  
up|uploads|errorpages|/..php)."$ => "$0",  
    "^/(.*)$" => "/index.php?ufu=$1"  
)
```

Для IIS можно найти модули, позволяющие использовать правила расширения mod_rewrite Apache, который используется системой для реализации ЧПУ. Но работа ЧПУ под этим веб-сервером не проверялась.

Вопросы про ограничение размера загружаемых файлов системой

*

Модули

Модули - это / Модули предназначены.

Находятся все модули в папке /modules/ в корневой директории сайта, причем каждый модуль имеет свой каталог (например, /modules/news/). В папке с модулем находятся следующие РНР файлы:

index.php - файл модуля, предназначенный для работы в пользовательской части сайта; можно назвать его так же точкой входа у модуля (файл может отсутствовать, если модуль не предназначен для использования в пользовательской части сайта)

admin.php - администраторская часть модуля, предназначенная для работы в админ-панеле, которая доступна только администраторам сайта; так же является точкой входа модуля, но только уже в админ-панеле (может отсутствовать, если модуль не имеет административной части)

init.php - файл инициализации модуля, который выполняется системой до исполнения index.php модуля и подключения шаблонизатора (фактически после базовой инициализации системы; файл может отсутствовать; не действует для админ-панели)

info.php - файл с информацией о модуле, который используется системой при поиске неустановленных модулей (может отсутствовать, если модуль уже установлен; обычно такое наблюдается у модулей, которые идут в дистрибутиве вместе с самой системой LinkorCMS)

info.xml - файл с информацией в формате XML, введен в v1.4 (в шаблонах используется, в модулях пока не замечен?)

В точках входа index.php и admin.php для пользовательской части и админ-панели управления обычно в начале происходит так называемый роутинг адресов (обычно с помощью конструкции

switch). То есть определяется нужная функция-обработчик для текущего запроса. Про строение адресов и работу системы ЧПУ рассказано в "*".

Дальше располагаются функции, обрабатывающие запрос, функции-рендеры и вспомогательные. Строение сильно может варьироваться в зависимости от функций и масштаба модуля, хотя и имеет общие черты, описанные выше.

С системой модуль может взаимодействовать через системный класс System. Он был введен в LinkorCMS v1.4 для того, чтобы отказаться от глобальных переменных \$site (объект шаблонизатора пользовательской части сайта), \$user (объект класса пользователей), \$config (массив настроек модулей и системы до 1.4), \$db (объект для работы с базой данных), \$admin (объект шаблонизатора администраторской части сайта), \$cache (объект для работы с кэшем).

Класс System не совсем равнозначно заменяет глобальные переменные. Например, для работы с настройками предлагается использовать методы вместо прямой работы с массивом. Так же в нем есть методы для работы с логом системы и вывода ошибок.

Начиная с LinkorCMS v1.4, существует новая возможность построения модулей. В ней вместо традиционного роутинга с помощью switch и нескольких функций, предлагается создать класс с методами. А в комментариях к методам указать правила роутинга. Самим роутингом и выводом нужных методов дальше займется класс System.

Блоки

Блоки предназначены для отображения различного контента в боковых колонках сайта (или вверху или внизу центральной колонки, обрамляя вывод модуля).

Находятся все блоки в папке /blocks/ корневой директории сайта. Каждый блок имеет свою подпапку для хранения PHP файлов с кодом модуля. Обычно у блока имеются следующие файлы:

index.php -

constructor.php - используется для вывода дополнительных полей настройки блока при редактировании блока (может отсутствовать, если дополнительных настроек для блока не требуется)

editsave.php - используется для сохранения значений дополнительных полей блока после отправки формы редактирования (может отсутствовать, если дополнительных настроек для блока не требуется)

info.php - файл с информацией о блоке, который используется системой при поиске неустановленных блоке (отсутствует в системах до v1.4)

install.php - файл с кодом установки блока, который выполняется при установке блока (отсутствует в системах до v1.4)

uninstall.php - файл с кодом удаления блока, который используется при удалении блока (отсутствует в системах до v1.4)

Системы до версии 1.4 не поддерживают установку и удаление блоков. Установленные блоки должны быть зарегистрированы в таблице blocks подобно модулям (у них используется таблица modules), но, в отличие от последних, штатных способов установки/удаления блоков в них не было реализовано. Проводить установку/удаление блоков в этих системах необходимо или вручную, или с помощью внешнего PHP скрипта (как правило, их код представляет собой

инициализацию ядра системы и последующий запуск кода, который аналогичен `install.php/uninstall.php` для 1.4).

Плагины

Действие плагинов, в отличие от модулей и блоков, довольно разнообразное. Их можно разделить на несколько видов:

- Плагин, который запускается при каждом запросе к системе и исполняет какой-то код
- Плагин, который запускается при каждом запросе к пользовательской части сайта и исполняет какой-то код
- Плагин, который запускается при каждом запросе к администраторной части сайта (админ-панеле) и исполняет какой-то код
- Плагин, который вызывается при вызове по ссылке `/index.php&name=plugins&p=plugin_name` и исполняет какой-то код
- Может быть группа плагинов, которые взаимодействуют с модулем. Например, стандартные модули поиска и карты сайта используют свои группы плагинов, производя поиск или получение контента с помощью плагинов. Таким образом, в самом модуле производится только вывод информации и вызов плагинов из какой-то группы. Само же построение контента производится с помощью плагинов. Так можно добавить возможность поиска по новому модулю в стандартный поиск без изменения кода системы.
- Может быть группа плагинов, но из нее вызывается только один плагин. Чем-то похоже на предыдущий пункт. Например, так подключаются HTML редакторы в админ-панеле.

Все плагины находятся в папке `/plugins/` в корневой директории системы. Плагины находятся каждый с своей папке, плагины из групп находятся в подпапке соответствующей группы.

Плагины имеют следующие PHP файлы:

index.php - файл с кодом плагина, является точкой входа

info.php -

install.php - файл, который исполняется при активации (установки) плагина

uninstall.php - файл, который исполняется при деактивации (удаления) плагина

мб что-то еще

Ссылки и ЧПУ

Работа ЧПУ

Системная реализация ЧПУ появилась только с версии 1.3.4. До этого лишь некоторые модули самостоятельно его реализовывали.

Работа ЧПУ устроена следующим образом. Для поддержки системы ЧПУ, многие ссылки обрабатываются перед выводом с помощью функции ядра:

Ufu(\$Url, \$Ufu, \$NavLink, \$NavParam, \$MinOrder);

Она принимает следующие параметры: \$Url - обычная не ЧПУ ссылка (используется для вывода при отключенной системе ЧПУ, а так же из нее берутся GET параметры для составления ЧПУ ссылки), \$Ufu - шаблон ЧПУ ссылки **

При обработке ссылки, происходят многие вещи. Это можно посмотреть в файле ufu.php, который является модулем ядра. Если кратко, то проверяется наличие шаблоны для данной ссылки в талице rewrite_rules. Если ее нет, то происходит разбор ссылки, добавление в таблицу, сброс кэша, построения готовой для вывода ссылки.

При запросе к системе, веб-сервер должен передавать все запросы к сайту, за исключением некоторых реально существующих папок. Этот запрос должен быть передан системе как GET параметр ufu. Далее система в своей точке входа после базовой инициализации вызывает функцию ядра UfuInit(), которая запускает разбор запроса и пытается преобразовать ЧПУ ссылку по уже разобранным правилам в таблице rewrite_rules в обычные GET параметры.

Как видно, система ЧПУ является некой надстройкой, которая появилась в LinkorCMS v1.3.4.